Soft Clustering Algorithms
Theoretical and Practical Improvements

Kathrin Bujna

A dissertation submitted to the
Department of Computer Science
Paderborn University

for the degree of
Doktor der Naturwissenschaften
(doctor rerum naturalium)

July 25, 2017

accepted at the recommendation of

Prof. Dr. Johannes Blömer
(Paderborn University)

and

Prof. Dr. Eyke Hüllermeier
(Paderborn University)

defended on Oktober 4, 2017

Acknowledgements

I would like to thank my advisor Prof. Dr. Johannes Blömer for the regular discussions about the ongoing progress in research. I am particularly grateful to him for supporting my application for the doctoral forum at the SDM and my travel to the ECML PKDD. I also have to thank my co-authors, Sascha Brauer and Dr. Daniel Kuntze, and the other members of our clustering research group, Dr. Melanie Schmidt and Prof. Dr. Christian Sohler, for the good cooperation. I also want to thank all my dear colleagues at the Paderborn University, especially, Jan Bobolz, Sascha Brauer, Fabian Eidens, Dr. Peter Günther, Claudia Jahn, Jakob Juhnke, Dr. Daniel Kuntze, Dr. Gennadij Liske, Nils Löken, and David Teusner, for the relaxed and creative working atmosphere. Special thanks go to our patient IT crew, especially, Ulrich Ahlers and Thomas Thissen. Last but not least, I want to thank Martin Wistuba for his encouragement and our discussions about practical machine learning topics.

Abstract

In this thesis, we study two soft clustering approaches: fuzzy K-means clustering and model-based clustering with Gaussian mixture models. In contrast to the popular K-means hard clustering, there are hardly any algorithms for these approaches that provide guarantees on the quality of the computed clusterings.

In the first part of this thesis, we present the very first theoretical analysis of the fuzzy K-means problem that deals with approximation algorithms. The key to our results is the identification of three properties: First, we show that for certain fuzzy clusters there exist hard clusters with similar characteristics. Second, we show that there is a coarse relation between the objective functions of the fuzzy K-means and the K-means problem. Third, we show that the notion of empty hard clusters, which can effectively be ignored in a hard clustering, has a counterpart in fuzzy K-means. Due to these three properties, we are able to utilize two techniques that are known from the analysis of hard clustering problems: We show how the so-called superset sampling technique can be applied to compute an approximation for the fuzzy K-means problem. Moreover, we adapt a coreset construction that is known from K-means clustering. Simply speaking, a coreset is a small summary of a large data set that can be processed by a clustering algorithm instead of the original data set, without affecting the quality of the solution too much. We show that there is a refined version of a coreset construction for the K-means problem that yields a coreset for the fuzzy K-means problem. Furthermore, we use this construction to derive another approximation algorithm for the fuzzy K-means problem. Finally, we also consider alternative notions of fuzziness and generalize all of our results to a large class of soft clustering problems.

In the second part of this thesis, we consider a model-based clustering approach, namely, the method of maximum likelihood for estimating Gaussian mixture models. Our contribution is threefold: First, we compare two popular heuristics with one another, namely the expectation-maximization algorithm and a stochastic variant thereof. Second, we tackle the problem of initializing the expectation-maximization algorithm. We propose two new initialization methods. Thereby, we aim to close the gap between simple, but rather unreliable, methods and complex methods, whose performance crucially depends on the right choice of hyperparameters. Third, we initiate the theoretical analysis of a constrained version of the maximum likelihood estimation problem, which is known as the soft K-means problem. We derive a variant of this problem that focuses more on determining a soft clustering (than on determining a Gaussian mixture model) and present a first approach towards an approximation algorithm.

Zusammenfassung

In dieser Arbeit betrachten wir zwei Soft-Clustering Methoden: Fuzzy K-Means Clustering und modellbasiertes Clustering mittels Gaußmixturen. Im Gegensatz zum populären K-Means Clustering gibt es für diese beiden Ansätze kaum Algorithmen, die Garantien für die Güte der berechneten Clusterings bieten.

Im ersten Teil dieser Arbeit präsentieren wir die erste theoretische Analyse des Fuzzy K-Means Problems, die sich mit Approximationsalgorithmen beschäftigt. Der Schlüssel zu unseren Ergebnissen ist die Identifikation von drei grundlegenden Eigenschaften eines Fuzzy K-Means Clusterings: Erstens zeigen wir, dass es für bestimmte Fuzzy-Cluster entsprechende harte Cluster gibt, die den Fuzzy-Clustern ähneln. Zweitens zeigen wir, dass es einen groben, aber dennoch nützlichen Zusammenhang zwischen der Zielfunktion des Fuzzy K-Means Problems und der Zielfunktion des klassischen K-Means Problems gibt. Drittens zeigen wir, dass es Fuzzy-Cluster gibt, die in einem Fuzzy-Clustering in etwa so vernachlässigbar sind wie leere Cluster in einem harten Clustering. Diese drei Eigenschaften helfen uns dabei, Methoden, die für das K-Means Problem entwickelt wurden, auf das Fuzzy K-Means Problem zu übertragen: Wir zeigen, dass mit Hilfe der sogenannten Superset-Sampling Technik auch ein Approximationsalgorithmus für das Fuzzy K-Means Problem konstruiert werden kann. Außerdem übertragen wir eine Kernmengen-Konstruktion, die für das K-Means Problem entwickelt wurde, auf das Fuzzy K-Means Problem. Einfach gesagt ist eine Kernmenge eine kurze Zusammenfassung eines großen Datensatzes, die anstatt des ursprünglichen Datensatzes von einem Clusteringalgorithmus bearbeitet werden kann, ohne dass sich dadurch die Qualität des Ergebnisses zu sehr verschlechtert. Wir zeigen nicht nur, dass sich eine Kernmenge für das Fuzzy K-Means Problem berechnen lässt, wir nutzen die Konstruktion auch, um einen weiteren Approximationsalgorithmus für das Fuzzy K-Means Problem herzuleiten. Darüber hinaus betrachten wir verschiedene Varianten des Fuzzy K-Means Problems und verallgemeinern all unsere Ergebnisse.

Der zweite Teil dieser Arbeit dreht sich um den modellbasierten Clustering Ansatz, genauer gesagt, die Maximum-Likelihood-Methode für das Schätzen von Gaußmixturen. Als erstes vergleichen wir zwei Heuristiken, den klassischen Expectation-Maximization Algorithmus und eine seiner randomisierten Varianten, miteinander. Zweitens beschäftigen wir uns mit dem Problem, eine vernünftige initiale Lösung für den Expectation-Maximization Algorithmus für Gaußmixturen zu finden. Wir präsentieren zwei neue Initialisierungsmethoden und versuchen damit die Lücke zwischen den einfachen, aber eher unzuverlässigen Methoden und komplizierten Methoden, deren Qualität stark von den gewählten Hyperparametern abhängt, zu schließen. Drittens versuchen wir uns an einer theoretischen Analyse des Maximum-Likelihood-Estimation Problems. Dazu betrachten wir einen Spezialfall, der auch schlicht als das Soft-Clustering Problem bekannt ist. Wir leiten eine Variante dieses Problems her, in deren Mittelpunkt die Bestimmung eines Soft-Clusterings (anstatt die Bestimmung einer Gaußmixtur) steht, und präsentieren einen ersten Ansatz für einen Approximationsalgorithmus.

Contents

"Melmac was the name of my planet. It's also what it was made out of."

Alf

Cheat Sheet

$[N]$	$= \{1, 2, \dots, N-1, N\}$ for $N \in \mathbb{N}$ (Notation 2.1)
$\mathbb{R}_+$	$= \{x \in \mathbb{R} \mid x > 0\} = \mathbb{R}_{>0}$
$\tilde{\mathcal{O}}$	hides logarithmic factors that would appear in the classic $\mathcal{O}$-notation
max	$\max\{f(a) \mid a \in A\} = f(c)$ for all $c \in A$ with $\forall b \in A: \ f(c) \geq f(b)$
arg max	$\arg\max\{f(a) \mid a \in A\} = \{a \in A \mid \forall b \in A: \ f(a) \geq f(b)\} \subseteq A$
	Vectors and Matrices
$I_D, 0_{D,D}, 0_D$	identity $(D \times D)$-matrix, zero $(D \times D)$-matrix, D-dimensional zero vector
$(x_n)_{n \in [N]} \subset \mathbb{A}^D$	list whose n-th element is $x_n \in \mathbb{A}^D$, where $\mathbb{A} \subseteq \mathbb{R}$
$(v_d)_{d \in [D]} \subset \mathbb{A}$	D-dimensional vector whose d-th coordinate is $x_d \in \mathbb{A} \subseteq \mathbb{R}$
$(x_n)_d$	d-th coordinate of vector $x_n \in \mathbb{R}^D$ with $d \in [D]$
$\langle v, w \rangle$	$= \sum_{d=1}^{D} v_d \cdot w_d$ scalar product of $v = (v_d)_{d \in [D]}$ and $w = (w_d)_{d \in [D]}$
$\|v\|_2$	Euclidean norm of a vector $v \in \mathbb{R}^D$
$(M)_{i,j}$	(i,j)-th entry of a matrix $M \in \mathbb{A}^{D_1 \times D_2}$
$\langle M, L \rangle_F$	$= \sum_{i=1}^{D_1} \sum_{j=1}^{D_2} (M)_{ij} \cdot (L)_{ij}$ Frobenius inner product of $M, L \in \mathbb{R}^{D_1 \times D_2}$
$\|M\|_F$	$= \sqrt{\langle M, M \rangle_F}$ Frobenius norm of a matrix $M \in \mathbb{R}^{D_1 \times D_2}$
	Data Sets
X	usually a weighted data set (think of a *multi*-set)
$\mathrm{Dom}(\mathbb{A}^D, \mathbb{B})$	set of all finite data sets with data points from $\mathbb{A}^D \times \mathbb{B}$ where $\mathbb{A}^D \subseteq \mathbb{R}^D$ and $B \subseteq \mathbb{R}_{\geq 0}$ (Definition 2.3)
$((x_n, w_n))_{n \in [N]}$	weighted data set containing N ordered data points (Definition 2.3)
(x_n, w_n)	n-th data point consisting of a point (object) $x_n \in \mathbb{A}^D$ and its weight (importance) $w_n \in \mathbb{B}$
$(x_n)_{n \in [N]}$	unweighted data set containing N ordered points x_n (Definition 2.3)
$((x_n, 1))_{n \in [N]}$	also an unweighted data set containing N ordered data points $(x_n, 1)$ (Definition 2.3)
$\|X\|$	the size of the data set X (Notation 2.4)
$X \subset Y$	indicates that the data points from X are contained in the data set Y (with their corresponding multiplicity) (Notation 2.4)
$\mathtt{r}_d(X)$	range of the d-th coordinates of the points in X (Definition 2.18)
$\mathtt{diam}(X)$	maximum Euclidean distance between points in X (Definition 2.18)
$\mathtt{w}_{\min}^{(X)}$ ($\mathtt{w}_{\max}^{(X)}$)	minimum (maximum) weight of a data point in X (Definition 2.18)

Hard and Soft Cluster(ings)

$\Delta_{K-1} \subseteq \mathbb{R}^K$	closed K-simplex that contains all categorical distributions over K classes (Definition 2.6)
$\Delta_{N,K-1} \subseteq \mathbb{R}^{N\times K}$	set of all soft K-clusterings of N objects (Definition 2.6)
$P \in \Delta_{N,K-1}$	soft K-clustering of N objects (Notation 2.7)
$p_{nk} \in [0,1]$	probability that the n-th object belongs to the k-th cluster
$Z \in \{0,1\}^{N\times K}$	indicator matrix of N objects to K clusters (Notation 2.9) and a hard K-clustering of N objects if $Z \in \Delta_{N,K-1}$ (Notation 2.10)
$z_{nk} \in \{0,1\}$	indicates whether the n-th object is assigned to the k-th cluster
$R \in [0,1]^{N\times K}$	$R = (r_{nk})_{n,k}$ probabilistic membership matrix that assigns the n-th object to the k-th cluster with probability r_{nk} ($\sum_{k=1}^{K} \leq r_{nk}$) (Definition 3.1)
$r_{nk} \in [0,1]$	probability that the n-th object belongs to the k-th cluster ($\sum_{k=1}^{K} r_{nk} \leq 1$)
$A_k^{(X,R)}$	$= ((x_n, r_{nk}\cdot w_n))_{n\in[N]}$ k-th cluster of X given by the membership matrix R (Definition 3.2)
$A_k^{(X,P)}$	k-th soft cluster of X given by the soft clustering P (Eq. 2.1)
$A_k^{(X,Z)}$	k-th hard cluster of X given by the indicator matrix Z (Eq. 2.2)
$A_k \subseteq X$	k-th hard cluster written as a subset of X; corresponds to the data set $A_k^{(X,Z)}$ with appropriate indicator matrix Z (Notation 2.10)

Cluster Statistics

$\mathbf{w}(A_k^{(X,R)})$	$= \sum_{n=1}^{N} r_{nk} w_n$ weight (Definition 2.13)
$\mathbf{m}(A_k^{(X,R)})$	$= \left(\sum_{n=1}^{N} r_{nk} w_n x_n\right) / \mathbf{w}(A_k^{(X,R)})$ mean (Definition 2.14)
$\mathbf{d}(A_k^{(X,R)}, z)$	$= \sum_{n=1}^{N} r_{nk} w_n \lVert x_n - z\rVert_2^2$, where $z \in \mathbb{R}^D$, cost (Definition 2.16)
$\mathbf{var}(A_k^{(X,R)})$	$= \mathbf{d}(A_k^{(X,R)}, \mathbf{m}(A_k^{(X,R)}))/\mathbf{w}\left(A_k^{(X,R)}\right)$ variance (Definition 2.16)
$\mathbf{ucov}(A_k^{(X,R)}, z)$	$= \sum_{n=1}^{N} r_{nk}\cdot w_n (x_n - z)(x_n - z)^T$, where $z \in \mathbb{R}^D$, unnormalized covariance (Definition 2.15)
$\mathbf{cov}(A_k^{(X,R)}, z)$	$= \mathbf{ucov}(A_k^{(X,R)}, \mathbf{m}(A_k^{(X,R)}))/\mathbf{w}\left(A_k^{(X,R)}\right)$ covariance (Definition 2.15)
$\mathbf{w}_k, \mathbf{m}_k, \ldots$	short notations for $\mathbf{w}(A_k^{(X,R)})$, $\mathbf{m}(A_k^{(X,R)})$, … with respect to the given X and R (Notation 3.11)
y_{nk}	$= (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T$ (Lemma 3.17)

Fuzzy

r	a fuzzifier function (Definition 5.8)
$\mathbf{i}_r \in [1,\infty)$	increase-bounded (Definition 5.15)
$\mathbf{c}_r \in (0,1]$	contribution-bounded (Definition 5.13)
$D\cdot\mathbf{t}_r(K)$	time needed to compute a r-fuzzy K-clustering of a D-dimensional data point induced by K means (Assumption 5.19)
id	identity function (Section 5.3.1)
p_m	polynomial fuzzifier function (Section 5.3.2)
s_β	quadratic-linear fuzzifier function (Section 5.3.3)
e_γ	exponential fuzzifier function (Section 5.3.4)
$r(P)$	$= (r(p_{nk}))_{n,k}$ for $P = (p_{nk})_{n,k}$ (Definition 5.2)
$A_k^{(X,r(P))}$	$= ((x_n, r(p_{nk})w_n))_n$ is an r-fuzzy cluster (Definition 5.2)
$\frac{c}{2\cdot\mathbf{i}_r\cdot K^2}$	negligible support (Definition 6.2)

Fuzzy (cont.)

$\phi_X^{(r)}(C,P)$	r-fuzzy K-means cost of X with respect to means C and soft clustering P (Definition 5.1)
$\phi_X^{(r)}(C)$	r-fuzzy K-means cost of X of the solution induced by C (Notation 5.3)
$\phi_X^{(r)}(P)$	r-fuzzy K-means cost of X of the solution induced by P (Notation 5.3)
$A_k^{(X,r(P))}$	$=((x_n, r(p_{nk})\cdot w_n))_{n\in[N]}$ is the k-th r-fuzzy cluster of X given by P (Definition 5.2)
$\mathbf{d}\left(A_k^{(X,r(P))}\right)$	r-fuzzy K-means cost of the k-th cluster (Definition 5.1)
$\phi_{(X,K,r)}^{OPT}$	minimum r-fuzzy K-means cost of X (Definition 5.6)
$\mathrm{km}_X(C)$	K-means cost of X with respect to K means C (Problem 4.3)
$\mathrm{km}_{(X,K)}^{OPT}$	minimum K-means cost of X (Problem 4.3)

Discretization

$\mathtt{dist}(x,C)$	$=\min\left\{\left\|x-\mu\right\|_2 \mid \mu\in C\right\}$ for $C\subseteq\mathbb{R}^D$ and $x\in\mathbb{R}^D$ (Definition 9.1)
α-approx.	a K-clustering whose cost are at most a factor α worse than the cost of an optimal K-clustering
(α,β)-approx.	a $\lfloor\beta K\rfloor$-clustering whose cost are at most a factor α worse than the cost of an optimal K-clustering (Definition 9.3)
$\mathtt{B}(\mu,r)$	closed ball around μ with radius r (Definition 9.2)
$\mathfrak{Descr}$	contains triples $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ describing search spaces (Definition 9.5)
$\mathfrak{M}=(\mathfrak{m}_l)_{l\in[L]}$	a vector of L means
$2^{\mathfrak{E}}\cdot\mathfrak{R}$	radius of closed balls around means from $\mathfrak{M}$
$\mathfrak{U}(\mathfrak{E},\mathfrak{R},\mathfrak{M})$	$\subset\mathbb{R}^D$ is a search space (Definition 9.5)
$\mathfrak{U}_{l,j}$	(l,j)-th ring around mean $\mathfrak{m}_l$ (Definition 9.6)
$\mathfrak{g}(x)\in\mathfrak{G}$	representative of $x\in\mathfrak{U}$ (Definition 9.7)
$\mathfrak{G}=\mathfrak{g}(\mathfrak{U})$	discrete search space; set of all representatives (Definition 9.9)

Special Sets of Solutions

$A^{\leq M}$	$=\cup_{l\in[L]}\left\{(\mu_1,\ldots,\mu_l)\,\middle|\,\forall k\in[l]:\mu_k\in M\right\}$ for $M\subseteq\mathbb{R}^D$ (Notation 10.1)
$\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$	$=\{\,C\in(\mathbb{R}^D)^{\leq L}\mid C$ induces some r-fuzzy clustering that has no $(\mathbf{i}_r,K,\epsilon)$-negligible clusters $\}$ (Definition 12.6)

Density & Likelihood

$\mathcal{N}_D(\mu,\Sigma)$	D-variate Gaussian with mean μ and covariance Σ (Definition 14.1)
θ	parameters of a parameterized density function
θ^{old}	input of an EM (SEM) update step (Algorithm 16, Algorithm 17)
$\mathcal{L}_X(\theta)$	$=\mathrm{p}(X\mid\theta)$ likelihood of θ given observations X (Definition 14.4)
$\Lambda_X(\theta_1,\theta_2)$	$=\mathcal{L}_X(\theta_1)/\mathcal{L}_X(\theta_2)$; likelihood ratio (Definition 14.5)
$\mathcal{H}(q)$	entropy (Definition 17.1)
$\mathrm{KLD}(p\|q)(q)$	relative entropy; Kullback-Leibler divergence (Definition 17.1)

Soft K-Means

$\check{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((\mu_k)_k)$	$=-\ln(\mathcal{L}_X(\theta))$ where GMM $\theta=((\omega_k,\mu_k,\frac{2}{\beta}I_D)_{k\in[K]}$ and $\omega=(\omega_k)_{k\in[K]}$ (Problem 17.4)
$\mathrm{skm}_X^{(\beta,\omega)}(C,P)$	generalized soft K-means cost (Problem 17.9)
$\bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}(P)$	$=\mathrm{skm}_X^{(\beta,\omega)}((\mathbf{m}(A_k^{(X,P)}))_k,P)$ (Problem 17.7)

EM* and SEM* Algorithm

W_k, M_k, S_k	parameter updates by SEM* algorithm (Algorithm 21)
w_k, μ_k, Σ_k	parameter updates by EM* algorithm (Algorithm 20)
ζ_{nk}	a constant depending on the given data set X, the initial model θ^{old}, and the indices n, k (Algorithm 20)
a_δ	$= \sqrt{3\ln(2/\delta)}$ (Theorem 15.2)
b_δ	$= \sqrt{2e\ln(2/\delta)}$ (Theorem 15.2)
GMM	Gaussian mixture model
LMM	Laplacian mixture model

"Stell' dir vor es geht und keiner kriegt's hin."

Wolfgang Neuss[1]

Chapter 1

Preface

The term clustering refers to the task of dividing a set of objects into groups (clusters) such that objects that are in the same group are more similar to each other than objects that belong to different groups. This task arises in a wide variety of fields such as image analysis, bioinformatics, data compression, and pattern recognition. However, there is no algorithm that is universally right for all of the concrete clustering problems that arise in these fields. Consequently, there is a vast number of concrete clustering problems and algorithms. Finding the appropriate formulation for the problem at hand is a crucial question in practise. Nonetheless, certain clustering problems and algorithms have become very popular.

One can roughly divide the existing clustering methods into two classes: hard clustering methods and soft clustering methods. Hard clustering methods assign each object to exactly one cluster. An exceptionally popular representative of this class is known as K-means clustering. In recent years, the K-means clustering problem and various variants thereof have been studied in detail. In contrast, relatively little is known about soft clustering methods. With these methods, each object is not necessarily assigned to exactly one cluster, but to several clusters to a certain degree. In this thesis, we focus on two soft clustering problems: the fuzzy K-means problem and the maximum likelihood estimation problem with respect to Gaussian mixture models. Both are related to the K-means problem, but there are hardly any algorithms for these problems with performance guarantees.

1.1 Outline

This thesis is divided into three parts:

Part I deals with soft clustering in general. In Chapter 2, we introduce the basic notation and definitions that we will use throughout this whole thesis. In Chapter 3, we present a very general technique that helps us to relate soft clusterings to hard clusterings. We apply this technique in both subsequent parts of this thesis.

Part II focuses on the fuzzy K-means problem. We present the first approximation algorithms for this problem. The runtimes of these algorithms are similar to some of the best approximation algorithms for the classical K-means problem.

Part III deals with the maximum likelihood estimation (MLE) problem with respect to Gaussian mixture models. We consider three different topics in this part of the thesis: First, in Chapter 15, we compare two existing heuristics with one another, namely the classical expectation-maximization algorithm and a stochastic variant thereof. Second, in Chapter 16, we tackle the initialization problem of the expectation-maximization algorithm for Gaussian mixture models. Third, in Chapter 17, we

[1] Source: DIE ZEIT, 12.11.2009 Nr. 47

initiate the theoretical analysis of approximation algorithms for a constrained version of the MLE problem, which is known as the soft K-means problem.

Last but not least, we point out that there is an overview of our notation (Cheat Sheet), which can be found after the table of contents.

1.2 Publications & Credits

Parts of this thesis were obtained in cooperation with my coauthors and published in:

Blömer et al. (2014): Blömer, J., Bujna, K., and Kuntze, D. (2014).
A Theoretical and Experimental Comparison of the EM and SEM Algorithm.
In 22nd International Conference on Pattern Recognition (ICPR 2014), pages 1419 – 1424, Stockholm, Sweden. IEEE.

Blömer and Bujna (2016): Blömer, J. and Bujna, K. (2016).
Adaptive Seeding for Gaussian Mixture Models.
In Proceedings of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2016), volume 9652 of Lecture Notes in Computer Science, pages 296 – 308, Auckland, New Zealand. Springer.

Blömer et al. (2016): Blömer, J., Brauer, S., and Bujna, K. (2016).
A Theoretical Analysis of the Fuzzy K-Means Problem.
In IEEE 16th International Conference on Data Mining (ICDM 2016), pages 805 – 810, Barcelona, Spain. IEEE.

Blömer et al. (2017): Blömer, J., Brauer, S., and Bujna, K. (2017).
On Coreset Constructions for the Fuzzy K-Means Problem.
In Computing Research Repository (abs/1612.07516).

As usual in computer science, the authors are listed lexicographically.

Part I

Soft Clusterings

"All models are wrong, but some are useful."

Box & Draper[1]

Chapter 2

Basics

The goal of clustering is to group a given set of objects into clusters such that objects that belong to the same cluster are more similar to one another than objects that belong to different clusters. There are various ways to interpret and define objects, their similarity, the notion of clusters, and the quality of a clustering. Typically, objects consist of a fixed number of scalar features, which means that they can be interpreted as vectors from the real space. Besides that, we can roughly classify clusterings either as soft or hard clusterings: In a hard clustering, each object is allocated to exactly one cluster. In other words, a hard clustering partitions the given set of objects into (hard) clusters. A soft clustering allows each object to belong to multiple clusters to a certain degree. Hence, the (soft) clusters of a soft clustering are not subsets of the given set of objects. Instead, each soft cluster defines a degree of membership for each object. In this chapter, we formalize these notions and introduce definitions that we will use throughout this whole thesis.

Overview. In Section 2.1, we explain our basic notation and, in particular, our notation of data sets. In Section 2.2, we formalize our notion of soft clusterings, hard clusterings, and clustering problems. Finally, in Section 2.3, we define statistics that we will use to describe data sets as well as clusters, and derive useful properties of these statistics.

2.1 Notation: Indices, Vectors, Data Sets

Let us start with the following handy notation:

Notation 2.1 ($\{1,\dots,N\}$)**.** *For all $N \in \mathbb{N}$, we let*

$$[N] := \{1,2,\dots,N\} \ .$$

In mathematics, a finite ordered list of elements is usually referred to as a tuple. We stick to the term "vector" to stress the fact that the elements have the same type.

Notation 2.2 (tuples, vectors, matrices)**.** *Let $v_1, v_2, \dots, v_N$ be some elements from the same domain $\mathbb{A}$ (e.g. $\mathbb{R}$, $\mathbb{R}^D$, or $\mathbb{R}^D \times \mathbb{R}$). Given a finite set of indices $S = \{i_1,\dots,i_M\} \subseteq \mathbb{N}$, where $i_{\pi(1)} \leq i_{\pi(2)} \leq \dots \leq i_{\pi(M)}$ for some permutation π of $[M]$, we call the tuple*

$$(v_s)_{s\in S} := \left(v_{i_{\pi(1)}}, v_{i_{\pi(2)}}, \dots, v_{i_{\pi(M)}}\right)$$

a vector. We write $(v_s)_{s\in S} \subseteq \mathbb{A}$ or $(v_s)_{s\in S} \in \mathbb{A}^{|S|}$ to indicate that $v_s \in \mathbb{A}$ for all $s \in S$.

For the sake of simplicity, we identify $x = (x_1\, x_2 \dots x_D)^T \in \mathbb{R}^D$ with $(x_d)_{d\in[D]}$. To avoid confusion, we denote the d-th coordinate of a vector $x_n \in \mathbb{R}^D$ by $(x_n)_d$.

For matrices, we use the standard notation $M = \left(m_{ij}\right)_{i=1,\dots,L;j=1,\dots,N} = \left(m_{ij}\right)_{i\in[L],j\in[N]}$ for an $(L \times N)$-matrix M. We denote the (i,j)-th entry of M by $(M)_{ij} = m_{ij}$.

[1] Source: Empirical model-building and response surfaces, Wiley, 1987, p. 424.

The objects that we want to cluster are numerical feature vectors of fixed length. That is, objects are vectors $x_1,\dots,x_N \in \mathbb{R}^D$, where $D \in \mathbb{N}$ is some fixed value. Additionally, each object x_n has a certain weight $w_n \in \mathbb{R}_{\geq 0}$ which determines the importance of this object. We refer to the tuple (x_n, w_n) as a data point.

We think of a data set as a finite collection of data points. In principle, the arrangement of the data points in the data set is not important in this thesis. Even so, to keep our notation uncluttered, we think of a data set as an ordered list rather than a multi-set.

Definition 2.3 (data sets)**.** *Let $\mathbb{A} \subseteq \mathbb{R}$, $\mathbb{B} \subseteq \mathbb{R}_{\geq 0}$, and $D, N \in \mathbb{N}$. A D-dimensional data set X with N points from $\mathbb{A}^D$ and weights in $\mathbb{B}$ takes the form*

$$X = ((x_n, w_n))_{n\in[N]} \text{ with } x_n \in \mathbb{A}^D \text{ and } w_n \in \mathbb{B} \text{ for all } n \in [N] .$$

$\mathrm{Dom}(\mathbb{A}^D, \mathbb{B})$ *denotes the set of all data sets that contain a finite number of data points with points in $\mathbb{A}^D$ and weights in $\mathbb{B}$.*

A data set $X = ((x_n, w_n))_{n\in[N]}$ is unweighted *if $\forall n \in [N] : \ w_n = 1$. We identify an unweighted data set $V = ((v_m, 1))_{m\in[M]} \in \mathrm{Dom}(\mathbb{R}^D, \{1\})$ with the vector $V = (v_m)_{m\in[M]} \subseteq \mathbb{R}^D$.*

Our notation might seem a bit cumbersome but it will spare us from defining mappings between sets (which are now inherently given by indices) and from pointing out that we work with multi-sets. However, to get the best of both worlds, we have to introduce the following multi-set-like notation:

Notation 2.4 (multi-set-like notation)**.** *For all data sets $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}(\mathbb{R}^D, \mathbb{R}_{\geq 0})$ and data points $(x, w) \in \mathbb{R}^D \times \mathbb{R}_{\geq 0}$, we let*

$$count((x, w), X) := |\{n \in [N] \mid x_n = x \ \wedge \ w_n = w\}| .$$

We say that X contains $count((x, w), X)$ copies *of (x, w).*

For all data sets $X, Y \in \mathrm{Dom}(\mathbb{R}^D, \mathbb{R}_{\geq 0})$, we use the following notation:

- *D is the dimension of X (and of the data points in X).*
- *$(x, w) \in X$ indicates that $count((x, w), X) \geq 1$.*
- *$|X| := \sum_{(x,w)\in X} count((x, w), X)$ is the size (or length) of X. We write $X = \emptyset$ if $|X| = 0$.*
- *We write $X \subseteq Y$ if, for each each $(x, w) \in X$, we have $count(Y, (x, w)) \geq count(X, (x, w))$. Consequently, we write $X = Y$ if $X \subseteq Y$ and $Y \subseteq X$. We write $X \subset Y$ if, in addition to $X \subseteq Y$, there is some $(x, w) \in X$ where $count(Y, (x, w)) > count(X, (x, w))$.*
- *$X \dot\cup Y$ denotes a data set $Z \in \mathrm{Dom}(\mathbb{R}^D, \mathbb{R}_{\geq 0})$ where*

  $$count((x, w), Z) = count((x, w), X) + count((x, w), Y)$$

  for all $x \in \mathbb{R}^D$ and $w \in \mathbb{R}_{\geq 0}$.

  Let $K \in \mathbb{N}$ and $A_1,\dots,A_K \subseteq Z$. For all $l \in \{2,\dots,K\}$, we let $\dot\bigcup_{k=1}^{l} A_k = A_l \dot\cup \left(\dot\bigcup_{k=1}^{l-1} A_k\right)$. If $\dot\bigcup_{k=1}^{K} A_k = Z$, then we call $A_1,\dots,A_K$ a partition of Z. If $\dot\bigcup_{k=1}^{K} A_k \subseteq Z$, then we call $A_1,\dots,A_K$ pairwise disjoint subsets of Z.
- *$X \cap Y$ is a data set $Z \in \mathrm{Dom}(\mathbb{R}^D, \mathbb{R}_{\geq 0})$ where*

  $$count((x, w), Z) = \min\{count((x, w), X),\ count((x, w), Y)\}$$

  for all $x \in \mathbb{R}^D$ and $w \in \mathbb{R}_{\geq 0}$.

For the sake of simplicity, we use this notation also for vectors: Recall from Definition 2.3 that we identify a vector $(v_1,\dots,v_M) \subseteq \mathbb{R}^D$ with an unweighted data set $((v_m, 1))_{m\in[M]} \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$. When we use our multi-set-like notation with respect to vectors, we refer to the notation with respect to the corresponding unweighted data sets.

2.2 Clusterings

In the following, we formalize our notion of soft and hard clusterings and describe the form of the clustering tasks that we consider in this thesis.

2.2.1 Soft Clustering

In a soft clustering, each data point belongs to each cluster, but only with a certain probability.

Definition 2.5 (soft K-clustering)**.** *Let $K, N \in \mathbb{N}$. A soft-assignment of an object to K clusters is a discrete probability distribution over K classes. A soft K-clustering of N objects specifies a particular soft-assignment (to K clusters) for each object.*

We use the following notation to describe a soft clustering.

Definition 2.6. *For each $K \in \mathbb{N}$, we denote the (closed standard) $(K-1)$-simplex by*

$$\Delta_{K-1} := \left\{ (p_k)_{k\in[K]} \in [0,1]^K \;\middle|\; \sum_{k=1}^{K} p_k = 1 \right\}.$$

For all $K, N \in \mathbb{N}$, we let

$$\Delta_{N,K-1} := \left\{ (p_{nk})_{n\in[N],k\in[K]} \in [0,1]^{N\times K} \;\middle|\; \forall n \in [N] \colon (p_{nk})_{k\in[K]} \in \Delta_{K-1} \right\}.$$

Notation 2.7 (soft K-clustering)**.** *We denote a soft-assignment over K clusters by a distribution $(p_k)_{k\in[K]} \in \Delta_{K-1}$. We describe a soft K-clustering of N objects by a matrix $P = (p_{nk})_{n\in[N],k\in[K]} \in \Delta_{N,K-1}$ where $(p_{nk})_{k\in[K]}$ is the soft-assignment of the n-th object.*

Given a soft clustering $(p_{nk})_{n\in[N],k\in[K]}$ of the N data points of $X = ((x_n, w_n))_{n\in[N]}$, we say that (x_n, w_n) is assigned to the k-th cluster with probability p_{nk}.

We think of the k-th soft cluster of $X = ((x_n, w_n))_{n\in[N]}$ given by $P = (p_{nk})_{n,k} \in \Delta_{N,K-1}$ as the data set

$$A_k^{(X,P)} := ((x_n, w_n \cdot p_{nk}))_{n\in[N]}. \tag{2.1}$$

That is, if (x_n, w_n) is assigned to the k-th cluster with probability p_{nk}, then its importance (weight) in the k-th cluster is reduced to $p_{nk} \cdot w_n$. Observe that the total importance of data point (x_n, w_n) in the soft clustering is still given by its weight: $\sum_{k=1}^{K}(p_{nk} \cdot w_n) = w_n$.

2.2.2 Hard Clustering

In a hard clustering, each data point belongs to exactly one cluster. Hence, a hard clustering is actually a special case of a soft clustering.

Definition 2.8 (hard K-clustering)**.** *Let $K, N \in \mathbb{N}$. A hard-assignment of an object to K clusters is simply given by a single value in $[K]$. A hard K-clustering of N objects defines a particular hard-assignment to K clusters for each of the N objects.*

We use indicator vectors to formally describe a hard clustering.

Notation 2.9 (indicator)**.** *We refer to a vector $(z_k)_{k\in[K]} \in \{0,1\}^K$ as an indicator vector and to a matrix $(z_{nk})_{n\in[N],k\in[K]} \in \{0,1\}^{N\times K}$ as an indicator matrix. We say that the indicator matrix assigns the n-th object to the k-th cluster if $z_{nk} = 1$. Likewise, we say the indicator vector indicates an assignment to the k-th cluster if $z_k = 1$.*

Note that indicator vectors and matrices might assign an object to more than one cluster or to no cluster at all.

Notation 2.10 (hard K-clustering)**.** *Let $X = ((x_n, w_n))_{n\in[N]}$ be a data set. We describe a hard K-clustering of X by an indicator matrix $Z = (z_{nk})_{n\in[N],k\in[K]} \in \{0,1\}^{N\times K}$ where*

$$\forall n \in [N] : \sum_{k=1}^{K} z_{nk} = 1 \ .$$

We think of the k-th hard cluster of $X = ((x_n, w_n))_{n\in[N]}$ given by the hard clustering $Z = (z_{nk})_{n\in[N],k\in[K]}$ as

$$A_k^{(X,Z)} = ((x_n, w_n \cdot z_{nk}))_{n\in[N]} \ , \tag{2.2}$$

which is a re-weighted version of the original data set, or as

$$A_k = ((x_n, w_n))_{n\in\{m\in[N] \mid z_{mk}=1\}} \subseteq X \ . \tag{2.3}$$

Moreover, we denote data sets $A \subseteq X$ also as hard clusters of X.

If $Z \in \{0,1\}^{N\times K}$ describes a hard K-clustering with clusters $A_1, \dots, A_K \subseteq X$, then we have $\dot\cup_{k\in[K]} A_k = X$. Moreover, hard clusterings are special kinds of soft clusterings: For each indicator matrix $H \in \{0,1\}^{N\times K}$ that describes a hard clustering, we have $H \in \Delta_{N,K-1}$.

2.2.3 Clustering Problems

We stress the fact that we only consider clustering problems where we search for a clustering with a *predefined* number of clusters K. We consider two types of such clustering tasks: First, in Part II, we consider the fuzzy K-means problem, which belongs to the class of representative-based clustering problems.

Problem 2.11 (Representative-Based)**.** *We are given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, a number of clusters $K \in \mathbb{N}$, a set of representatives $\Theta \subseteq \mathbb{R}^D$, a set of clusterings $\Delta \subseteq \Delta_{N,K-1}$, and a function*

$$\Phi : \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right) \times \Theta^K \times \Delta \to \mathbb{R} \ .$$

Find representatives $\theta_1, \dots, \theta_K \in \Theta$ and a clustering $P \in \Delta$ minimizing $\Phi\left(X, (\theta_k)_{k\in[K]}, P\right)$.

Note that each representative is from the same domain as the points in the given data set (i.e., $\theta_k, x_n \in \mathbb{R}^D$). That is, the k-th representative can be thought of as the prototype of the points in the k-th cluster.

Second, in Part III, we consider a model-based clustering problem, namely the maximum likelihood estimation problem for Gaussian mixture models.

Problem 2.12 (Model-Based Clustering)**.** *We are given a data set X, a set of generative models $\Theta_{(K)}$, and a function*

$$\Phi : \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right) \times \Theta_{(K)} \to \mathbb{R} \ .$$

Find a model $\theta \in \Theta_{(K)}$ that minimizes $\Phi(X, \theta)$.

In this problem formulation, we do not not explicitly mention a soft clustering. We assume that the generative models $\theta \in \Theta_{(K)}$, which describe the generation of data sets, implicitly depend on a number (of clusters) K. In Part III, we show that there is an obvious way to derive a soft clustering $P \in \Delta_{N,K-1}$ for a given data set X and a generative model $\theta \in \Theta_{(K)}$, with respect to the maximum likelihood estimation problem for Gaussian mixture models.

2.3 Descriptive Statistics

Next, we define statistics that describe data sets and clusters.

2.3.1 Cluster Statistics

Recall from (2.1) and (2.2) that hard clusters and soft clusters can be thought of as reweighted versions of the given data set. In this thesis, we use the following statistics to describe clusters in particular.

Definition 2.13 (weight)**.** *The weight of* $A = ((x_n, \alpha_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ *is*

$$\mathbf{w}(A) \coloneqq \sum_{n=1}^{N} \alpha_n \ .$$

Observe that $\mathbf{w}(A) \geq 0$ for all $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$.

Definition 2.14 (mean)**.** *Let* $A = ((x_n, \alpha_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$*. If* $\mathbf{w}(A) > 0$*, then the mean of* A *is given by*

$$\mathbf{m}(A) \coloneqq \frac{\sum_{n=1}^{N} \alpha_n x_n}{\mathbf{w}(A)} \ .$$

Otherwise, we let $\mathbf{m}(A) \coloneqq 0_D$ *be the zero vector.*

Definition 2.15 (covariance)**.** *Let* $A = ((x_n, \alpha_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ *and* $z \in \mathbb{R}^D$*. The unnormalized covariance of* A *with respect to* z *is given by*

$$\mathbf{ucov}(A, z) \coloneqq \sum_{n=1}^{N} \alpha_n (x_n - z)(x_n - z)^T \ .$$

In particular, we let

$$\mathbf{ucov}(A) \coloneqq \mathbf{ucov}(A, \mathbf{m}(A)) \ .$$

If $\mathbf{w}(A) > 0$*, then the covariance of* A *is*

$$\mathbf{cov}(A) \coloneqq \frac{\mathbf{ucov}(A)}{\mathbf{w}(A)} \ .$$

Otherwise, we let $\mathbf{cov}(A) \coloneqq 0_{D,D}$ *be the zero matrix.*

Definition 2.16 (cost, variance)**.** *Let* $A = ((x_n, \alpha_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ *and* $z \in \mathbb{R}^D$*. The cost of* A *with respect to* z *is the sum of squared Euclidean distances*

$$\mathbf{d}(A, z) \coloneqq \sum_{n=1}^{N} \alpha_n \left\| x_n - z \right\|_2^2 \ .$$

The cost of A *is*

$$\mathbf{d}(A) \coloneqq \mathbf{d}(A, \mathbf{m}(A)) \ .$$

The variance of A *is given by*

$$\mathbf{var}(A) \coloneqq \frac{\mathbf{d}(A)}{\mathbf{w}(A)}$$

if $\mathbf{w}(A) > 0$*. Otherwise, we let* $\mathbf{var}(A) \coloneqq 0$*.*

For all $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ with $\mathbf{w}(A) = 0$, we have $\mathbf{d}(A) = 0$ and $\mathbf{ucov}(A) = 0$.

"Sample" Statistics. The quantities that we call the mean, variance, and covariance of A are also denoted as the sample mean, sample variance (in case $D = 1$), and sample covariance (in case $D > 1$), respectively (cf. Bishop, 2006, p. 27). This is because the data points in A are assumed to be the outcomes (samples) of independently and identically distributed random variables. Under this assumption, our statistics are (biased) estimates of the statistics of a (presumed) underlying distribution. In Section 14.2.7, we formalize this idea.

Variance. The variance $\mathbf{var}(A)$ of a data set $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ with $D > 1$ should not be confused with the sample variance of a set of real-valued samples or with the variance of a single real-valued random variable. However, it can be interpreted as a sum of variances:

Lemma 2.17. *For all $A = ((x_n, \alpha_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, we have*

$$\mathbf{var}(A) = \mathrm{Tr}(\mathbf{cov}(A)) \ .$$

Proof. If $\mathbf{w}(A) > 0$, then $\mathrm{Tr}(\mathbf{cov}(A)) = \sum_{d=1}^{D} (\mathbf{cov}(A))_{dd} = \frac{1}{\mathbf{w}(A)} \sum_{d=1}^{D} \left(\sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A))_d^2\right) = \frac{1}{\mathbf{w}(A)} \left(\sum_{n=1}^{N} \alpha_n \sum_{d=1}^{D} (x_n - \mathbf{m}(A))_d^2\right) = \mathbf{var}(A)$. If $\mathbf{w}(A) = 0$, then $\mathrm{Tr}(\mathbf{cov}(A)) = 0 = \mathbf{var}(A)$ □

Hence, the variance $\mathbf{var}(A)$ can be thought of as the *sum* of the variances of the one-dimensional (sample) data sets $(((x_n)_d, w_n))_{n \in [N]}$ with $d \in [D]$.

Consistency of Hard Cluster Notions. Recall from Notation 2.10 that we identify hard clusters $A_k \subseteq X$ with data sets $A_k^{(X,Z)}$ for appropriately chosen hard clusterings Z. Observe that the statistics of A_k coincide with the statistics of these data sets $A_k^{(X,Z)}$.

2.3.2 Data Set Statistics

The following statistics will be useful to describe the given data set X, in particular.

Definition 2.18 (statistics of a data set)**.** *Let $X = ((x_n, w_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$.*

The range of X in the d-th coordinate is given by

$$\mathtt{r}_d(X) := \max\{(x_n)_d - (x_m)_d \mid n, m \in [N]\} \ .$$

The diameter of X is given by

$$\mathtt{diam}(X) := \max\{\|x_n - x_m\|_2 \mid n, m \in [N]\} \ .$$

The minimum and maximum weight of a data point in X are

$$\mathtt{w}_{\min}^{(X)} := \min\{w_n \mid n \in [N]\} \quad \textit{and} \quad \mathtt{w}_{\max}^{(X)} := \max\{w_n \mid n \in [N]\} \ .$$

We describe the "extent" of the data set by the maximum Euclidean distance between points (the diameter) or the maximum absolute difference between points per dimension (the range). Observe that $\mathtt{r}_d(X) \geq 0$ and that $\mathtt{diam}(X)^2 \leq \sum_{d=1}^{D} \mathtt{r}_d(X)^2$.

To describe the way the data points are scattered in the input domain $\mathbb{R}^D$, we use the covariance $\mathbf{cov}(X)$ of the given data set X, which we defined in Definition 2.15.

2.3.3 Lemmata

In this section, we derive alternative formulations of the cost $\mathbf{d}(A)$ and the covariance $\mathbf{cov}(A)$. The mean $\mathbf{m}(A)$ is exceptionally useful to understand both statistics.

Observation 2.19 (zero)**.** *For all* $A = ((x_n, \alpha_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$*, we have*

$$\sum_{n=1}^{N} \alpha_n \left(x_n - \mathbf{m}(A)\right) = \left(\sum_{n=1}^{N} \alpha_n x_n\right) - \mathbf{w}(A) \cdot \mathbf{m}(A) = 0_D \quad \textit{and}$$

$$\sum_{n=1}^{N} \alpha_n \left((x_n - \mathbf{m}(A))(x_n - \mathbf{m}(A))^T - \mathbf{cov}(A)\right) = \mathbf{ucov}(A) - \mathbf{w}(A)\mathbf{cov}(A) = 0_{D,D} \ ,$$

where 0_D *denotes the* D*-dimensional zero vector and* $0_{D,D}$ *denotes the zero* $(D \times D)$*-matrix.*

The following lemma is well known (Inaba et al., 1994, proof of Theorem 2).

Lemma 2.20. *For every* $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ *and* $z \in \mathbb{R}^D$*, we have*

$$\mathbf{d}(A, z) = \mathbf{d}(A) + \mathbf{w}(A) \cdot \|z - \mathbf{m}(A)\|_2^2 \ .$$

Proof. Let $A = ((x_n, \alpha_n))_{n\in[N]}$. Observe that

$$\begin{aligned}
\mathbf{d}(A, z) &= \sum_{n=1}^{N} \alpha_n \|x_n - z\|_2^2 \\
&= \sum_{n=1}^{N} \alpha_n \|x_n - \mathbf{m}(A) + \mathbf{m}(A) - z\|_2^2 \\
&= \sum_{n=1}^{N} \alpha_n \langle x_n - \mathbf{m}(A) + \mathbf{m}(A) - z,\ x_n - \mathbf{m}(A) + \mathbf{m}(A) - z\rangle \\
&= \sum_{n=1}^{N} \alpha_n \left(\|x_n - \mathbf{m}(A)\|_2^2 + 2\langle x_n - \mathbf{m}(A),\ \mathbf{m}(A) - z\rangle + \|\mathbf{m}(A) - z\|_2^2\right) \\
&= \mathbf{d}(A) + 2\sum_{n=1}^{N} \alpha_n \langle x_n - \mathbf{m}(A),\ \mathbf{m}(A) - z\rangle + \mathbf{w}(A) \cdot \|\mathbf{m}(A) - z\|_2^2 \ .
\end{aligned}$$

Due to Observation 2.19, the second summand computes to

$$\sum_{n=1}^{N} \alpha_n \langle x_n - \mathbf{m}(A),\ \mathbf{m}(A) - z\rangle = \left\langle \sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A)),\ \mathbf{m}(A) - z \right\rangle = 0 \ .$$

This yields the claim. □

In particular, this lemma implies that $z = \mathbf{m}(A)$ is the vector minimizing the cost $\mathbf{d}(A, z)$. For the unnormalized covariance, we obtain an analogous result:

Lemma 2.21. *For every* $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ *and* $z \in \mathbb{R}^D$*, we have*

$$\mathbf{ucov}(A, z) = \mathbf{ucov}(A) + \mathbf{w}(A) \cdot (\mathbf{m}(A) - z)(\mathbf{m}(A) - z)^T \ .$$

Proof. The proof is an analogon of Lemma 2.20 where we replace inner products by corresponding outer products, which are bi-linear as well. Write $A = ((x_n, \alpha_n))_{n\in[N]}$. We have

$$\begin{aligned}
\mathbf{ucov}(A, z) &= \sum_{n=1}^{N} \alpha_n (x_n - z)(x_n - z)^T \\
&= \sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A) + \mathbf{m}(A) - z)(x_n - \mathbf{m}(A) + \mathbf{m}(A) - z)^T \\
&= \mathbf{ucov}(A) + \sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A))(\mathbf{m}(A) - z)^T \\
&\quad + \left(\sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A))(\mathbf{m}(A) - z)^T\right)^T + \mathbf{w}(A) \cdot (\mathbf{m}(A) - z)(\mathbf{m}(A) - z)^T \ ,
\end{aligned}$$

where, due to Observation 2.19, we have

$$\sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A))(\mathbf{m}(A) - z)^T = \left(\sum_{n=1}^{N} \alpha_n (x_n - \mathbf{m}(A)) \right) (\mathbf{m}(A) - z)^T = 0_{D,D} \ .$$

This yields the claim. □

With our definitions, we directly obtain the following corollary.

Corollary 2.22. *For every $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ with $\mathbf{w}(A) > 0$ and $z \in \mathbb{R}^D$, we have*

$$\mathbf{cov}(A, z) = \mathbf{cov}(A) + (\mathbf{m}(A) - z)(\mathbf{m}(A) - z)^T \ .$$

The following lemmata express the cost $\mathbf{d}(A)$ and the unnormalized covariance $\mathbf{ucov}(A)$ of a cluster without explicitly using the mean $\mathbf{m}(A)$.

Corollary 2.23. *For every $A = ((x_n, \alpha_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ with $\mathbf{w}(A) > 0$, we have*

$$\mathbf{d}(A) = \frac{\sum_{n=1}^{N} \sum_{m<n} \alpha_n \alpha_m \|x_n - x_m\|_2^2}{\mathbf{w}(A)} \ .$$

Proof. Observe that

$$\begin{aligned}
2 \sum_{n=1}^{N} \sum_{m<n} \alpha_m \alpha_n \|x_n - x_m\|_2^2 &= \sum_{n=1}^{N} \sum_{n=1}^{N} \alpha_m \alpha_n \|x_n - x_m\|_2^2 \\
&= \sum_{n=1}^{N} \alpha_n \mathbf{d}(A, x_n) \\
&= \sum_{n=1}^{N} \alpha_n \left(\mathbf{d}(A) + \mathbf{w}(A) \|x_n - \mathbf{m}(A)\|_2^2 \right) && \text{(Lemma 2.20)} \\
&= \mathbf{w}(A)\mathbf{d}(A) + \mathbf{w}(A) \sum_{n=1}^{N} \alpha_n \|x_n - \mathbf{m}(A)\|_2^2 = 2\mathbf{w}(A)\mathbf{d}(A) \ .
\end{aligned}$$

This yields the claim. □

Corollary 2.24. *For every $A = ((x_n, \alpha_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ with $\mathbf{w}(A) > 0$, we have*

$$\mathbf{ucov}(A) = \frac{\sum_{n=1}^{N} \sum_{m<n} \alpha_n \alpha_m (x_n - x_m)(x_n - x_m)^T}{\mathbf{w}(A)} \ .$$

Proof. Analogously to Corollary 2.23, with the help of Lemma 2.21 instead of Lemma 2.20. □

2.3.4 Scaling Weights and Copying Data Points

Sometimes it is useful to manipulate the data set. First, consider scaling all weights by the same constant factor. We apply this trick in the proof of Theorem 3.21.

Lemma 2.25 (scaling weights)**.** *Let $A = ((x_n, \alpha_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ and $s \in \mathbb{R}_+$. For the re-weighted data set $A_s = ((x_n, s \cdot \alpha_n))_{n \in [N]}$, we have $\mathbf{w}(A_s) = s \cdot \mathbf{w}(A)$, $\mathbf{m}(A_s) = \mathbf{m}(A)$, $\mathbf{cov}(A_s) = \mathbf{cov}(A)$, and $\mathbf{d}(A_s) = s \cdot \mathbf{d}(A)$.*

Besides that, $|A_s| = |A|$, while $\mathrm{w}_{\max}^{(A_s)} = s \cdot \mathrm{w}_{\max}^{(A)}$ and $\mathrm{w}_{\min}^{(A_s)} = s \cdot \mathrm{w}_{\min}^{(A)}$.

Proof. By definition, we have $\mathbf{w}(A_s) = \sum_{n=1}^{N} s\alpha_n = s\sum_{n=1}^{N} \alpha_n = s \cdot \mathbf{w}(A)$ and hence

$$\mathbf{m}(A_s) = \frac{\sum_{n=1}^{N} s \cdot \alpha_n x_n}{\mathbf{w}(A_s)} = \frac{s\sum_{n=1}^{N} \alpha_n x_n}{s\mathbf{w}(A)} = \mathbf{m}(A_s) \ .$$

Hence, $\mathbf{d}(A_s) = \sum_{n=1}^{N} s \cdot \alpha_n \, \|x_n - \mathbf{m}(A_s)\|_2 = s \cdot \sum_{n=1}^{N} \alpha_n \, \|x_n - \mathbf{m}(A)\|_2 = s \cdot \mathbf{d}(A)$. Analogously, we obtain $\mathbf{ucov}(A_s) = s \cdot \mathbf{ucov}(A)$. Thus, $\mathbf{cov}(A_s) = \mathbf{ucov}(A_s)/\mathbf{w}(A_s) = s \cdot \mathbf{ucov}(A)/(s \cdot \mathbf{w}(A)) = \mathbf{cov}(A)$. □

Second, consider adding $(c-1)$ copies of each data point to the data set. We apply this manipulation in the proof of Corollary 8.8.

Corollary 2.26 (adding copies)**.** *Let $A \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ and $c \in \mathbb{N}$. For the data set A_c that contains c copies of each data point from A, we have $\mathbf{w}(A_c) = c \cdot \mathbf{w}(A)$, $\mathbf{m}(A_c) = \mathbf{m}(A)$, $\mathbf{cov}(A_c) = \mathbf{cov}(A)$, and $\mathbf{d}(A_c) = c \cdot \mathbf{d}(A)$.*

Besides that, $|A_c| = c \cdot |A|$, while $\mathtt{w}_{\max}^{(A_c)} = \mathtt{w}_{\max}^{(A)}$ and $\mathtt{w}_{\min}^{(A_c)} = \mathtt{w}_{\min}^{(A)}$.

The statistics change in the same way as they do when we scale the weights by a factor c. However, the resulting data set is c times larger than the original data set, while the maximum and minimum weight of a data point remain the same.

"There is nothing worse than a sharp image of a fuzzy concept."

Ansel Adams[1]

Chapter 3

From Soft Clusters to Hard Clusters

In contrast to soft clustering problems, hard clustering problems such as the K-means problem have been well studied from a theoretical point of view. Accordingly, there are numerous techniques to handle hard clustering problems. Unfortunately, most of these techniques are not directly applicable for soft clustering problems.

In this chapter, we aim to cover the gap between soft and hard clusterings via a Monte Carlo method: A soft clustering describes the assignment of data points to clusters via probability distributions. That is, in principle, the soft clustering itself is deterministic. However, we can make use of its inherent probabilistic interpretation. By simulating the random assignment of each point, we can construct hard clusters. In the following, we show that, with certain probability, the statistics of these hard clusters are similar to the statistics of the given soft clusters.

Overview. First, we give an overview of some related work in Section 3.1 and sum up our contribution in Section 3.2. In Section 3.3, we generalize and formalize our idea in terms of a randomized algorithm. In Section 3.4, we give an overview of various probabilistic bounds that can be used to analyse the algorithm and discuss their differences. In Section 3.5, we use them to analyse the algorithm. In Section 3.6, we draw some conclusions from this analysis: First, we prove the existence of hard clusters imitating a given soft clustering. Second, we analyse the quality of a single run of the randomized algorithm. Finally, in Section 3.6.3, we discuss a possible improvement and limits of these results.

Publications. In this chapter, we generalize and discuss two previously published results: the analysis of the stochastic expectation-maximization algorithm (SEM) for Gaussian mixture models (GMMs) presented in (Blömer et al., 2014) and the technique that helps to "relate fuzzy clusters to hard clusters" from (Blömer et al., 2016, Theorem 5).

3.1 Related Work

The term Monte Carlo technique refers to a broad class of inference methods that are based on random sampling. The main idea is to solve a problem, which might be deterministic in principle, by using randomness. For an introduction to this topic, we refer to (Bishop, 2006, pp. 523). There is a vast number of practical applications that make use of this technique, such as the stochastic expectation-maximization algorithm (SEM) for Gaussian mixture models (GMMs), which we consider in Chapter 15.

[1] Source: David Prutchi. Exploring Ultraviolet Photography, 2016.

3.2 Contribution

The main contribution of this chapter is a detailed analysis of an algorithm that helps to relate soft clusters to hard clusters by sampling hard assignments according to the probabilistic interpretation behind the given soft clustering. We derive probabilistic bounds on the similarity between the statistics of the resulting hard clusters and the statistics of the given soft clusters. We keep our analysis modular and compare the application of different concentration bounds. Later in this thesis, we show that these results are useful in three regards: First, in Chapter 8, we use them to derive an approximation algorithm for the fuzzy K-means problem. Second, in Chapter 15, we use them for an (non-asymptotic) comparison of the stochastic expectation-maximization (SEM) algorithm for Gaussian mixture models with the expectation-maximization (EM) for Gaussian mixture models. Third, in Chapter 17, we use them to analyse a variant of the so-called soft K-means problem.

3.3 Imitating Softness by Randomness

We want to imitate soft assignments by sampling hard assignments according to the distributions given by the soft assignments. Before we formalize this approach, let us introduce a more general class of soft assignments for which this approach works as well.

3.3.1 Probabilistic Memberships

The following generalization of soft clustering matrices and soft clusters is useful with regard to the fuzzy K-means problem, which we consider in Part II.

Definition 3.1 (probabilistic memberships)**.** *We call a matrix $(r_{nk})_{n\in[N],k\in[K]} \in [0,1]^{N\times K}$ a probabilistic membership matrix if*

$$\forall n \in [N] \colon \sum_{k=1}^{K} r_{nk} \leq 1 \,.$$

We call its single entries $r_{nk} \in [0,1]$ probabilistic membership values.

A probabilistic membership matrix describes a relaxed soft clustering where some elements are possibly not assigned to any cluster at all.

Definition 3.2 (clusters)**.** *Given a data set $X = ((x_n, w_n))_{n\in[N]}$, the membership matrix $R = (r_{nk})_{n\in[N],k\in[K]}$, and index $k \in [K]$, we let*

$$A_k^{(X,R)} := ((x_n, w_n \cdot r_{nk}))_{n\in[N]}$$

be the k-th soft cluster of X defined by the membership matrix R.

The n-th data point (x_n, w_n) is assigned to the k-th soft cluster with probability r_{nk}. Therefore, its weight (importance) in the k-th cluster is only $r_{nk} \cdot w_n$.

Observe that the soft clusters that are given by a probabilistic membership matrix $(r_{nk})_{n,k}$ do not necessarily form a soft clustering. They do not necessarily "cover" the whole data set: Possibly, we have $\sum_{k=1}^{K} r_{nk} < 1$ for some data point (x_n, w_n). Hence, the overall weight of all data points $\sum_{k=1}^{K} \mathbf{w}(A_k^{(X,R)})$ in the clusters might be less than the overall weight $\mathbf{w}(X)$ of the original data set.

Example 3.3 (fuzzy K-means)**.** *In the fuzzy K-means objective function, each soft assignment p_{nk} is exponentiated by some number $m \in (1,\infty)$. For all $(p_{nk})_{n,k} \in \Delta_{N,K-1}$ and $n \in [N]$, we have $\sum_{k=1}^{K} p_{nk}^m \leq \sum_{k=1}^{K} p_{nk} = 1$. Hence, the matrix $R = (p_{nk}^m)_{n,k}$ is a probabilistic membership matrix. It defines the clusters $A_k^{(X,R)} = \left((x_n, w_n \cdot p_{nk}^m)\right)_{n\in[N]}$ of $X = ((x_n, w_n))_{n\in[N]}$. In general, the overall weight of these clusters sums up to less than $\mathbf{w}(X)$.*

3.3.2 Algorithm

We are given a probabilistic membership matrix $R = (r_{nk})_{n,k}$ and a data set $X = ((x_n, w_n))_n$. R assigns data point (x_n, w_n) to the k-th cluster with *probability* r_{nk}. We apply this probabilistic interpretation as follows:

Algorithm 1 From Soft to Hard Clusters

Require: probabilistic membership matrix $(r_{nk})_{n\in[N],k\in[K]} \in [0,1]^{N\times K}$ and a data set $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$

1: Let $A_1, \dots, A_K$ be empty hard clusters.
2: **for all** $n \in [N]$ **do**
3: Sample $(Z_{nk})_{k\in[K]} \in \{0,1\}^K$ with $\sum_{k=1}^K Z_{nk} \in \{0,1\}$ according to $\Pr(Z_{nk} = 1 \wedge \forall l \in [K] \setminus \{k\} : Z_{nl} = 0) = r_{nk}$ for all $k \in [K]$ and $\Pr(\forall l \in [K] : Z_{nl} = 0) = 1 - \sum_{l=1}^K r_{nl}$.
4: Add (x_n, w_n) to the cluster A_k where $Z_{nk} = 1$.
5: **return** $((Z_{nk})_{n\in[N],k\in[K]}, (A_k)_{k\in[K]})$

3.4 Concentration Bounds

The statistics of the hard clusters constructed by Algorithm 1 can be interpreted as (matrices of) real-valued random variables. In this section, we present different ways to bound the probability that a real-valued random variable is far away from its expected value: In Section 3.4.1, we state and discuss Markov's and Chebyshev's inequality. Then, in Section 3.4.2, we derive some Chernoff-type inequalities. In the next Section 3.5, we apply these inequalities to analyse Algorithm 1.

To illustrate the differences between the inequalities, we use the following example.

Example 3.4. *Consider the number of heads in 100 fair coin flips. Formally, let F_i be the binary random variable that indicates whether the i-th flip is heads ($F_i = 1$). Then we know $\Pr(F_i = 1) = 1/2$, $\mathrm{E}[F_i] = 1/2$, and $\mathrm{Var}(F_i) = 1/4$. Let $X = \sum_{i=1}^{100} F_i$ be the total number of heads. We observe $\mathrm{E}[X] = 50$ heads in expectation.*

Intuitively, observing no or 100 heads is extremely unlikely. To be precise, the probability of this event computes to $\Pr(X = 0 \vee X = 100) = \Pr(X = 0) + \Pr(X = 100) = 2 \cdot (1/2)^{100} \approx 2 \cdot 10^{-30}$.

3.4.1 Elementary Inequalities

The most elementary bound is Markov's inequality.

Theorem 3.5 (Markov)**.** *Let X be a real-valued non-negative random variable. Then, for all $a \geq 1$, we have*

$$\Pr(X \geq a \cdot \mathrm{E}[X]) \leq \frac{1}{a}\,.$$

Proof. A proof can be found e.g. in (Mitzenmacher and Upfal, 2005, p. 44). □

Example 3.4 (continued)**.** *Markov's inequality says that the probability of observing 100 heads is at most $E[X]/100 = 0.5$.*

Apparently, this bound is often too weak to yield useful results. If not only the expected value $\mathrm{E}[X]$ but also the variance $\mathrm{Var}(X)$ is known, then Markov's inequality can be refined:

Corollary 3.6 (Chebyshev)**.** *Let X be a real-valued random variable with finite variance* $\mathrm{Var}(X)$*. Then, for all $a \geq 1$, we have*

$$\Pr\left(|X - \mathrm{E}[X]| \geq a \cdot \sqrt{\mathrm{Var}(X)}\right) \leq \frac{1}{a^2} .$$

Proof. A proof can be found e.g. in (Mitzenmacher and Upfal, 2005, p. 49). □

Example 3.4 (continued)**.** *Due to the independence of the coin flips, the variance of X computes to* $\mathrm{Var}(X) = \sum_{i=1}^{100} \mathrm{Var}(F_i) = 100 \cdot 1/4 = 25$*. Hence, Chebyshev's inequality says that the probability of observing no or 100 heads is at most* $(\sqrt{25}/50)^2 = 0.01$*.*

Note that Chebyshev's inequality provides a bound on the absolute difference $|X - \mathrm{E}[X]|$ (and not only on the difference $X - \mathrm{E}[X]$) and, as can be seen in the example, usually yields tighter bounds than Markov's inequality. The only drawback is that we need to know and be able to interpret the variance $\mathrm{Var}(X)$.

Overall, both inequalities presented in this section do not yield particularly tight bounds. In spite of this, in Section 3.5 and Section 3.6.1, we use these inequalities to show that there exist hard clusters that imitate given soft clusters.

3.4.2 Chernoff Inequalities

In comparison to the inequalities from the previous section, Chernoff bounds are exceptionally tight as they give exponentially decreasing bounds on the probability that an observation is far from its expected value (see Mitzenmacher and Upfal, 2005, p. 61ff). In the following, we focus on deriving Chernoff bounds for random variables that are sums of mutually independent random variables with bounded absolute value.

Using the Expected Value

The following Chernoff-type bounds measure the absolute difference between a random variable and its expected value in terms of the expected value itself.

Theorem 3.7. *Let $X_1, \ldots, X_n$ be mutually independent random variables in* $[0,1]$ *and let $Y = \sum_{i=1}^{n} X_i$. Then, for all $\lambda \in [0,1]$ we have*

$$\Pr(|Y - \mathrm{E}[Y]| \geq \lambda \cdot \mathrm{E}[Y]) \leq 2e^{-\mathrm{E}[Y]\frac{\lambda^2}{3}} .$$

Proof. A proof can be found e.g. in (McDiarmid, 1998, Thm. 2.3). □

Example 3.4 (continued)**.** *The Chernoff-type bound from Theorem 3.7 states that the probability of observing no or 100 heads is at most* $2 \cdot \exp(-50 \cdot 1/3) \approx 2 \cdot 10^{-7}$*.*

The following corollary explicitly states the maximum deviation that Theorem 3.7 guarantees for a given probability δ.

Corollary 3.8. *Let $X_1, \ldots, X_n$ be mutually independent random variables in* $[0,1]$ *and let $Y = \sum_{i=1}^{n} X_i$. Let $\delta \in (0,1)$. If we have*

$$\mathrm{E}[Y] \geq 3\ln(2/\delta) ,$$

then

$$\Pr\left(|Y - \mathrm{E}[Y]| \geq \sqrt{3\ln(2/\delta)}\sqrt{\mathrm{E}[Y]}\right) \leq \delta .$$

Proof. Due to $\mathrm{E}[Y] \geq 3\ln(2/\delta)$, we have $\lambda := \sqrt{3\ln(2/\delta)/E[Y]} \in [0,1]$. Applying Theorem 3.7 yields the claim. □

From this latter proof we see that Theorem 3.7 is actually only meaningful for random variables with a certain expected value.

Using the Variance

In the following, we derive Chernoff-type bounds that measure the absolute difference between a random variable and its expected value in terms of its variance. To this end, we use the following lemma by Levchenko (2013).

Lemma 3.9 (A Chernoff-Type Bound)**.** *Let $X_1,\dots,X_n$ be discrete and mutually independent random variables with $\mathrm{E}[X_i]=0$ and $|X_i|\le C$ for all $i\in[n]$ and some constant $C\ge 0$. Let $Y=\sum_{i=1}^n X_i$. Then, for all $t\ge 0$, it holds*

$$\Pr\left(|Y|\ge te^{tC}\operatorname{Var}(Y)\right)\le 2e^{-t^2e^{tC}\operatorname{Var}(Y)/2}\,,$$

where $\operatorname{Var}(Y)=\sum_{i=1}^n\operatorname{Var}(X_i)$.

Proof. Just as the lemma itself, this proof is based on (Levchenko, 2013).

Due to symmetry, we only show $\Pr\left(Y\ge te^{tC}\operatorname{Var}(Y)\right)\le 1\cdot e^{-t^2e^{tC}\operatorname{Var}(Y)/2}$.

By Markov's inequality, we have

$$\Pr\left(Y\ge\lambda\sqrt{\operatorname{Var}(Y)}\right)=\Pr\left(e^{tY}\ge e^{t\lambda\sqrt{\operatorname{Var}(Y)}}\right)\le\frac{E\left[e^{tY}\right]}{e^{t\lambda\sqrt{\operatorname{Var}(Y)}}}$$

for all $t>0$ and all $\lambda>0$.

Let Ω_i be the set of possible outcomes of X_i. Then, we have

$$\begin{aligned}
\mathrm{E}\left[e^{tX_i}\right] &= \sum_{x\in\Omega_i}\Pr(X_i=x)e^{t\cdot x}\\
&= \sum_{x\in\Omega_i}\Pr(X_i=x)\sum_{m=0}^{\infty}\frac{(tx)^m}{m!} && \text{((Stewart, 2009, p. 772))}\\
&= \sum_{x\in\Omega_i}\Pr(X_i=x)\cdot\left(1+tx+\sum_{m=2}^{\infty}\frac{(tx)^m}{m!}\right)\\
&= 1+t\cdot E[X_i]+\sum_{x\in\Omega_i}\Pr(X_i=x)\sum_{m=2}^{\infty}\frac{(tx)^m}{m!}\\
&= 1+\sum_{x\in\Omega_i}\Pr(X_i=x)\sum_{m=0}^{\infty}\frac{(tx)^{m+2}}{(m+2)!} && (E[X_i]=0)\\
&\le 1+\sum_{x\in\Omega_i}\Pr(X_i=x)\cdot\frac{(tx)^2}{2}\cdot\sum_{m=0}^{\infty}\frac{(tx)^m}{m!} && (\forall m\ge 0:\ (m+1)(m+2)\ge 2)\\
&= 1+\sum_{x\in\Omega_i}\Pr(X_i=x)\cdot\frac{(tx)^2}{2}\cdot e^{tx}. && \text{((Stewart, 2009, p. 772))}
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathrm{E}\left[e^{tX_i}\right] &\le 1+\sum_{x\in\Omega_i}\Pr(X_i=x)\cdot\frac{(tx)^2}{2}\cdot e^{tC} && (|X_i|\le C)\\
&= 1+\frac{t^2e^{tC}}{2}\sum_{x\in\Omega_i}\Pr(X_i=x)\cdot x^2\\
&= 1+\frac{t^2e^{tC}}{2}\operatorname{Var}(X_i)\,, && (3.1)
\end{aligned}$$

where we use the fact that $\operatorname{Var}(X_i)=\mathrm{E}[X_i^2]-(\mathrm{E}[X_i])^2$ (Mitzenmacher and Upfal, 2005, p. 45) and that, by assumption, $\mathrm{E}[X_i]=0$.

Due to the mutual independence of $X_1, \dots, X_n$, we have

$$\operatorname{Var}(Y) = \sum_{i=1}^{n} \operatorname{Var}(X_i) \tag{3.2}$$

(Mitzenmacher and Upfal, 2005, pp. 46). Likewise, $e^{tX_1}, \dots, e^{tX_n}$ are mutually independent and hence

$$\mathrm{E}\left[\prod_{i=1}^{n} e^{tX_i}\right] = \prod_{i=1}^{n} \mathrm{E}\left[e^{tX_i}\right] \tag{3.3}$$

(Mitzenmacher and Upfal, 2005, pp. 46). Consequently,

$$\begin{aligned}
\mathrm{E}\left[e^{tY}\right] &= \mathrm{E}\left[\prod_{i=1}^{n} e^{tX_i}\right] \\
&= \prod_{i=1}^{n} \mathrm{E}\left[e^{tX_i}\right] && \text{(Equation (3.3))} \\
&\le \prod_{i=1}^{n} \left(1 + \frac{t^2 e^{tC}}{2} \operatorname{Var}(X_i)\right) && \text{(Equation (3.1))} \\
&\le \prod_{i=1}^{n} e^{t^2 e^{tC} \operatorname{Var}(X_i)/2} && (\forall \alpha \in \mathbb{R}_+ : \ 1 + \alpha \le e^{\alpha}) \\
&= e^{t^2 e^{tC} \operatorname{Var}(Y)/2} . && \text{(Equation (3.2))}
\end{aligned}$$

Putting it all together, we have

$$\Pr\left(Y \ge \lambda \sqrt{\operatorname{Var}(Y)}\right) \le e^{t^2 e^{tC} \operatorname{Var}(Y)/2 - t\lambda\sqrt{\operatorname{Var}(Y)}} .$$

Set $\lambda := te^{tC}\sqrt{\operatorname{Var}(Y)}$. With Boole's inequality, the claim follows. □

Example 3.4 (continued). *Observe that* $Y = X - \mathrm{E}[X]$ *for* $Y := \sum_{i=1}^{100} X_i$ *with* $X_i := F_i - 1/2$. *Note that* $\mathrm{E}[X_i] = 0$, $|X_i| \le 1/2 =: C$, *and* $\operatorname{Var}(Y) = 25$.

Let $t \in \mathbb{R}$ *be the value satisfying* $t \cdot \exp(t \cdot C) \operatorname{Var}(Y) = 50$. *One can easily check that* $t \in [1,2]$. *This means that* $t^2 e^{t \cdot C} \operatorname{Var}(Y)/2 \in [25, 50]$. *Hence, the Chernoff-type bound from Lemma 3.9 says that the probability of observing no or 100 heads is at most some probability* δ *with* $\delta = 2 \cdot \exp\left(t^2 e^{t \cdot C} \operatorname{Var}(Y)/2\right) \in \left[4 \cdot 10^{-22},\ 2 \cdot 10^{-11}\right]$.

As one can see from this example, it is somewhat demanding to apply Lemma 3.9. Therefore, we try to substantiate this result as follows:

Theorem 3.10. *Let* $X_1, \dots, X_n$ *be discrete and mutually independent random variables with* $\mathrm{E}[X_i] = 0$ *and* $|X_i| \le C$ *for all* $i \in [n]$ *and some constant* $C \ge 0$. *Let* $Y := \sum_{i=1}^{n} X_i$. *Then, for all* $\delta \in (0,1)$ *we have*

$$\Pr\left(|Y| \ge \lambda \sqrt{\operatorname{Var}(Y)}\right) \le \delta$$

for

$$\lambda = \begin{cases} b_\delta & \text{if } \sqrt{\operatorname{Var}(Y)} \ge C \cdot \frac{b_\delta}{e} \\ \frac{b_\delta^2}{e} \cdot \frac{C}{\sqrt{\operatorname{Var}(Y)}} & \text{otherwise} , \end{cases} \qquad \text{where } \ b_\delta = \sqrt{2e \ln(2/\delta)} .$$

Proof. Let $\lambda(t) := te^{tC}\sqrt{\operatorname{Var}(Y)}$ and $\epsilon(t) := 2e^{-t\lambda(t)\sqrt{\operatorname{Var}(Y)}/2}$ for $t \in \mathbb{R}$.

Due to Lemma 3.9, we have $\Pr\left(|Y| \ge \lambda(t)\sqrt{\operatorname{Var}(Y)}\right) \le \epsilon(t)$ for all $t > 0$.

First, consider the case $\sqrt{\operatorname{Var}(Y)} \ge C\sqrt{2\ln(2/\delta)/e}$. We have $\lambda(0) = 0$ and

$$\lambda(1/C) = (1/C)e\sqrt{\operatorname{Var}(Y)} \ge \sqrt{2e\ln(2/\delta)} .$$

Hence, due to the intermediate value theorem (Golub and Loan, 1996, p. 104) we know that there is some $t_1 \in (0, 1/C]$ such that

$$\lambda(t_1) = \sqrt{2e \ln(2/\delta)} \,. \tag{3.4}$$

Due to the definition of $\lambda(\cdot)$ and since $t_1 \leq 1/C$, we have $\lambda(t_1) \leq t_1 \cdot e^1 \cdot \sqrt{\mathrm{Var}(Y)}$. A combination of this inequality with (3.4) implies

$$t_1 \geq \frac{\sqrt{2e \ln(2/\delta)}}{e\sqrt{\mathrm{Var}(Y)}} \,. \tag{3.5}$$

Thus,

$$\begin{aligned}
\epsilon(t_1) &= 2e^{-t_1^2 e^{t_1 C} \mathrm{Var}(Y)/2} \\
&= 2e^{-t_1 \cdot \sqrt{2e\ln(2/\delta)} \cdot \sqrt{\mathrm{Var}(Y)}/2} && \text{(Equation (3.4))} \\
&\leq 2e^{-\left(\frac{\sqrt{2e\ln(2/\delta)}}{e\sqrt{\mathrm{Var}(Y)}}\right) \cdot \sqrt{2e\ln(2/\delta)} \cdot \sqrt{\mathrm{Var}(Y)}/2} && \text{(Equation (3.5))} \\
&= 2e^{-\ln(2/\delta)} = \delta \,.
\end{aligned}$$

This yields the first part of the claim.

Second, consider the case $\sqrt{\mathrm{Var}(Y)} < C\sqrt{2\ln(2/\delta)/e}$. Observe that $\frac{2\ln(2/\delta)C}{\sqrt{\mathrm{Var}(Y)}} \geq 0$ and that $\lambda(0) = 0$. Since λ is a continuous and strictly increasing function, this means that there is a value $t_2 \in \mathbb{R}_+$ where

$$\lambda(t_2) = \frac{2\ln(2/\delta)C}{\sqrt{\mathrm{Var}(Y)}} \,.$$

Using the definition of $\lambda(\cdot)$, we can conclude

$$t_2 e^{t_2 C} = \frac{\lambda(t_2)}{\sqrt{\mathrm{Var}(Y)}} = \frac{2\ln(2/\delta)C}{\mathrm{Var}(Y)} \,. \tag{3.6}$$

Due to the condition $\sqrt{\mathrm{Var}(Y)} < C\sqrt{2\ln(2/\delta)/e}$, it follows

$$t_2 e^{t_2 C} > \frac{2\ln(2/\delta)C}{(C\sqrt{2\ln(2/\delta)/e})^2} = \frac{e}{C} = (1/C)e^{(1/C)\cdot C} \,.$$

Hence, $t_2 > \frac{1}{C}$. Putting all these inequalities together, we obtain

$$\begin{aligned}
\epsilon(t_2) &= 2e^{-t_2^2 e^{t_2 C} \mathrm{Var}(Y)/2} \\
&< 2e^{-t_2 \cdot \left(\frac{2\ln(2/\delta)C}{\mathrm{Var}(Y)}\right) \cdot \mathrm{Var}(Y)/2} && \text{(by Eq. (3.6))} \\
&= 2e^{-t_2 \cdot (\ln(2/\delta)C)} \\
&< \delta && (t_2 > 1/C)
\end{aligned}$$

This yields the claim. □

Example 3.4 (continued)**.** *For* $\delta := \exp(-49.3) \leq 4 \cdot 10^{-22}$*, we have*

$$\sqrt{\mathrm{Var}(Y)} = 5 > 1/2 \cdot \sqrt{\frac{2\ln(2) + 49.3}{e}} = C \cdot \sqrt{\frac{2\ln(2/\delta)}{e}}$$

and $\sqrt{2e\ln(2/\delta)} = \sqrt{2e(\ln(2) + 49.3)} \geq 49.9$*. Hence, Theorem 3.10 tells us that the probability of observing no or at least 100 heads is at most* $4 \cdot 10^{-22}$*. This matches the (lower bound on the) probability that we derived in the last part of this example.*

Finally, note that there are various other bounds, especially numerous kinds of Chernoff-type bounds, that might be useful for our purpose. Some of these can be found in McDiarmid (1998) and Mitzenmacher and Upfal (2005), for instance. However, to the best of our knowledge, there are no bounds that would lead to substantially better results in the next chapters.

3.5 Analysis

With the help of the probabilistic bounds that we presented in the previous section, we can now analyse the hard clusters constructed by Algorithm 1 in comparison to the given soft clusters. In the following, we analyse the different statistics of the hard clusters separately. Having said that, note that the statistics do actually depend on each other. For instance, the mean and covariance of a hard cluster both directly depend on the weight of this hard cluster (see Section 2.3). Thus, we do not only analyse the statistics separately but decompose each analysis according to these dependencies. We defer the problem of combining the resulting separate probabilistic bounds to Section 3.6, where we discuss two different applications.

3.5.1 Preliminaries

In the following, we consider an arbitrary but fixed probabilistic membership matrix $R = (r_{nk})_{n\in[N],k\in[K]}$ and data set $X = ((x_n, w_n))_{n\in[N]}$. We analyse the statistics of the hard clusters $A_1,\dots,A_K \subseteq X$ that are constructed by a single run of the randomized Algorithm 1, given the membership matrix R and data set X. We compare these statistics with the statistics of the soft clusters of X defined by R. To keep our notation uncluttered, we use the following shorthand notation throughout the rest of this chapter.

Notation 3.11 (shorthand notation)**.** *Given a data set X and a probabilistic membership matrix $R \in [0,1]^{N\times K}$, we let*

$$\mathbf{w}_k := \mathbf{w}\left(A_k^{(X,R)}\right), \quad \mathbf{m}_k := \mathbf{m}\left(A_k^{(X,R)}\right), \quad \mathbf{d}_k := \mathbf{d}\left(A_k^{(X,R)}\right), \quad \mathbf{var}_k := \mathbf{var}\left(A_k^{(X,P)}\right)$$
$$\mathbf{ucov}_k := \mathbf{ucov}\left(A_k^{(X,R)}\right), \quad \textit{and} \quad \mathbf{cov}_k := \mathbf{cov}\left(A_k^{(X,R)}\right)$$

for each $k \in [K]$.

Recall from Algorithm 1 that the assignment of the data point (x_n, w_n) to the k-th cluster is indicated by the value of the variable Z_{nk}, which we consider a binary random variable. Likewise, each hard cluster A_k as well as each of its statistics can be considered a random variable which depends on *all* the binary random variables $\{Z_{nl} \mid n \in [N], l \in [L]\}$. We will extensively use the following properties of these binary random variables.

Lemma 3.12 (assignment variable)**.** *For each $k \in [K]$, the set $\{Z_{nk} \mid n \in [N]\}$ is a set of mutually independent random variables.*

For all $n \in [N]$, $m \in [N]$ with $m \neq n$, and $k \in [K]$, the following equations hold true:

$$\begin{aligned}
\mathrm{E}[Z_{nk}] = \mathrm{E}\left[Z_{nk}^2\right] &= r_{nk}\ , \\
\mathrm{E}[Z_{nk}Z_{mk}] &= r_{nk}r_{mk}\ , \\
\mathrm{Var}(Z_{nk}) &= r_{nk}(1-r_{nk})\ , \textit{ and} \\
\mathrm{Var}\left(\sum_{n=1}^{N} Z_{nk}\right) &= \sum_{n=1}^{N} r_{nk}(1-r_{nk})\ .
\end{aligned}$$

Proof. By construction, for each $k \in [K]$, the set $\{Z_{nk}\}_{n\in[N]}$ is a set of mutually independent variables. Z_{nk} is a binary random variable. Hence, $\mathrm{E}[Z_{nk}] = 1\cdot\Pr(Z_{nk}=1) + 0\cdot\Pr(Z_{nk}=0) = \Pr(Z_{nk}=1) = r_{nk}$. Moreover, this implies that $Z_{nk}^2 = Z_{nk}$. Hence, $\mathrm{E}\left[Z_{nk}^2\right] = \mathrm{E}[Z_{nk}] = r_{nk}$. This means that $\mathrm{Var}(Z_{nk}) = \mathrm{E}\left[Z_{nk}^2\right] - \mathrm{E}[Z_{nk}]^2 = r_{nk}(1-r_{nk})$. Due to the independence, $\mathrm{E}[Z_{nk}Z_{mk}] = \mathrm{E}[Z_{nk}]\cdot\mathrm{E}[Z_{mk}]$ and $\mathrm{Var}\left(\sum_{n=1}^{N} Z_{nk}\right) = \sum_{n=1}^{N}\mathrm{Var}(Z_{nk})$. This yields the claim. □

3.5.2 Weight

We compare the weight $\mathbf{w}(A_k)$ of the k-th hard cluster and the weight $\mathbf{w}_k$ of the k-th soft cluster in terms of their absolute difference

$$|\mathbf{w}(A_k) - \mathbf{w}_k| = \left| \sum_{n=1}^{N} Z_{nk} w_n - \sum_{n=1}^{N} r_{nk} w_n \right| .$$

Basic Properties of $\mathbf{w}(A_k)$

$\mathbf{w}(A_k)$ is a sum of the scaled binary random variables $w_n Z_{nk}$ with $n \in [N]$. Hence, by the linearity of expectation, $\mathrm{E}[\mathbf{w}(A_k)]$ computes to the desired value

$$\mathrm{E}[\mathbf{w}(A_k)] = \mathrm{E}\left[\sum_{n=1}^{N} Z_{nk} w_n \right] = \sum_{n=1}^{N} \mathrm{E}[Z_{nk}] w_n = \sum_{n=1}^{N} r_{nk} w_n = \mathbf{w}_k . \tag{3.7}$$

So we have the nice property that $|\mathbf{w}(A_k) - \mathbf{w}_k| = |\mathbf{w}(A_k) - \mathrm{E}[\mathbf{w}(A_k)]|$ matches the form of the probabilistic bounds that we considered in Section 3.4. Besides that, due to the independence of the summands, we have

$$\mathrm{Var}(\mathbf{w}(A_k)) = \mathrm{Var}\left(\sum_{n=1}^{N} Z_{nk} w_n \right) = \sum_{n=1}^{N} \mathrm{Var}(Z_{nk}) w_n^2 = \sum_{n=1}^{N} r_{nk}(1 - r_{nk}) w_n^2 \tag{3.8}$$

which is upper bounded by

$$\mathrm{Var}(\mathbf{w}(A_k)) = \sum_{n=1}^{N} r_{nk}(1 - r_{nk}) w_n^2 \leq \mathtt{w}_{\max}^{(X)} \cdot \mathbf{w}_k = \mathtt{w}_{\max}^{(X)} \cdot \mathrm{E}[\mathbf{w}(A_k)] . \tag{3.9}$$

In the following, we measure the absolute difference $|\mathbf{w}(A_k) - \mathbf{w}_k|$ in terms of the standard deviation $\sqrt{\mathrm{Var}(\mathbf{w}(A_k))}$ and in terms of its upper bound $\sqrt{\mathtt{w}_{\max}^{(X)} \cdot \mathbf{w}_k}$, respectively.

Using Chebyshev's Inequality

We can bound $|\mathbf{w}(A_k) - \mathbf{w}_k|$ in terms of the standard deviation $\sqrt{\mathrm{Var}(\mathbf{w}(A_k))}$ by using Chebyshev's inequality.

Lemma 3.13. *Let $\delta \in (0,1)$ and $k \in [K]$. We have*

$$\Pr\left(|\mathbf{w}(A_k) - \mathbf{w}_k| \geq \frac{1}{\delta} \cdot \sqrt{\sum_{n=1}^{N} r_{nk}(1 - r_{nk}) w_n^2} \right) \leq \delta^2 . \tag{3.10}$$

Proof. Applying Corollary 3.6 and using (3.7) and (3.8) yields the claim. □

Using a Chernoff-Type Bound

In case the weight $\mathbf{w}_k$ is large enough, we can use a Chernoff-type bound to estimate the difference $|\mathbf{w}(A_k) - \mathbf{w}_k|$ in terms of $\sqrt{\mathtt{w}_{\max}^{(X)} \cdot \mathbf{w}_k}$.

Lemma 3.14. *Let $\delta \in (0,1)$ and $k \in [K]$. If we have*

$$\mathbf{w}_k \geq 3 \ln(2/\delta) \cdot \mathtt{w}_{\max}^{(X)} ,$$

then we know

$$\Pr\left(|\mathbf{w}(A_k) - \mathbf{w}_k| \geq \sqrt{3 \ln(2/\delta) \cdot \mathtt{w}_{\max}^{(X)}} \cdot \sqrt{\mathbf{w}_k} \right) \leq \delta . \tag{3.11}$$

Proof. Observe that $\mathbf{w}(A_k)/\mathrm{w}_{\max}^{(X)} = \sum_{n=1}^{N} Z_{nk} w_n / \mathrm{w}_{\max}^{(X)}$. Each summand $Z_{nk} w_n / \mathrm{w}_{\max}^{(X)}$ lies in $[0,1]$. From (3.7) we know that $\mathrm{E}\left[\mathbf{w}(A_k)/\mathrm{w}_{\max}^{(X)}\right] = \mathbf{w}_k/\mathrm{w}_{\max}^{(X)}$. Applying Corollary 3.8 with $Y = \mathbf{w}(A_k)/\mathrm{w}_{\max}^{(X)}$ yields the claim. □

In comparison to Lemma 3.13, Lemma 3.14 uses a larger unit of measurement (3.9), but bounds the difference by a smaller multiple of these units, namely, $\sqrt{3\ln(2/\delta)}$ instead of $1/\delta$, with respect to the same probability of success $1-\delta$.

3.5.3 Mean Vector

In this section, we analyse the squared Euclidean distance

$$\|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 = \left\| \frac{\sum_{n=1}^{N} Z_{nk} w_n x_n}{\mathbf{w}(A_k)} - \frac{\sum_{n=1}^{N} r_{nk} w_n x_n}{\mathbf{w}_k} \right\|_2^2 . \tag{3.12}$$

Basic Properties of $\mathbf{m}(A_k)$

Consider the quotient $\mathbf{m}(A_k) = \sum_{n=1}^{N} Z_{nk} w_n x_n / \mathbf{w}(A_k)$. Observe that the numerator as well as the denominator are both random variables which depend on each other through the random variables $\{Z_{nk} \mid n \in [N]\}$. In expectation, the numerator and the denominator compute to the desired values: Due to the linearity of expectation and Lemma 3.12, we have

$$\mathrm{E}[\mathbf{w}(A_k) \cdot \mathbf{m}(A_k)] = \mathrm{E}\left[\sum_{n=1}^{N} Z_{nk} w_n x_n\right] = \sum_{n=1}^{N} \mathrm{E}[Z_{nk}] w_n x_n = \sum_{n=1}^{N} r_{nk} w_n x_n = \mathbf{w}_k \cdot \mathbf{m}_k .$$

Besides that, $\mathrm{E}[\mathbf{w}(A_k)] = \mathbf{w}_k$, as already shown in the previous section. Unfortunately, due to the dependency, this does *not* imply that $\mathrm{E}[\mathbf{m}(A_k)]$ equals $\mathbf{m}_k$.

Decomposition of $\|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2$

Observe that we can rewrite the difference as

$$\mathbf{m}(A_k) - \mathbf{m}_k = \frac{\sum_{n=1}^{N} Z_{nk} w_n x_n}{\mathbf{w}(A_k)} - \frac{\sum_{n=1}^{N} Z_{nk} w_n}{\mathbf{w}(A_k)} \mathbf{m}_k = \frac{\sum_{n=1}^{N} Z_{nk} w_n (x_n - \mathbf{m}_k)}{\sum_{n=1}^{N} Z_{nk} w_n} .$$

In the following, we consider the numerator and denominator of

$$\|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 = \frac{\left\|\sum_{n=1}^{N} Z_{nk} w_n (x_n - \mathbf{m}_k)\right\|_2^2}{\mathbf{w}(A_k)^2} \tag{3.13}$$

separately. We already saw in Section 3.5.3 that we can upper and lower bound the denominator $\mathbf{w}(A_k)$. It remains to analyse the numerator

$$\left\| \sum_{n=1}^{N} Z_{nk} w_n (x_n - \mathbf{m}_k) \right\|_2^2 = \sum_{d=1}^{D} \left| \sum_{n=1}^{N} Z_{nk} w_n (x_n - \mathbf{m}_k)_d \right|^2 . \tag{3.14}$$

Using Markov's Inequality

We can bound the numerator in terms of its expected value via Markov's inequality.

Lemma 3.15. *Let $\delta \in (0,1)$ and $k \in [K]$. We have*

$$\Pr\left(\left\| \sum_{n=1}^{N} Z_{nk} w_n (x_n - \mathbf{m}_k) \right\|_2^2 \geq \frac{1}{\delta} \cdot \tau_k^2 \right) \leq \delta \ , \tag{3.15}$$

where

$$\tau_k^2 := \sum_{n=1}^{N} r_{nk} (1 - r_{nk}) w_n^2 \|x_n - \mathbf{m}_k\|_2^2 . \tag{3.16}$$

Proof. Let $M_k := \left\| \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k) \right\|_2^2$. Observe that

$$\begin{aligned} M_k &= \left\langle \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k),\ \sum_{m=1}^N Z_{mk} w_m (x_m - \mathbf{m}_k) \right\rangle \\ &= \sum_{n=1}^N \sum_{m=1}^N Z_{nk} Z_{ok} w_n w_m \left\langle x_n - \mathbf{m}_k,\ x_m - \mathbf{m}_k \right\rangle \ . \end{aligned}$$

Because the expectation is linear, we obtain

$$\begin{aligned} \mathrm{E}[M_k] &= \sum_{n=1}^N \sum_{m=1}^N E\left[Z_{nk} Z_{ok}\right] w_n w_m \left\langle x_n - \mathbf{m}_k,\ x_m - \mathbf{m}_k \right\rangle \\ &= \sum_{n=1}^N E\left[Z_{nk}^2\right] w_n^2 \left\| x_n - \mathbf{m}_k \right\|_2^2 + \sum_{m \neq n} E\left[Z_{nk} Z_{ok}\right] w_n w_m \left\langle (x_n - \mathbf{m}_k),\ (x_m - \mathbf{m}_k) \right\rangle \ . \end{aligned}$$

With Lemma 3.12, we can conclude

$$\begin{aligned} \mathrm{E}[M_k] &= \sum_{n=1}^N r_{nk} w_n^2 \left\| x_n - \mathbf{m}_k \right\|_2^2 + \sum_{m \neq n} r_{nk} r_{mk} w_n w_m \left\langle (x_n - \mathbf{m}_k),\ (x_m - \mathbf{m}_k) \right\rangle \\ &= \sum_{n=1}^N \left(r_{nk} - r_{nk}^2 \right) w_n^2 \left\| x_n - \mathbf{m}_k \right\|_2^2 + \sum_{m=1}^N r_{nk} r_{mk} w_n w_m \left\langle (x_n - \mathbf{m}_k),\ (x_m - \mathbf{m}_k) \right\rangle \\ &= \sum_{n=1}^N r_{nk}(1 - r_{nk}) w_n^2 \left\| x_n - \mathbf{m}_k \right\|_2^2 + r_{nk} w_n \left\langle (x_n - \mathbf{m}_k),\ \sum_{m=1}^N r_{mk} w_m (x_m - \mathbf{m}_k) \right\rangle \\ &= \sum_{n=1}^N r_{nk}(1 - r_{nk}) w_n^2 \left\| x_n - \mathbf{m}_k \right\|_2^2 \ , \end{aligned}$$

where the last equality is due to Observation 2.19. Applying Markov's inequality yields the claim. □

Using a Chernoff-Type Bound

We obtain a more detailed bound by considering each of the D summands

$$\left| \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k)_d \right|^2 \qquad \text{with } d \in [D] \tag{3.17}$$

of the numerator (3.14) separately. We can measure the d-th summand in terms of its variance by using a Chernoff-type inequality.

Lemma 3.16. *Let $\delta \in (0,1)$, $k \in [K]$, and $d \in [D]$. We have*

$$\Pr\left(\left| \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k)_d \right| \geq \lambda_{kd} \cdot \tau_{kd} \right) \leq \delta \tag{3.18}$$

where

$$\tau_{kd}^2 := \sum_{n=1}^N r_{nk}(1 - r_{nk}) w_n^2 (x_n - \mathbf{m}_k)_d^2 \ , \tag{3.19}$$

$$b_\delta := \sqrt{2e \ln(2/\delta)} \ , \text{ and}$$

$$\lambda_{kd} = \begin{cases} b_\delta & \text{if } \tau_{kd} \geq \frac{b_\delta}{e} \mathtt{w}_{\max}^{(X)} \mathtt{r}_d(X) \\ \frac{b_\delta^2}{e} \cdot \frac{\mathtt{w}_{\max}^{(X)} \mathtt{r}_d(X)}{\tau_{kd}} & \text{otherwise} \end{cases} \ . \tag{3.20}$$

Proof. For each $n \in [N]$, define the real random variable

$$M_{kdn} := (Z_{nk} - r_{nk}) w_n (x_n - \mathbf{m}_k)_d \,.$$

Using Lemma 3.12 and the linearity of expectation, we get that $\mathrm{E}[M_{kdn}] = 0$ and

$$\mathrm{Var}(M_{kdn}) = r_{nk}(1 - r_{nk}) w_n^2 (x_n - \mathbf{m}_k)_d^2 \,.$$

Since the Z_{nk} are binary random variables and since each r_{nk} lies in $[0,1]$, we have $|Z_{nk} - r_{nk}| \leq 1$. Since $(\mathbf{m}_k)_d$ is a convex combination of the $(x_n)_d$ with $n \in [N]$, we have $\left|(\mathbf{m}_k - x_n)_d\right| \leq \mathtt{r}_d(X)$. A combination of these inequalities gives

$$|M_{kdn}| = |Z_{nk} - r_{nk}| \cdot |w_n| \cdot \left|(x_n - \mathbf{m}_k)_d\right| \leq \mathtt{w}_{\max}^{(X)} \cdot \mathtt{r}_d(X) \,.$$

Due to Observation 2.19, we know that $\sum_{n=1}^N r_{nk} w_n (x_n - \mathbf{m}_k) = 0_D$. Hence,

$$M_{kd} := \sum_{n=1}^N M_{kdn} = \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k)_d \,.$$

With our previous results, we can conclude that $\mathrm{E}[M_{kd}] = \sum_{n=1}^N \mathrm{E}[M_{kdn}] = 0$. Moreover, since $\{M_{kdn}\}_{n \in [N]}$ is a set of mutually independent random variables, we obtain

$$\mathrm{Var}(M_{kd}) := \sum_{n=1}^N \mathrm{Var}(M_{kdn}) = \sum_{n=1}^N r_{nk}(1 - r_{nk}) w_n^2 (x_n - \mathbf{m}_k)_d^2 = \tau_{kd}^2 \,.$$

Applying Theorem 3.10 with $C := \mathtt{w}_{\max}^{(X)} \mathtt{r}_d(X)$ yields the claim. □

Comparison

Let us briefly compare our bound from Lemma 3.16 with our bound from Lemma 3.15. Fix some $k \in [K]$ and let $M_k := \left\| \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k) \right\|_2^2$. From Lemma 3.16 and Boole's inequality, we can conclude that $M_k \leq \sum_{d=1}^D \lambda_{kd}^2 \tau_{kd}^2$ holds true with a probability of at least $1 - D \cdot \delta$. If the k-th cluster is scattered enough in the sense that

$$\forall d \in [D] \colon \tau_{kd} = \sqrt{\sum_{n=1}^N r_{nk}(1 - r_{nk}) w_n^2 (x_n - \mathbf{m}_k)_d^2} \geq \frac{b_\delta}{e} \mathtt{w}_{\max}^{(X)} \mathtt{r}_d(X) \,,$$

then, for all $d \in [D]$, the factor λ_{kd} is equal to $\sqrt{2e \ln(2/\delta)}$. That is,

$$M_k \leq \sum_{d=1}^D \lambda_{kd}^2 \tau_{kd}^2 = 2e \ln(2/\delta) \cdot \sum_{d=1}^D \tau_{kd}^2 = 2e \ln(2/\delta) \cdot \tau_k^2 \,.$$

with probability $1 - \delta \cdot D$, where τ_k^2 is defined as in Lemma 3.15. For the same probability of success $1 - D \cdot \delta$, Lemma 3.15 guarantees that $M_k \leq \frac{1}{D \cdot \delta} \cdot \tau_k^2$. For sufficiently small δ, our bound from Lemma 3.16 is tighter.

Interpretation

Consider some fixed $k \in [K]$ and let $M_k := \left\| \sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k) \right\|_2^2$. Assume that the k-th cluster does not have too small a weight. More precisely, assume that $\mathbf{w}_k \geq c^2 \cdot 3 \ln(2/\delta) \cdot \mathtt{w}_{\max}^{(X)}$ for some $c \geq 1$. From Lemma 3.14, we can conclude that $\mathbf{w}(A_k) \geq (1 - 1/c)\mathbf{w}_k$ with a probability of at least $1 - \delta$. In this section, we derived a bound on M_k that uses its expected value τ_k^2 as a unit of measurement (see Lemma 3.15). We combine the lower bound on

$\mathbf{w}(A_k)$ and the upper bound on M_k via (3.12). Hence, we measure the squared distance $\|\mathbf{m}(A_k)-\mathbf{m}_k\|_2^2$ between the mean vectors in terms of the quotient $\tau_{kd}^2/\mathbf{w}_k^2$. Observe that

$$\frac{\tau_k^2}{\mathbf{w}_k^2}=\frac{\sum_{d=1}^D\tau_{kd}^2}{\mathbf{w}_k^2}\leq\frac{\sum_{n=1}^N r_{nk}\cdot 1\cdot w_n\cdot \mathtt{w}_{\max}^{(X)}\cdot\|x_n-\mathbf{m}_k\|_2^2}{\mathbf{w}_k^2}=\frac{\mathtt{w}_{\max}^{(X)}}{\mathbf{w}_k}\cdot\frac{\mathbf{d}_k}{\mathbf{w}_k}=\frac{\mathtt{w}_{\max}^{(X)}}{\mathbf{w}_k}\cdot\mathbf{var}_k\leq\mathbf{var}_k\ ,$$

where the last inequality is due to our initial assumption that $\mathbf{w}_k\geq c^2\cdot(3\ln(2/\delta)\mathtt{w}_{\max}^{(X)})$.

To sum up, if the weight of the k-th soft cluster is large enough, then we effectively measure the squared distance $\|\mathbf{m}(A_k)-\mathbf{m}_k\|_2^2$ in terms of a *lower bound* on the variance of the k-th soft cluster.

3.5.4 Covariance Matrix

The difference between the covariance matrices $\mathbf{cov}(A_k)$ and $\mathbf{cov}_k$ can be described by the Frobenius norm

$$\|\mathbf{cov}(A_k)-\mathbf{cov}_k\|_F^2=\left\|\frac{\mathbf{ucov}(A_k)}{\mathbf{w}(A_k)}-\frac{\mathbf{ucov}_k}{\mathbf{w}_k}\right\|_F^2.$$

Basic Properties of $\mathbf{cov}(A_k)$

Just as the mean vector $\mathbf{m}(A_k)$, the covariance $\mathbf{cov}(A_k)$ is the quotient of two random variables that depend on each other through the random variables $\{Z_{nk}\mid n\in[N]\}$. Due to Corollary 2.24, we have

$$\mathbf{cov}(A_k)=\frac{\mathbf{ucov}(A_k)}{\mathbf{w}(A_k)}=\frac{\sum_{n=1}^N\sum_{m<n}Z_{nk}Z_{mk}w_nw_m(x_n-x_m)(x_n-x_m)^T}{\mathbf{w}(A_k)^2}\ . \tag{3.21}$$

In expectation, the numerator of this quotient computes to the desired value

$$\begin{aligned}
\mathrm{E}\left[\mathbf{w}(A_k)^2\cdot\mathbf{cov}(A_k)\right]&=\mathrm{E}\left[\sum_{n=1}^N\sum_{m<n}Z_{nk}Z_{mk}w_nw_m(x_n-x_m)(x_n-x_m)^T\right] && \text{(Eq. (3.21))}\\
&=\sum_{n=1}^N\sum_{m<n}\mathrm{E}[Z_{nk}Z_{mk}]w_nw_m(x_n-x_m)(x_n-x_m)^T && \text{(linearity)}\\
&=\sum_{n=1}^N\sum_{m<n}r_{nk}r_{mk}w_nw_m(x_n-x_m)(x_n-x_m)^T && \text{(Lemma 3.12)}\\
&=\mathbf{w}_k\cdot\mathbf{ucov}_k=\mathbf{w}_k^2\cdot\mathbf{cov}_k\ . && \text{(Corollary 2.24)}
\end{aligned}$$

Besides that, we already know from the previous sections that $\mathrm{E}[\mathbf{w}(A_k)]=\mathbf{w}_k$. Still, from this we cannot conclude that the expected value of $\mathbf{cov}(A_k)$ computes to $\mathbf{cov}_k$.

Decomposition of $\|\mathbf{cov}(A_k)-\mathbf{cov}_k\|_F^2$

Now we could proceed analogously to our analysis of the mean vectors (Section 3.5.3). That is, with the help of (3.21), we can write

$$\|\mathbf{cov}(A_k)-\mathbf{cov}_k\|_F^2=\frac{\left\|\sum_{n=1}^N\sum_{m=1}^N Z_{nk}Z_{mk}w_nw_m\left((x_n-x_m)(x_n-x_m)^T-\mathbf{cov}_k\right)\right\|_F^2}{4\mathbf{w}(A_k)^4}\ .$$

Due to the symmetry of each $(x_n-x_m)(x_n-x_m)^T$ and $\mathbf{cov}_k$, the numerator is equal to

$$\begin{aligned}
&\left\|\sum_{n=1}^N\sum_{m=1}^N Z_{nk}Z_{mk}w_nw_m\left((x_n-x_m)(x_n-x_m)^T-\mathbf{cov}_k\right)\right\|_F^2\\
&=\sum_{i=1}^D\sum_{j\leq i}(2-\delta_{ij})\left|\sum_{n=1}^N\sum_{m\leq n}(2-\delta_{nm})Z_{nk}Z_{mk}w_nw_m\left((x_n-x_m)(x_n-x_m)^T-\mathbf{cov}_k\right)_{ij}\right|^2\ ,
\end{aligned}\tag{3.22}$$

where δ_{pq} denotes the Kronecker delta. With the help of Lemma 3.12, one can compute the expected value of the numerator and apply Markov's inequality (similarly to Lemma 3.15). However, this yields a quiet lengthy result for which we have no better interpretation than for the results that we present in the remainder of this section. Besides that, since the single summands of (3.22) are not mutually independent, we cannot apply our Chernoff-type bounds as in our analysis of the mean vectors (Section 3.5.3). Therefore, we take a different approach.

Upper Bound on $\|\mathbf{cov}(A_k) - \mathbf{cov}_k\|_F^2$

Instead of analysing $\|\mathbf{cov}(A_k) - \mathbf{cov}_k\|_F$, we analyse the following upper bound.

Lemma 3.17 (upper bound)**.** *For all $k \in [K]$, we have*

$$\|\mathbf{cov}(A_k) - \mathbf{cov}_k\|_F \leq \frac{\left\|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)\right\|_F}{\mathbf{w}(A_k)} + \|\mathbf{m}_k - \mathbf{m}(A_k)\|_2^2 \ ,$$

where, for all $n \in [N]$, we have

$$y_{nk} \coloneqq (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T \in \mathbb{R}^{D \times D} \ . \tag{3.23}$$

Proof. With the help of Lemma 2.21, we can write

$$\begin{aligned}
\mathbf{cov}(A_k) - \mathbf{cov}_k &= \frac{\mathbf{ucov}(A_k, \mathbf{m}_k) - \mathbf{w}(A_k) \cdot (\mathbf{m}_k - \mathbf{m}(A_k)(\mathbf{m}_k - \mathbf{m}(A_k))^T}{\mathbf{w}(A_k)} - \mathbf{cov}_k \\
&= \frac{\mathbf{ucov}(A_k, \mathbf{m}_k) - \mathbf{w}(A_k)\mathbf{cov}_k}{\mathbf{w}(A_k)} - (\mathbf{m}_k - \mathbf{m}(A_k)(\mathbf{m}_k - \mathbf{m}(A_k))^T \ .
\end{aligned}$$

The numerator of the minuend computes to

$$\begin{aligned}
\mathbf{ucov}(A_k, \mathbf{m}_k) - \mathbf{w}(A_k)\mathbf{cov}_k &= \left(\sum_{n=1}^N Z_{nk} w_n y_{nk}\right) - \left(\sum_{n=1}^N Z_{nk} w_n \mathbf{cov}_k\right) \\
&= \sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k) \ .
\end{aligned}$$

Observe that, for all $v \in \mathbb{R}^D$, we have

$$\left\|v\, v^T\right\|_F^2 = \sum_{i=1}^D \sum_{j=1}^D \left((v)_i \cdot (v)_j\right)^2 = \sum_{i=1}^D \sum_{j=1}^D (v)_i^2 \cdot (v)_j^2 = \left(\sum_{i=1}^D (v)_i^2\right)^2 = \|v\|_2^4 \ .$$

Hence, the Frobenius norm of the subtrahend computes to

$$\left\|(\mathbf{m}_k - \mathbf{m}(A_k))(\mathbf{m}_k - \mathbf{m}(A_k))^T\right\|_F = \|\mathbf{m}_k - \mathbf{m}(A_k)\|_2^2 \ .$$

Finally, applying the triangle inequality yields the claim. □

We analyse and bound the single terms of the upper bound from Lemma 3.17 separately: Recall that we already know how to bound the denominator $\mathbf{w}(A_k)$ of the first summand and the second summand $\|\mathbf{m}_k - \mathbf{m}(A_k)\|_2^2$. Hence, in the following, we only consider the numerator $\left\|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)\right\|_F$ of the first summand. Observe that

$$\begin{aligned}
&\left\|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)\right\|_F^2 \\
&= \sum_{i=1}^D \sum_{j=1}^D \left|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_{ij}\right|^2 \\
&= \sum_{i=1}^D \left|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_{ii}\right|^2 + 2 \sum_{\substack{j \in [D] \\ j<i}} \left|\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_{ij}\right|^2 \ ,
\end{aligned} \tag{3.24}$$

where the last inequality is due to the symmetry of the matrices $y_{nk} = (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T$ and $\mathbf{cov}_k$ (see Definition 2.15). Via Boole's inequality, we will later combine the resulting bounds into a bound on (the upper bound on) the difference between the covariance matrices.

Using Markov's Inequality

We can bound the random variable from (3.24) in terms of its expected value by using Markov's inequality.

Lemma 3.18. *Let $\delta \in (0,1)$ and $k \in [K]$. We have*

$$\Pr\left(\left\| \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k) \right\|_F^2 \geq \frac{1}{\delta} \cdot \rho_k^2 \right) \leq \delta \ , \tag{3.25}$$

where $y_{nk} = (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T$ for all $n \in [N]$ and

$$\rho_k = \sum_{n=1}^{N} r_{nk}(1 - r_{nk}) w_n^2 \, \| y_{nk} - \mathbf{cov}_k \|_F^2 \ . \tag{3.26}$$

Proof. The following proof is an analogon of Lemma 3.15. Observe that

$$\begin{aligned} S_k &\coloneqq \left\| \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k) \right\|_F^2 \\ &= \left\langle \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k), \sum_{m=1}^{N} Z_{mk} w_m (y_{mk} - \mathbf{cov}_k) \right\rangle_F \\ &= \sum_{n=1}^{N} \sum_{m=1}^{N} Z_{nk} Z_{mk} w_n w_m \, \langle y_{nk} - \mathbf{cov}_k, \, y_{mk} - \mathbf{cov}_k \rangle_F \ , \end{aligned}$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product. By linearity of expectation, we know

$$\mathrm{E}[S_k] = \sum_{n=1}^{N} \sum_{m=1}^{N} \mathrm{E}[Z_{nk} Z_{mk}] w_n w_m \, \langle y_{nk} - \mathbf{cov}_k, \, y_{mk} - \mathbf{cov}_k \rangle_F \ .$$

With the help of Lemma 3.12, we can conclude

$$\begin{aligned} &\mathrm{E}[S_k] \\ &= \sum_{n=1}^{N} r_{nk} w_n^2 \, \| y_{nk} - \mathbf{cov}_k \|_F^2 + \sum_{m \neq n} r_{nk} r_{mk} w_n w_m \, \langle y_{nk} - \mathbf{cov}_k, \, y_{mk} - \mathbf{cov}_k \rangle_F \\ &= \sum_{n=1}^{N} (r_{nk} - r_{nk}^2) w_n^2 \, \| y_{nk} - \mathbf{cov}_k \|_F^2 + \sum_{m=1}^{N} r_{nk} r_{mk} w_n w_m \, \langle y_{nk} - \mathbf{cov}_k, \, y_{mk} - \mathbf{cov}_k \rangle_F \\ &= \sum_{n=1}^{N} (r_{nk} - r_{nk}^2) w_n^2 \, \| y_{nk} - \mathbf{cov}_k \|_F^2 + r_{nk} w_n \left\langle y_{nk} - \mathbf{cov}_k, \sum_{m=1}^{N} r_{mk} w_m (y_{mk} - \mathbf{cov}_k) \right\rangle_F \\ &= \sum_{n=1}^{N} r_{nk}(1 - r_{nk}) w_n^2 \, \| y_{nk} - \mathbf{cov}_k \|_F^2 \\ &= \rho_k^2 \ , \end{aligned}$$

where the second to the last equality is due to Observation 2.19. Applying Markov's inequality yields the claim. □

Using a Chernoff-Type Bound

We can refine this bound by considering each of the $D(D+1)/2$ different summands

$$\left| \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_{ij} \right|^2 \quad \text{with } i,j \in [D],\ j \leq i$$

of the random variable from (3.24) separately. We can bound the summand with indices $i,j \in [D]$ in terms of its variance by using a Chernoff-type bound.

Lemma 3.19. *Let $\delta \in (0,1)$, $k \in [K]$, and $i,j \in [D]$. We have*

$$\Pr\left(\left| \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_{ij} \right|^2 > \lambda_{kij}^2 \cdot \rho_{kij}^2 \right) \leq \delta \ , \tag{3.27}$$

where $y_{nk} = (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T$ for all $n \in [N]$,

$$\begin{aligned} \rho_{kij}^2 &:= \sum_{n=1}^{N} r_{nk}(1-r_{nk}) w_n^2 (y_{nk} - \mathbf{m}_k)_i (y_{nk} - \mathbf{m}_k)_j \ , \\ b_\delta &:= \sqrt{2e \ln(2/\delta)} \ , \text{ and} \\ \lambda_{kij} &= \begin{cases} b_\delta & \text{if } \rho_{kij} \geq \frac{2b_\delta}{e} \mathtt{w}_{\max}^{(X)} \mathtt{r}_i(X) \mathtt{r}_j(X) \\ \frac{2b_\delta^2}{e} \frac{\mathtt{w}_{\max}^{(X)} \mathtt{r}_i(X) \mathtt{r}_j(X)}{\rho_{kij}} & \text{otherwise} \end{cases} . \end{aligned} \tag{3.28}$$

Proof. For each $n \in [N]$, define the real random variable

$$S_{kijn} := (Z_{nk} - r_{nk}) w_n (y_{nk} - \mathbf{cov}_k)_{ij} \ ,$$

where $(y_{nk})_{ij} = (x_n - \mathbf{m}_k)_i (x_n - \mathbf{m}_k)_j$. Since the Z_{nk} are binary random variables and since each membership r_{nk} lies in $[0,1]$, we have $|Z_{nk} - r_{nk}| \leq 1$. Since $(\mathbf{m}_k)_d$ is a convex combination of the coordinates $(x_m)_d$ with $m \in [N]$, we know that $(x_n - \mathbf{m}_k)_d \in [-\mathtt{r}_d(X), +\mathtt{r}_d(X)]$ for all $n \in [N]$. Hence, for all $n \in [N]$ and $i,j \in [D]$, we can conclude that $(y_{nk})_{ij} = (x_n - \mathbf{m}_k)_i (x_n - \mathbf{m}_k)_j \in \left[-\mathtt{r}_i(X) \cdot \mathtt{r}_j(X),\ +\mathtt{r}_i(X) \cdot \mathtt{r}_j(X)\right]$. As $(\mathbf{cov}_k)_{ij}$ is a convex combination of all values $(y_{mk})_{ij}$ with $m \in [N]$, it follows that $(y_{nk} - \mathbf{cov}_k)_{ij} \in \left[-2 \cdot \mathtt{r}_i(X)\mathtt{r}_j(X),\ +2 \cdot \mathtt{r}_i(X)\mathtt{r}_j(X)\right]$ for all $n \in [N]$. Hence, $\left|(y_{nk} - \mathbf{cov}_k)_{ij}\right| \leq 2\mathtt{r}_i(X)\mathtt{r}_j(X)$ for all $n \in [N]$. Putting these inequalities together yields

$$\left|S_{kijn}\right| = |Z_{nk} - r_{nk}| \cdot |w_n| \cdot \left|(y_{nk} - \mathbf{cov}_k)_{ij}\right| \leq 2\mathtt{w}_{\max}^{(X)} \mathtt{r}_i(X)\mathtt{r}_j(X) \ .$$

Due to Observation 2.19, we know that

$$S_{kij} := \sum_{n=1}^{N} S_{kijn} = \sum_{n=1}^{N} Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)_i (y_{nk} - \mathbf{cov}_k)_j \ .$$

With Lemma 3.12, we can conclude that

$$\mathrm{E}[S_{kijn}] = 0 \quad \text{and} \quad \mathrm{Var}(S_{kijn}) = r_{nk}(1-r_{nk}) w_n^2 (y_{nk} - \mathbf{cov}_k)_{ij}^2 \ .$$

Hence, we have $\mathrm{E}[S_{kij}] = 0$. Moreover, since the summands S_{kijn} are mutually independent random variables, we have $\mathrm{Var}(S_{kij}) = \rho_{kij}^2$.

Hence, applying Theorem 3.10 with $C := 2\mathtt{w}_{\max}^{(X)} \mathtt{r}_i(X)\mathtt{r}_j(X)$ yields the claim. □

3.5.5 Cost and Variance

In this section, we want to measure the cost

$$\mathbf{d}(A_k) = \sum_{n=1}^{N} z_{nk} w_n \|x_n - \mathbf{m}(A_k)\|_2 \tag{3.29}$$

of the hard cluster A_k in terms of the cost $\mathbf{d}_k = \sum_{n=1}^{N} r_{nk} w_n \|x_n - \mathbf{m}_k\|_2$ of the given soft cluster. Recall that the variance $\mathbf{var}(A_k)$ is the quotient of the cost $\mathbf{d}(A_k)$ and the weight $\mathbf{w}(A_k)$. Hence, our results from this section and our bounds from Section 3.5.2 can easily be combined into an upper bound that measures the variance $\mathbf{var}(A_k)$ in terms of $\mathbf{var}_k$.

Basic Properties of $\mathbf{d}(A_k)$

Recall from Corollary 2.23 that

$$\mathbf{d}(A_k) = \frac{\sum_{n=1}^{N} \sum_{m<n} Z_{nk} Z_{mk} w_n w_m \|x_n - x_m\|_2^2}{\mathbf{w}(A_k)} \,. \tag{3.30}$$

We do not know whether the expected value of $\mathbf{d}(A_k)$ computes to the cost $\mathbf{d}_k$. However, we know that the numerator and denominator of (3.30) compute to the desired values: We have

$$\begin{aligned}
\mathrm{E}[\mathbf{w}(A_k) \cdot \mathbf{d}(A_k)] &= \mathrm{E}\left[\sum_{n=1}^{N} \sum_{m<n} Z_{nk} Z_{mk} w_n w_m \|x_n - x_m\|_2^2 \right] \\
&= \sum_{n=1}^{N} \sum_{m<n} \mathrm{E}[Z_{nk} Z_{mk}] w_n w_m \|x_n - x_m\|_2^2 && \text{(linearity)} \\
&= \sum_{n=1}^{N} \sum_{m<n} r_{nk} r_{mk} w_n w_m \|x_n - x_m\|_2^2 && \text{(Lemma 3.12)} \\
&= \mathbf{w}_k \cdot \mathbf{d}(A_k) \,.
\end{aligned}$$

and, as we already showed in Section 3.5.2, $\mathrm{E}[\mathbf{w}(A_k)] = \mathbf{w}_k$. The ratio of these expected value is equal to $\mathbf{d}_k$.

Now we could analyse the numerator and denominator from (3.30) separately via Markov's inequality and combine the resulting bounds via Boole's inequality. However, this bound does not yield better results in the following chapters. Besides that, we cannot apply to Chernoff-type bounds (similarly to Section 3.5.3) since neither the summands of (3.29) nor the summands of the numerator from (3.30) are mutually independent. Therefore, we proceed as follows.

Upper Bound on $\mathbf{d}(A_k)$

Consider the following upper bound. Due to Lemma 2.20, we have

$$\mathbf{d}(A_k) = \mathbf{d}(A_k, \mathbf{m}_k) - \mathbf{w}(A_k) \cdot \|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 \leq \mathbf{d}(A_k, \mathbf{m}_k) = \sum_{n=1}^{N} Z_{nk} w_n \|x_n - \mathbf{m}_k\|_2^2 \,. \tag{3.31}$$

The upper bound $\mathbf{d}(A_k, \mathbf{m}_k)$ is a sum of mutually independent random variables. Due to the linearity of expectation and Lemma 3.12, its expected value computes to

$$\mathrm{E}[\mathbf{d}(A_k, \mathbf{m}_k)] = \sum_{n=1}^{N} \mathrm{E}[Z_{nk}] w_n \|x_n - \mathbf{m}_k\|_2^2 = \sum_{n=1}^{N} r_{nk} w_n \|x_n - \mathbf{m}_k\|_2^2 = \mathbf{d}_k \,. \tag{3.32}$$

Using Markov's Inequality

We can easily bound the upper bound from (3.31) via Markov's inequality.

Lemma 3.20. *Let $\delta \in (0,1)$ and $k \in [K]$. We have*

$$\Pr\left(\mathbf{d}(A_k) \geq \frac{1}{\delta} \cdot \mathbf{d}_k\right) \leq \delta \ . \tag{3.33}$$

Proof. Apply Markovs's inequality to the upper bound from (3.31) and use (3.32). □

3.6 Conclusions

In this section, we present two different ways of using the probabilistic bounds from Section 3.5. Again, we use the following shorthand notation:

Notation 3.11 (shorthand notation)**.** *Given a data set X and a probabilistic membership matrix $R \in [0,1]^{N \times K}$, we let*

$$\mathbf{w}_k := \mathbf{w}\left(A_k^{(X,R)}\right), \quad \mathbf{m}_k := \mathbf{m}\left(A_k^{(X,R)}\right), \quad \mathbf{d}_k := \mathbf{d}\left(A_k^{(X,R)}\right), \quad \mathbf{var}_k := \mathbf{var}\left(A_k^{(X,P)}\right)$$
$$\mathbf{ucov}_k := \mathbf{ucov}\left(A_k^{(X,R)}\right), \quad \textit{and} \quad \mathbf{cov}_k := \mathbf{cov}\left(A_k^{(X,R)}\right)$$

for each $k \in [K]$.

3.6.1 Existence of Similar Hard Clusters

By combining our probabilistic bounds from Section 3.5 via the union bound and applying the probabilistic method, we can prove the existence of hard clusters that imitate given soft clusters. In this way, we obtain the following Theorem 3.21, which we will apply in Chapter 8 in the context of the fuzzy K-means problem. More precisely, the theorem is obtained by combining our probabilistic bounds (on the similarity of the weights, means, and costs) that are based on Markov's and Chebyshevs's inequality. In contrast to our Chernoff-type bounds, which distinguish between several cases, these inequalities yield a much simpler result.

Theorem 3.21 (Existence of Similar Hard Clusters)**.** *Let $X = ((x_n, w_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $R = (r_{nk})_{n \in [N], k \in [K]}$ be a probabilistic membership matrix, and $\epsilon \in (0,1]$. Use Notation 3.11.*

If we have

$$\forall k \in [K] : \ \mathbf{w}_k \geq \frac{16K}{\epsilon} \mathrm{w}_{\max}^{(X)} \ , \tag{3.34}$$

then there exist pairwise disjoint hard clusters $A_1, \ldots, A_K \subseteq X$ such that

$$\mathbf{w}(A_k) \geq \frac{1}{2} \cdot \mathbf{w}_k \ , \tag{3.35}$$

$$\|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 \leq \epsilon \cdot \frac{\mathbf{d}_k}{\mathbf{w}_k} \ , \textit{ and} \tag{3.36}$$

$$\mathbf{d}(A_k) \leq 4K \cdot \mathbf{d}_k \tag{3.37}$$

for all $k \in [K]$.

If R is a soft K-clustering of X, then the clusters $A_1, \ldots, A_K$ form a hard clustering of X. That is, $\dot{\bigcup}_{k=1}^K A_k = X$.

In the remainder of this section, we prove Theorem 3.21. A straightforward combination of Lemma 3.13, Lemma 3.15, and Lemma 3.20 yields the following result:

Corollary 3.22. *Let $\epsilon \in (0,1]$, $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, and let $R = (r_{nk})_{n\in[N],k\in[K]}$ be a membership matrix. Use Notation 3.11. There exist pairwise disjoint hard clusters $A_1,\dots,A_K \subseteq X$ such that for all $k \in [K]$ we have*

$$|\mathbf{w}(A_k) - \mathbf{w}_k| \leq \sqrt{4K}\cdot\eta_k \ , \tag{3.38}$$

$$\|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 \leq 4K \cdot \frac{\tau_k^2}{\left(\mathbf{w}_k - \sqrt{4K}\eta_k\right)^2} \ , \text{ and} \tag{3.39}$$

$$\mathbf{d}(A_k) \leq 4K\cdot\mathbf{d}_k \ , \tag{3.40}$$

where $\eta_k = \sqrt{\sum_{n=1}^N r_{nk}(1-r_{nk})w_n^2}$ and $\tau_k^2 = \sum_{n=1}^N r_{nk}(1-r_{nk})w_n^2\|x_n - \mathbf{m}_k\|_2^2$.

If R is a soft K-clustering of X, then $\dot{\cup}_{k=1}^K A_k = X$.

Proof. Apply Lemma 3.13 with $\delta^2 = \frac{1}{4K}$, Lemma 3.15 with $\delta = \frac{1}{4K}$, and Lemma 3.20 with $\delta = \frac{1}{4K}$, we can take the union bound and obtain that the inequalities stated in the lemmata hold simultaneously with probability $1 - 3K\cdot\frac{1}{4K} > 0$ strictly larger than 0. Finally, using (3.13) yields the claim. □

Given this result, we can prove Theorem 3.21.

Proof of Theorem 3.21. Write $X = ((x_n, w_n))_{n\in[N]}$. Scale the weights of the data points by $1/\mathtt{w}_{\max}^{(X)}$, and let

$$\hat{X} := \left(\left(x_n, \frac{w_n}{\mathtt{w}_{\max}^{(X)}}\right)\right)_{n\in[N]} .$$

Similar to Notation 3.11, let

$$\hat{\mathbf{w}}_k := \mathbf{w}\left(A_k^{(\hat{X},R)}\right) \ , \ \hat{\mathbf{m}}_k := \mathbf{m}\left(A_k^{(\hat{X},R)}\right) \ , \text{ and } \ \hat{\mathbf{d}}_k := \mathbf{d}\left(A_k^{(\hat{X},R)}\right)$$

for each $k \in [K]$.

Let $\hat{A}_1,\dots,\hat{A}_K \subseteq \hat{X}$ be the pairwise disjoint subsets of X whose existence is guaranteed by Corollary 3.22 when applied to the given ϵ, the given membership matrix $R = (r_{nk})_{n,k}$, and data set $\hat{X}$ (instead of X). For each cluster $\hat{A}_k = \left((x_n, w_n/\mathtt{w}_{\max}^{(X)})\right)_{n\in I_k} \subseteq \hat{X}$ with some $I_k \subseteq [N]$, define the corresponding clusters $A_k := ((x_n,w_n))_{n\in I_k} \subseteq X$.

Consider an arbitrary but fixed $k \in [K]$. Due to Lemma 2.25, we have:

$$\hat{\mathbf{w}}_k = \frac{1}{\mathtt{w}_{\max}^{(X)}}\cdot\mathbf{w}_k \qquad \hat{\mathbf{m}}_k = \mathbf{m}_k \qquad \hat{\mathbf{d}}_k = \frac{1}{\mathtt{w}_{\max}^{(X)}}\cdot\mathbf{d}_k \tag{3.41}$$

$$\mathbf{w}\left(\hat{A}_k\right) = \frac{1}{\mathtt{w}_{\max}^{(X)}}\cdot\mathbf{w}(A_k) \qquad \mathbf{m}\left(\hat{A}_k\right) = \mathbf{m}(A_k) \qquad \mathbf{d}\left(\hat{A}_k\right) = \frac{1}{\mathtt{w}_{\max}^{(X)}}\cdot\mathbf{d}(A_k) \tag{3.42}$$

Moreover, due to (3.41) and Condition (3.34), we have

$$\epsilon\cdot\hat{\mathbf{w}}_k = \epsilon\cdot\frac{1}{\mathtt{w}_{\max}^{(X)}}\mathbf{w}_k \geq 16K \ . \tag{3.43}$$

In the following, we show that A_k satisfies (3.35), (3.36), and (3.37). First, we prove (3.35). With the help of (3.43), we can bound

$$\sqrt{4K} = \frac{\sqrt{16K}}{2} \leq \frac{\sqrt{\epsilon\hat{\mathbf{w}}_k}}{2} \leq \frac{\sqrt{\hat{\mathbf{w}}_k}}{2} \ . \tag{3.44}$$

Since we apply Corollary 3.22 to $\hat{X}$ instead of X, we have

$$\eta_k = \sqrt{\sum_{n=1}^N r_{nk}(1-r_{nk})\left(\frac{w_n}{\mathtt{w}_{\max}^{(X)}}\right)^2} \leq \sqrt{\sum_{n=1}^N r_{nk}\left(\frac{w_n}{\mathtt{w}_{\max}^{(X)}}\right)} = \sqrt{\hat{\mathbf{w}}_k} \ . \tag{3.45}$$

Consider (3.38) from Corollary 3.22. By combining this inequality with (3.41), (3.42), (3.44) and (3.45), we obtain

$$\frac{1}{\mathtt{w}_{\max}^{(X)}} |\mathbf{w}(A_k) - \mathbf{w}_k| = \left|\mathbf{w}(\hat{A}_k) - \hat{\mathbf{w}}_k\right| \leq \sqrt{4K}\eta_k < \frac{\hat{\mathbf{w}}_k}{2} = \frac{1}{\mathtt{w}_{\max}^{(X)}} \cdot \frac{1}{2}\mathbf{w}_k \ . \tag{3.46}$$

This yields (3.35).

Next, we prove (3.36). Recall that we apply Corollary 3.22 to $\hat{X}$ instead of X. Hence, with the help of (3.41), we can conclude

$$\tau_k^2 = \sum_{n=1}^{N} r_{nk}(1 - r_{nk}) \left(\frac{w_n}{\mathtt{w}_{\max}^{(X)}}\right)^2 \|x_n - \hat{\mathbf{m}}_k\|_2^2 \tag{3.47}$$

$$\leq \sum_{n=1}^{N} r_{nk} \left(\frac{w_n}{\mathtt{w}_{\max}^{(X)}}\right) \|x_n - \hat{\mathbf{m}}_k\|_2^2 = \hat{\mathbf{d}}_k = \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{d}_k \ . \tag{3.48}$$

Using (3.41) and our upper bound on $\sqrt{4K}\eta_k$ from (3.46), we obtain

$$\hat{\mathbf{w}}_k - \sqrt{4K}\eta_k = \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{w}_k - \sqrt{4K}\eta_k \geq \frac{1}{2} \cdot \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{w}_k > 0 \ . \tag{3.49}$$

From (3.43), we can conclude

$$4K \leq \frac{\epsilon}{4} \cdot \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{w}_k \ . \tag{3.50}$$

Consider (3.39) from Corollary 3.22. By combining this inequality with (3.41), (3.42), (3.48), (3.49), and (3.50), we obtain

$$\begin{aligned} \|\mathbf{m}(A_k) - \mathbf{m}_k\|_2^2 &= \left\|\mathbf{m}(\hat{A}_k) - \hat{\mathbf{m}}_k\right\|_2^2 \\ &\leq \frac{4K}{(\hat{\mathbf{w}}_k - \sqrt{4K}\eta_k)^2} \tau_k^2 \\ &\leq \frac{\epsilon}{4} \cdot \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{w}_k \cdot \left(\frac{1}{2} \cdot \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{w}_k\right)^{-2} \cdot \frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{d}_k \\ &= \epsilon \cdot \frac{\mathbf{d}_k}{\mathbf{w}_k} \ . \end{aligned}$$

This yields (3.36).

Finally, we prove (3.37). Consider (3.40) from Corollary 3.22. With (3.41) and (3.42), we can conclude

$$\frac{1}{\mathtt{w}_{\max}^{(X)}} \mathbf{d}(A_k) = \mathbf{d}\left(\hat{A}_k\right) \leq 4K \cdot \hat{\mathbf{d}}_k = \frac{1}{\mathtt{w}_{\max}^{(X)}} \cdot 4K \cdot \mathbf{d}_k \ .$$

This yields (3.37). □

3.6.2 Quality of an Imitation

As already explained in Section 3.4, our Chernoff-type bounds are tighter than those based on Markov's or Chebyshev's inequality. Hence, to bound the similarity of the hard clusters computed by Algorithm 1 and the given soft clusters, one should use our Chernoff-type bounds. To obtain a bound on the "overall similarity", one can simply combine the bounds on the single cluster statistics via Boole's inequality. In this way, we obtain the following theorem, which we will use in Chapter 15 to analyse the stochastic expectation maximization algorithm for Gaussian mixture models.

Theorem 3.23 (Quality of an Imitation)**.** *Consider a single run of Algorithm 1 that is given some probabilistic membership matrix $R = (r_{nk})_{n\in[N],k\in[K]}$ and data set $X = ((x_n, w_n))_{n\in[N]}$. Use Notation 3.11. Let $A_1,\dots,A_K$ be the hard clusters constructed by the algorithm.*

Let $\delta \in (0,1)$. Set $a_\delta := 3\ln(2/\delta)\cdot \mathtt{w}_{\max}^{(X)}$ and $b_\delta := \sqrt{2e\ln(2/\delta)}$.

If we have

$$\forall k \in [K]\colon\ \mathbf{w}_k \geq a_\delta\ ,$$

then, with probability $1 - K\cdot\left(1 + D + \frac{D(D+1)}{2}\right)\cdot\delta$, for all $k\in[K]$ and $i,j\in[D]$ we have

$$|\mathbf{w}(A_k) - \mathbf{w}_k| \leq \sqrt{a_\delta}\cdot\sqrt{\mathbf{w}_k}\ , \tag{3.51}$$

$$\left|(\mathbf{m}(A_k) - \mathbf{m}_k)_i\right| \leq \frac{\lambda_{ki}}{\sqrt{\mathbf{w}_k} - \sqrt{a_\delta}}\cdot\frac{\tau_{ki}}{\sqrt{\mathbf{w}_k}}\ ,\ \textit{and} \tag{3.52}$$

$$\left|(\mathbf{cov}(A_k) - \mathbf{cov}_k)_{ij}\right| \leq \frac{\lambda_{kij}}{\sqrt{\mathbf{w}_k} - \sqrt{a_\delta}}\cdot\frac{\rho_{kij}}{\sqrt{\mathbf{w}_k}} + \frac{\lambda_{ki}\lambda_{kj}}{\left(\sqrt{\mathbf{w}_k} - \sqrt{a_\delta}\right)^2}\frac{\tau_{ki}\tau_{kj}}{\mathbf{w}_k}\ , \tag{3.53}$$

where, for all $k\in[K]$ and $i,j\in[D]$, we have

$$\tau_{ki}^2 = \sum_{n=1}^{N} p_{nk}(1-p_{nk})^2 w_n^2 (x_n - \mathbf{m}_k)_i^2$$

$$\lambda_{ki} = \begin{cases} b_\delta & \textit{if } \tau_{ki} \geq \frac{b_\delta}{e}\mathtt{w}_{\max}^{(X)}\mathtt{r}_i(X) \\ \frac{b_\delta^2}{e}\cdot\frac{\mathtt{w}_{\max}^{(X)}\mathtt{r}_i(X)}{\tau_{ki}} & \textit{otherwise} \end{cases}\ ,$$

$$\rho_{kij}^2 = \sum_{n=1}^{N} r_{nk}(1-r_{nk}) w_n^2 (y_{nk} - \mathbf{cov}_k)_i (y_{nk} - \mathbf{cov}_k)_j\ ,\ \textit{where}$$

$$y_{nk} = (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T\ \textit{ for all } n\in[N],\ \textit{and}$$

$$\lambda_{kij} = \begin{cases} b_\delta & \textit{if } \rho_{kij} \geq \frac{2b_\delta}{e}\cdot\mathtt{w}_{\max}^{(X)}\mathtt{r}_i(X)\mathtt{r}_j(X) \\ \frac{2b_\delta^2}{e}\frac{\mathtt{w}_{\max}^{(X)}\mathtt{r}_i(X)\mathtt{r}_j(X)}{\rho_{kij}} & \textit{otherwise} \end{cases}\ .$$

Proof. We combine Lemma 3.14, Lemma 3.16 and Lemma 3.19 by taking the union bound. Thereby, we obtain that, with probability $1 - K\cdot\left(1 + D + \frac{D(D+1)}{2}\right)\cdot\delta$, the events considered in (3.11), (3.18), and (3.27) hold simultaneously for all $d,i,j\in[D]$ and $k\in[K]$.

Fix some $d,i,j\in[D]$ and $k\in[K]$. (3.51) follows directly from (3.11). To prove (3.52), just observe that, due to (3.11) and (3.18), we have

$$\left|(\mathbf{m}(A_k) - \mathbf{m}_k)_d\right| = \frac{\left|\sum_{n=1}^N Z_{nk} w_n (x_n - \mathbf{m}_k)_d\right|}{\mathbf{w}(A_k)} \leq \frac{\lambda_{kd}}{(\sqrt{\mathbf{w}_k} - \sqrt{a_\delta})}\cdot\frac{\tau_{kd}}{\sqrt{\mathbf{w}_k}}\ . \tag{3.54}$$

To prove (3.53), we proceed as in the proof of Lemma 3.17. Let $\mathfrak{m}_k := (\mathbf{m}_k - \mathbf{m}(A_k))(\mathbf{m}_k - \mathbf{m}(A_k))^T$ and $y_{nk} := (x_n - \mathbf{m}_k)(x_n - \mathbf{m}_k)^T$, for all $n\in[N]$. Then,

$$\begin{aligned}
\left|(\mathbf{cov}(A_k) - \mathbf{cov}_k)_{ij}\right| &= \left|\left(\frac{\mathbf{ucov}(A_k,\mathbf{m}_k) - \mathbf{w}(A_k)\mathfrak{m}_k}{\mathbf{w}(A_k)} - \mathbf{cov}_k\right)_{ij}\right| && \text{(Lemma 2.21)} \\
&= \left|\left(\frac{\sum_{n=1}^N Z_{nk} w_n y_{nk}}{\sum_{n=1}^N Z_{nk} w_n} - \mathfrak{m}_k - \mathbf{cov}_k\right)_{ij}\right| \\
&\leq \left|\left(\frac{\sum_{n=1}^N Z_{nk} w_n y_{nk}}{\sum_{n=1}^N Z_{nk} w_n} - \mathbf{cov}_k\right)_{ij}\right| + \left|(\mathfrak{m}_k)_{ij}\right| && \text{(triangle inequality)} \\
&= \frac{\left|\left(\sum_{n=1}^N Z_{nk} w_n (y_{nk} - \mathbf{cov}_k)\right)_{ij}\right|}{\mathbf{w}(A_k)} + \left|(\mathfrak{m}_k)_{ij}\right|\ .
\end{aligned}$$

Due to (3.54) and (3.11), we have

$$\left|(\mathfrak{m}_k)_{ij}\right| = \left|(\mathbf{m}_k - \mathbf{m}(A_k))_i\right| \cdot \left|(\mathbf{m}_k - \mathbf{m}(A_k))_j\right| \leq \frac{\lambda_{ki}\lambda_{kj}}{\left(\sqrt{\mathbf{w}_k} - \sqrt{a_\delta}\right)^2} \cdot \frac{\tau_{ki}\tau_{kj}}{\mathbf{w}_k} .$$

By combining these inequalities with (3.11) and (3.27), we obtain

$$\left|(\mathbf{cov}(A_k) - \mathbf{cov}_k)_{ij}\right| \leq \frac{\lambda_{kij}}{\sqrt{\mathbf{w}_k} - \sqrt{a_\delta}} \cdot \frac{\rho_{kij}}{\sqrt{\mathbf{w}_k}} + \frac{\lambda_{ki}\lambda_{kj}}{(\sqrt{\mathbf{w}_k} - \sqrt{a_\delta})^2} \frac{\tau_{ki}\tau_{kj}}{\mathbf{w}_k} .$$

This yields the claim. □

For an interpretation of this result, we refer back to our discussion from Section 3.5.3 and to our application in Section 15.5.2.

3.6.3 Remarks

We showed that, for given a probabilistic membership matrix R, Algorithm 1 constructs hard clusters that are similar to the soft clusters defined by R. However, some aspects cannot be guaranteed.

Downsides. First, the hard clusters exhibit no locality property: Points that belong to the same hard cluster are not necessarily "close" to each other. The distance between a point and the mean of its cluster might be larger than any of the distances between this point and the means of other clusters. This means that the convex hulls of the hard clusters are not necessarily disjoint. The hard clusters themselves are pairwise disjoint, though.

Second, if the probabilistic membership matrix does not indicate a soft clustering, then the hard clusters do not necessarily cover the whole data set: We only showed that there exist pairwise disjoint hard clusters. In other words, these hard clusters do not necessarily form a hard clustering. Nevertheless, in the special case that $R \in \Delta_{N,K-1}$ is a *soft-clustering*, the hard clusters form a hard clustering (i.e., a $\dot{\cup}_{k=1}^{K} A_k = X$).

Repeated Sampling. Usually, Monte Carlo methods make use of repeated sampling. We can repeatedly run Algorithm 1 ($M \geq 2$ times) and merge the resulting hard clusters with the same index. Thereby, we obtain hard clusters $A_1, \dots, A_K$ of a data set X_M which contains M copies of each point in the given data set X (i.e., $|X_M| = M \cdot |X|$). Clearly, the corresponding probabilistic bounds on the (scaled) statistics of these hard clusters are better than the statistics of hard clusters that are computed by a single run of Algorithm 1:

Example 3.24. *Consider our bound from Lemma 3.13. Observe that* $\mathrm{E}\left[\frac{1}{M}\mathbf{w}(A_k)\right] = \mathbf{w}_k$ *and* $\mathrm{Var}\left(\frac{1}{M}\mathbf{w}(A_k)\right) = \frac{1}{M^2}\sum_{n=1}^{N} r_{nk}(1-r_{nk})w_n^2$. *Hence,*

$$\Pr\left(\left|\frac{1}{M}\mathbf{w}(A_k) - \mathbf{w}_k\right| \geq \frac{1}{\delta \cdot \mathbf{M}} \cdot \sqrt{\sum_{n=1}^{N} r_{nk}(1-r_{nk})w_n^2}\right) \leq \delta^2.$$

However, the expected size of the hard clusters increases by a factor M (cf. Corollary 2.26 and (3.7)). Assuming that we want to process the resulting hard clusters instead of the soft clusters, we have to accept that the computational costs (accordingly) increase by a factor M. Moreover, even when we repeat the assignment M times, we can not ensure that all points from X are covered with high probability (if R is no soft clustering). That is, there might still be a data point that appears in none of the clusters. In other words, the probability $(1 - (1 - \sum_{k=1}^{K} r_{nk})^M)^N$ that each of the N points from X appears in some cluster might still be very small.

Part II

Fuzzy K-Means Problems

"Is taxonomy art, or science, or both?"

Sydney Anderson[1]

Chapter 4

Introduction

Dunn (1973) was the first to present a fuzzy K-means objective function, which was later extended by Bezdek et al. (1984). Today, fuzzy K-means has found numerous practical applications, for example in image segmentation and biological data analysis, to name just a few. There has been a lot of work on theoretical analysis, extensions, variants and heuristics for the fuzzy K-means problem. However, previous to (Blömer et al., 2016), there had been no algorithm with an approximation guarantee. In this chapter, we introduce the classical fuzzy K-means problem, discuss its difficulties and flaws, and give an overview of the contribution of this thesis.

Overview. In Section 4.1, we introduce the fuzzy K-means problem and the fuzzy K-means algorithm. In Section 4.2, we discuss the difficulty behind the fuzzy K-means problem in comparison to the well-known and well analysed K-means problem. In Section 4.3 and Section 4.4, we give an overview of some related work concerning fuzzy K-means and K-means clustering in general. More specific references will be provided in the subsequent chapters. Finally, in Section 4.5, we give an overview of our contribution.

4.1 The Fuzzy K-Means Problem

In this section, we formally state the fuzzy K-means problem, describe a heuristic that is known as the fuzzy K-means algorithm, and discuss its flaws.

4.1.1 Problem Definition

In addition to the number of clusters $K \in \mathbb{N}$, the fuzzy K-means problem also requires the user to predefine the so-called fuzzifier $m \in (1,\infty)$, which determines the softness of the clustering that is sought for.

Problem 4.1 (fuzzy K-means problem)**.** *Given $X = ((x_n, w_n))_{n\in[N]} \subset \mathbb{R}^D \times \mathbb{R}_+$, $K \in \mathbb{N}$ and $m \in (1,\infty)$, the fuzzy K-means problem is to find mean vectors $C = (\mu_k)_{k\in[K]} \subset \mathbb{R}^D$ and a soft clustering $P = (p_{nk})_{n\in[N],k\in[K]} \in \Delta_{N,K-1}$ minimizing*

$$\phi_X^{(m)}(C,P) = \sum_{n=1}^{N}\sum_{k=1}^{K} p_{nk}^m w_n \left\|x_n - \mu_k\right\|_2^2 \ .$$

The fuzzifier value m determines the softness of an optimal clustering. For $m = 1$, the problem would coincide with the classical K-means problem. Usually, practitioners choose $m = 2$. We will briefly discuss and illustrate the choice of the fuzzifier value further in

[1] Source: Some Suggested Concepts for Improving Taxonomic Dialogue. Systematic Zoology, 23(1):58–70, 1974.

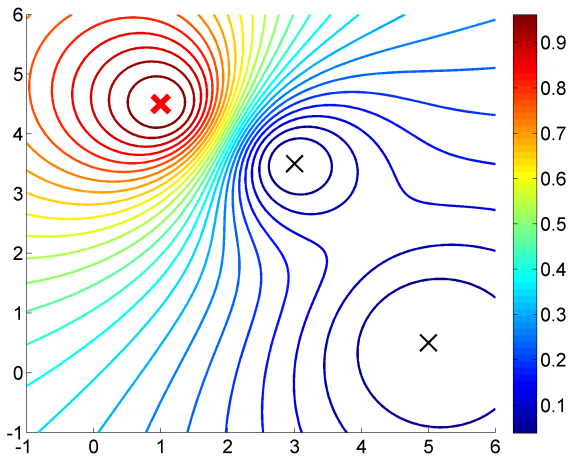


Figure 4.1: A contour plot of the soft assignment probabilities of a point to a cluster (4.1): We are given three mean vectors that are marked by crosses. We set the fuzzifier to $m = 2$. For each point $x_n \in \mathbb{R}^2$ in the plane, we evaluate the soft optimal assignment p_{n1} with respect to mean μ_1 (marked by the red bold cross).

Section 4.3.2. For the time being, note that the problem degenerates if we make the choice of $m \in (1, \infty)$ part of the problem, because $\lim_{m\to\infty} r^m \to 0$ for all $r \in [0, 1)$. Hence, m is not subject to optimization.

4.1.2 Fuzzy K-Means Algorithm

The fuzzy K-means algorithm is an alternating optimization algorithm for the fuzzy K-means problem. It is given by the first-order optimality conditions of the objective function (see Bezdek et al. (1987)):

For fixed means $(\mu_k)_{k\in[K]}$, the optimal soft clustering $(p_{nk})_{n\in[N],k\in[K]}$ satisfies

$$p_{nk} = \frac{\|x_n - \mu_k\|_2^{-\frac{2}{m-1}}}{\sum_{l=1}^{K} \|x_n - \mu_l\|_2^{-\frac{2}{m-1}}} \tag{4.1}$$

for all $k \in [K]$ and $n \in [N]$ where $\forall l \in [K] : x_n \neq \mu_l$. If x_n coincides with some mean vector, then x_n can be soft-assigned arbitrarily among these mean vectors (i.e., all μ_l with $\mu_l = x_n$). Figure 4.1 illustrates an optimal soft clustering.

For a fixed soft clustering $(p_{nk})_{n\in[N],k\in[K]}$, the optimal means $(\mu_k)_{k\in[K]}$ satisfy

$$\mu_k = \frac{\sum_{n=1}^{N} p_{nk}^m w_n x_n}{\sum_{n=1}^{N} p_{nk}^m w_n} \tag{4.2}$$

for all $k \in [K]$ with $\sum_{n=1}^{N} p_{nk}^m w_n > 0$. For all $k \in [K]$ with $\sum_{n=1}^{N} p_{nk}^m w_n = 0$, the mean μ_k may take an arbitrary form.

The fuzzy K-means algorithm alternates between updating the soft clustering and the representatives according to these formulas (until some termination criterion is met).

4.1.3 No Guarantees

Hathaway and Bezdek (1988) show that the fuzzy K-means algorithm converges to a saddle point or a minimum of the objective function. However, it is easy to see that the algorithm might converge to an arbitrarily poor solution. The following observation have been published in (Blömer et al., 2015, Observation 1) and mentioned in Blömer et al. (2016).

Figure 4.2: Illustration for the proof of Observation 4.2. Assume the fuzzy K-means algorithm is given means (μ_1, μ_2). Clearly, the algorithm should compute a soft clustering that assigns x_1 to μ_1 to the same degree to which it assigns x_4 to μ_2.

Observation 4.2. *Consider an arbitrary but fixed fuzzifier $m \in (1,\infty)$ and constant $c \in [1,\infty)$. Let $a := \sqrt{c \cdot 2^{m+2}}$. Let $X_a \in \mathrm{Dom}\left(\mathbb{R}^2, \{1\}\right)$ be the unweighted data set that contains only the four points*

$$x_1 = \begin{pmatrix} a \\ 1 \end{pmatrix},\ x_2 = \begin{pmatrix} -a \\ 1 \end{pmatrix},\ x_3 = \begin{pmatrix} -a \\ -1 \end{pmatrix},\ x_4 = \begin{pmatrix} a \\ -1 \end{pmatrix} \subseteq \mathbb{R}^2 ,$$

and let

$$I_a := (x_1,\ x_4) = \left(\begin{pmatrix} a \\ 1 \end{pmatrix},\ \begin{pmatrix} a \\ -1 \end{pmatrix} \right) \subseteq \mathbb{R}^2 .$$

Start the fuzzy K-means algorithm with $K = 2$, fuzzifier m, the data set X_a, and initial means I_a. Then, in each round, the algorithm computes a solution (C,P) with $\phi_X^{(m)}(C,P) \geq c \cdot \phi_{(X,K,m)}^{OPT}$.

Proof. The following proof is a corrected version of the proof from (Blömer et al., 2015).

First, we derive an upper bound on $\phi_{(X_a,2,m)}^{OPT}$. To this end, consider the means $C = ((-a,0)^T,(a,0)^T)$. Let Z be the hard clustering of X that assigns x_2 and x_3 to the first cluster and x_1 and x_4 to the second cluster. Then, $\phi_{X_a}^{(m)}(C,Z) = 4$. Hence, $\phi_{(X_a,2,m)}^{OPT} \leq 4$.

Second, consider arbitrary means $C = ((s,t)^T,(s,-t)^T) \subseteq \mathbb{R}^2$ with $s \in [-a,a]$ and $t \in [-1,1]$. Denote the single means by $\mu_1 := (s,t)^T$ and $\mu_2 := (s,-t)^T$. This setting is illustrated in Figure 4.2. It is obvious that there are certain symmetries and that the fuzzy K-means algorithm should compute a soft clustering that assigns x_1 to μ_1 to the same degree to which it assigns x_4 to μ_2. Formally, observe that $\|x_1 - \mu_1\|_2 = \|x_4 - \mu_2\|_2$, $\|x_1 - \mu_2\|_2 = \|x_4 - \mu_1\|_2$, $\|x_2 - \mu_1\|_2 = \|x_3 - \mu_2\|_2$, and $\|x_3 - \mu_1\|_2 = \|x_2 - \mu_2\|_2$. Then, from (4.1), we can conclude that, given C, the fuzzy K-means algorithm computes a soft clustering $(p_{nk})_{n,k}$ with $p_{11} = p_{42}$, $p_{21} = p_{32}$, $p_{31} = p_{22}$, and $p_{41} = p_{12}$. Then, given these probabilities, the algorithm computes new mean vectors $\tilde{C} = ((u,p)^T,(v,q)^T)$ according to (4.2). From (4.2) and the specific form of the soft clustering, we can conclude that

$$u = \frac{p_{11}^m a - p_{21}^m a - p_{31}^m a + p_{41}^m a}{p_{11}^m + p_{21}^m + p_{31}^m + p_{41}^m} = \frac{p_{42}^m a - p_{32}^m a - p_{22}^m a + p_{12}^m a}{p_{42}^m + p_{32}^m + p_{22}^m + p_{12}^m} = v$$

and

$$p = \frac{p_{11}^m + p_{21}^m - p_{31}^m - p_{41}^m}{p_{11}^m + p_{21}^m + p_{31}^m + p_{41}^m} = \frac{p_{42}^m + p_{32}^m - p_{22}^m - p_{12}^m}{p_{42}^m + p_{32}^m + p_{22}^m + p_{12}^m} = -q .$$

That is, $\tilde{C}$ takes the form $\tilde{C} = ((u,p)^T,(u,-p)^T) \subseteq \mathbb{R}^2$ for some $u \in [-a,a]$ and $p \in [-1,1]$. Consequently, started with the initial solution I_a, *each* round of the fuzzy K-means algorithm results in mean vectors that take the form $((s,t)^T,(s,-t)^T) \subseteq \mathbb{R}^2$ for some $s \in [-a,a]$ and $t \in [-1,1]$.

Next, we lower bound the cost of such means. Consider $C = ((s,t)^T,(s,-t)^T) \subseteq \mathbb{R}^2$ with $s \in [-a,a]$ and $t \in [-1,1]$. There are always at least 2 points in X_a that have Euclidean distance at least a from both means. Without loss of generality, we can assume that these

points are x_2 and x_3. Denote by $(p_{nk})_{n,k}$ the optimal probabilities, for the fixed means C. For all $n \in [4]$, we have $p_{n1} + p_{n2} = 1$ and, hence, $\sum_{k=1}^{2} p_{nk}^m \geq \left(\frac{1}{2}\right)^m$. Combining these bounds gives

$$\phi_X^{(m)}(C) = \sum_{n=1}^{4} \sum_{k=1}^{2} p_{nk}^m \left\| x_n - \tilde{\mu}_k \right\|_2^2 \geq \sum_{k=1}^{2} p_{2k}^m a^2 + \sum_{k=1}^{2} p_{3k}^m a^2 \geq \frac{a^2}{2^{m-1}} \geq c \cdot \phi_{(X_a,2,m)}^{OPT} ,$$

where in the last inequality we use the fact that $\phi_{(X_a,2,m)}^{OPT} \leq 4$, which we proved in the beginning. This yields the claim. □

4.2 A Comparison with the K-Means Problem

The fuzzy K-means problem can be seen as a generalization of the K-means problem, which is well-analysed and for which there are numerous approximation algorithms.

Problem 4.3 (K-means problem)**.** *Given a data set $X = ((x_n, w_n))_{n\in[N]} \subset \mathbb{R}^D \times \mathbb{R}_+$ and a number of clusters $K \in \mathbb{N}$, the K-means problem is to find means $C = (\mu_k)_{k\in[K]} \subset \mathbb{R}^D$ minimizing*

$$\mathrm{km}_X(C) := \sum_{n=1}^{N} w_n \min\left\{ \left\| x_n - \mu_k \right\|_2^2 \,\middle|\, k \in [K] \right\} .$$

We denote the cost of an optimal solution by $\mathrm{km}_{(X,K)}^{OPT}$.

Observe that the minimum used in the objective function implicitly models a hard clustering. This means that the K-means problem is a combinatorial problem and, hence, there exist exact algorithms. Besides that, there are numerous approximation algorithms. We will an overview of these algorithms and related work in Section 4.4.

4.2.1 Similarities

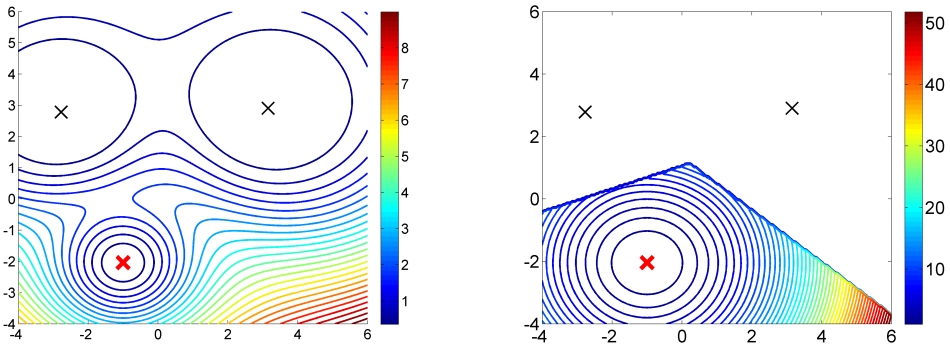
Despite their obvious differences, there are several similarities between the fuzzy K-means and the K-means problem.

Distances. The objective functions are similar in the sense that both are sums of squared Euclidean distances between points and mean vectors, weighted by terms that depend only on (soft) assignments. This enables us to use a key result regarding the K-means problem, namely Lemma 2.20 (cf. Inaba et al. (1994)). We study the relation between the objective functions more closely in Section 6.1.

Locality. In both clustering approaches, a point belongs more to means that are close than belonging to means that are farther away: Consider some fixed means $C = (\mu_k)_{k\in[K]}$. A K-means clustering assigns a data point solely to the cluster whose mean is closest. A fuzzy K-means clustering assigns a data point to each cluster with some (possibly arbitrarily small but) positive[1] probability (4.1). However, it assigns each point x_n to closer means μ_k more than to means μ_l far away since

$$p_{nk} = \frac{\left\| x_n - \mu_k \right\|_2^{-\frac{2}{m-1}}}{\sum_{o=1}^{K} \left\| x_n - \mu_o \right\|_2^{-\frac{2}{m-1}}} > \frac{\left\| x_n - \mu_l \right\|_2^{-\frac{2}{m-1}}}{\sum_{o=1}^{K} \left\| x_n - \mu_o \right\|_2^{-\frac{2}{m-1}}} = p_{nl} \quad \Leftrightarrow \quad \left\| x_n - \mu_k \right\|_2 < \left\| x_n - \mu_l \right\|_2 .$$

[1] In the following, we ignore the special case where a point coincide with some mean vector.



(a) fuzzy K-means ($m = 2$)

(b) K-means

Figure 4.3: Contribution of a point to the cost of a single cluster represented by μ_1 (marked by the red bold cross). Figure 4.3a depicts the fuzzy K-means cost: For each $x_n \in \mathbb{R}^2$, we evaluate the cost $p_{n1}^2 \left\|x_n - \mu_1\right\|_2^2$ of x_n with respect to mean μ_1 , where p_{n1} is the optimal assignment for the given means (cf. Figure 4.1 and (4.1)). Figure 4.3b depicts the K-means cost: For each $x_n \in \mathbb{R}^2$, we evaluate $\left\|x_n - \mu_1\right\|_2^2$ with respect to mean μ_1, if x_n is not closer to any other mean vector.

Special Case. The K-means problem is a limit case of the fuzzy K-means problem with fuzzifier $m = 1$ (Huang et al., 2012):

Observation 4.4. *Consider* $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ *and* $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$. *Each hard clustering* $Z = (z_{nk})_{n\in[N],k\in[K]}$ *that satisfies*

$$\forall n \in [N]\ \forall k \in [K] \colon\ z_{nk} = 1 \ \Rightarrow\ k \in \arg\min\left\{\left\|x_n - \mu_l\right\|_2 \,\middle|\, l \in [K]\right\}$$

minimizes $\phi^{(1)}(C, \cdot)$ *with respect to all soft* K*-clusterings of* X.

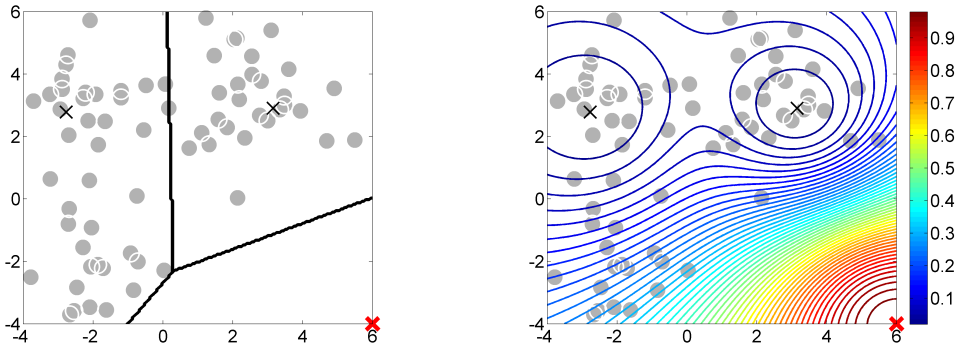
That is, optimal mean vectors for the K-means problem are also mean vectors of an optimal solution to the fuzzy K-means problem with $m = 1$ and vice versa. In other words, the fuzzy K-means problem is a generalization of the K-means problem.

4.2.2 Differences

In comparison to a K-means (hard) clustering, a fuzzy K-clustering is utterly soft.

Continuity of Assignments. Consider a fixed data set X and some fixed means C. Let P be the optimal soft clustering of X, given the fixed means C. Now change the position x_n of a single data point a little bit. From the optimality conditions (4.1) we see that (for the still fixed means C) each optimal assignment p_{nk} of this data point x_n is influenced. This is different from a K-means clustering. Assume x_n is closest to the k-th mean vector. As long as we do not choose a position that is closer to one of the other means than to the k-th mean, we can move x_n without changing its optimal hard assignment.

Wide-Ranging Assignments. Figure 4.3 illustrates the fact that in a classical fuzzy K-means clustering each data point is assigned to each cluster. The figure shows three mean vectors (marked by crosses) in the plane. Consider the mean vector marked by the bold red cross, say μ_1. A hard assignment minimizing the K-means cost assigns only those points to μ_1 which are closer to (or, at last, not farther away from) μ_1 than to any of the other two mean vectors. Hence, all other data points do not contribute to the cost of the cluster represented by μ_1 at all. In contrast, *each* data point contributes to the fuzzy K-means cost of this cluster.



(a) K-means (b) fuzzy K-means ($m = 2$)

Figure 4.4: Example data set (points marked by circles) and three mean vectors (marked by crosses). Figure 4.3b depicts the optimal partition of the data set induced by the K-means cost of the given means. Figure 4.4b depicts the optimal assignments to μ_1 (marked by the red bold cross): For each $x_n \in \mathbb{R}^2$, we evaluate the optimal assignment p_{n1} (4.1).

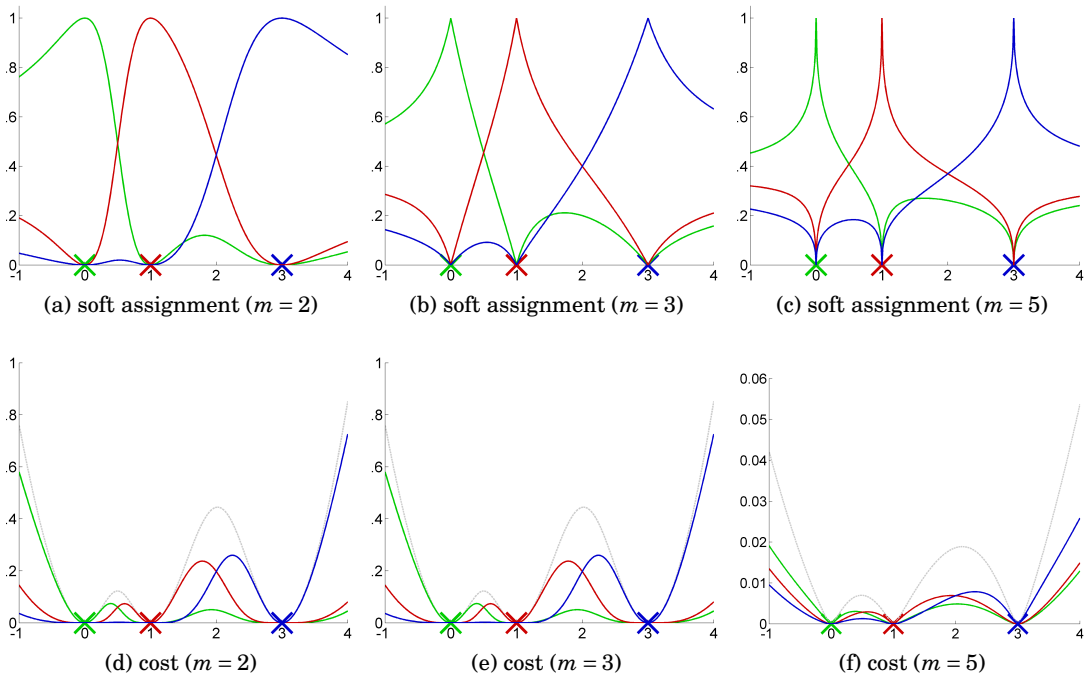
An Extreme Example. The discrepancy between fuzzy K-means and K-means clustering is most striking in the, admittedly, extreme case where a mean vector is far away from all the given data points. Figure 4.4 illustrates this case. Here, we are given three mean vectors (marked by crosses) and data points (marked by circles) in the plane. The mean vector marked by the bold red cross, say μ_1, is much farther away from all the data points than the other mean vectors. A soft assignment (4.1) that minimizes the fuzzy K-means cost (with fuzzifier $m = 2$) assigns roughly 8% of the membership mass to μ_1. The resulting soft cluster contributes more than 16% to the overall cost of this solution. In contrast, for the fixed means, the corresponding optimal K-means clustering assigns none of the points to the cluster represented by μ_1. It does not even matter if we move this mean vector, as long as we keep it far away from all points. In contrast, moving this cluster mean changes all optimal fuzzy K-means soft assignments (4.1).

4.2.3 Statistical Assumptions

There is no statistical assumption behind the fuzzy K-means problem. Thus, the fuzzy K-means problem is inherently different from soft clustering problems based on statistical models, such as the so-called soft K-means problem (Mackay, 2003). It also means that the nice statistical concept of consistency is not applicable here (see e.g. (Wald, 1949)). For instance, Hathaway and Bezdek (1988) report that, unsurprisingly, optimal fuzzy K-means solutions do not provide consistent estimators for normal mixtures.

In contrast, the K-means problem exhibits similarities to a soft clustering problem based on statistical models: The K-means problem is related to the maximum likelihood estimation problem for Gaussian mixture models (e.g. see Kearns et al. (1997)). Lloyd's K-means algorithm, which is the analogon of the fuzzy K-means algorithm, can be interpreted as a limit of the expectation-maximization (EM) algorithm for Gaussian mixture models (with fixed equal weights and for covariances converging to the zero matrix (see (Lloyd, 1982) and (Bishop, 2006, pp. 443)).

In Section 4.3.3 we give a brief overview of some variants of the fuzzy K-means problem that are inspired by the estimation of Gaussian mixture models. They do not close the gap of the missing statistical assumption, though.



(a) soft assignment ($m = 2$) (b) soft assignment ($m = 3$) (c) soft assignment ($m = 5$)

(d) cost ($m = 2$) (e) cost ($m = 3$) (f) cost ($m = 5$)

Figure 4.5: Illustration of the impact of the fuzzifier values $m \in \{2,3,5\}$:
We are given the 3 mean points $\mu_1 = 0$, $\mu_2 = 1$, and $\mu_3 = 3$ in $\mathbb{R}$. For each point $x_n \in [-1,4]$ and each $k \in [3]$, we evaluate the optimal soft assignments p_{nk} for the given means and the resulting cost $p_{nk}^m \|x_n - \mu_k\|_2^2$ per cluster (4.1). Figure 4.5a through Figure 4.5c depict the optimal soft assignments, while Figure 4.5d through Figure 4.5f depict the resulting costs per cluster and the overall cost (gray).

4.3 Related Work

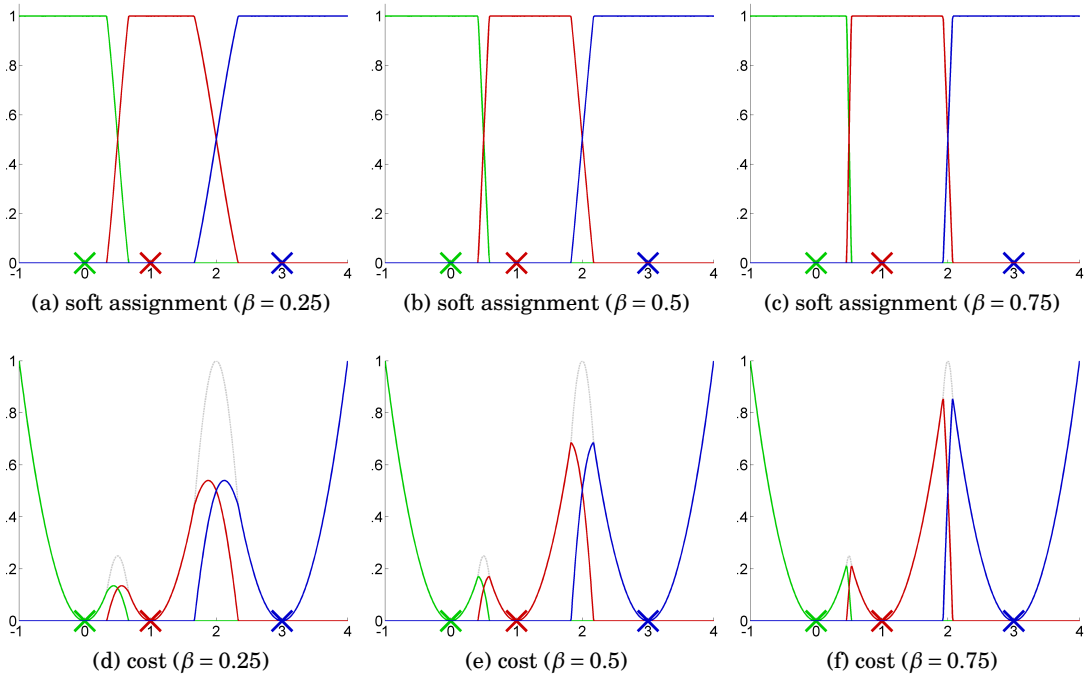
In the following, we give a brief overview of some work related to the fuzzy K-means problem and algorithm. More information can be found in (Oliveira and Pedrycz, 2007), (Höppner, 1999), and (Yang, 1993), for instance.

4.3.1 The Fuzzy K-Means Algorithm

Bezdek et al. (1984, 1987) prove convergence of the fuzzy K-means algorithm to a local minimum or a saddle point of the objective function. Among others, Höppner and Klawonn (2003) and Kim et al. (1988) address the problem of determining and distinguishing whether the algorithm has reached a local minimum or a saddle point. Furthermore, Hathaway and Bezdek (1986) show that the algorithm converges locally. That is, started sufficiently close to a minimizer, the iteration sequence converges to that particular minimizer. Hathaway and Bezdek (2001) generalize the fuzzy K-means algorithm in the sense that they replace the use of the Euclidean distance by an L_p (semi-)norm. Hu and Hathaway (2002) compare the fuzzy K-means algorithm with different general-purpose minimization methods (for instance, Newton-type methods) and conclude that the fuzzy K-means algorithm is the best and simplest method.

4.3.2 Fuzzifier

Which m Should We Choose? Recall that in our formulation of the fuzzy K-means problem, the fuzzifier m is some fixed constant. Figure 4.5 illustrates the significant effect of



(a) soft assignment ($\beta = 0.25$) (b) soft assignment ($\beta = 0.5$) (c) soft assignment ($\beta = 0.75$)

(d) cost ($\beta = 0.25$) (e) cost ($\beta = 0.5$) (f) cost ($\beta = 0.75$)

Figure 4.6: Illustration of the impact of the fuzzifier function s_β with $\beta \in \{0.25, 0.5, 0.75\}$: We are given the 3 mean points $\mu_1 = 0$, $\mu_2 = 1$, and $\mu_3 = 3$ in $\mathbb{R}$. For each point $x_n \in [-1, 4]$ and each $k \in [3]$, we evaluate the optimal soft assignments p_{nk} for the given means and the resulting cost $\mathsf{s}_\beta(p_{nk}) \|x_n - \mu_k\|_2^2$ per cluster (see Section 5.3.3). Figure 4.6a through Figure 4.6c depict the optimal soft assignments, while Figure 4.5d through Figure 4.5f depict the resulting costs per cluster and the overall cost (gray).

changing the value of the fuzzifier. The question remains which fuzzifier $m \in (1, \infty)$ should be chosen in practice. Often, practitioners just fix $m = 2$. However, there has been a lot of work on this topic. As a starting point for more information, we refer to Huang et al. (2012).

Relax? There is a well-known variant of the fuzzy K-means problem that imposes a relaxation on our notion of soft assignments. The restriction that the soft assignments form a probability distribution (i.e., sum up to 1) is removed. Instead, a regularization term is added to the objective function which ensures that setting all assignments to zero is not the best solution. This approach is known as possibilistic clustering. A critical discussion of this approach can be found in (Pal et al., 2005; Timm et al., 2004), for instance.

A Different Kind of Fuzziness? Klawonn and Höppner (2003) and Klawonn (2004) consider several alternatives to the exponentiation p_{nk}^m of the p_{nk}. They propose to use

$$\mathsf{s}_\beta(p) := \frac{1-\beta}{1+\beta} p^2 + \frac{2\beta}{1+\beta} p \ , \tag{4.3}$$

with some fixed $\beta \in [0, 1]$. That is, they replace the term p_{nk}^m in the objective function by the term $\mathsf{s}_\beta(p_{nk})$. Supported by some experiments, they claim that their fuzzifier function can help to overcome the undesired effect that all data tend to influence all clusters (Klawonn and Höppner, 2003, p. 10). Figure 4.6 illustrates this claim. A further alternative proposed by Klawonn (2004) is to use the exponential fuzzifier function

$$\mathsf{e}_\gamma(p) := \frac{e^{\gamma \cdot p} - 1}{e^\gamma - 1} \ , \tag{4.4}$$

with some fixed $\gamma \in [0, \infty)$. We discuss both alternative fuzzifier functions in Section 5.2.

4.3.3 Extensions

There are numerous extensions and variants of the fuzzy K-means problem and the fuzzy K-means algorithm (Höppner, 1999; Yang, 1993). Let us briefly describe two well-known extensions.

Gustafson and Kessel (1978) extend the fuzzy K-means problem such that it takes into account clusters of different geometric shapes. To this end, they replace the squared Euclidean distance by an Mahalanobis distance. That is, the representative of each cluster is extended by a covariance matrix $\Sigma_k \in \mathbb{R}^{D\times D}$ and the objective function becomes

$$\psi_X^{(m)}\left(\left((\mu_k,\Sigma_k)\right)_{k\in[K]},(p_{nk})_{n,k}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} w_n p_{nk}^m (x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k)\ .$$

Since this objective function is linear in the precision matrix Σ_k^{-1}, one needs to add additional constraints. To this end, they fix the determinant of each covariance: Given parameters $\lambda_1,\dots,\lambda_K \in \mathbb{R}_+$, which have to be chosen in advance, they add the constraints $|\Sigma_k| = \lambda_k$, $k \in [K]$. This means that the covariance matrix is allowed to be non-spherical (i.e., scale differently along each direction). Yet, it cannot scale arbitrarily as its volume is fixed.

Gath and Geva (1989) extend the Gustafson-Kessel algorithm further by using notions from maximum likelihood estimation for Gaussian mixture model: They enhance the representation of a cluster by a covariance and a weight parameter and use an exponential distance measure. That is, the representative of each cluster consists of a weight $w_k \in [0,1]$, a mean $\mu_k \in \mathbb{R}^D$, and a covariance $\Sigma_k \in \mathbb{R}^{D\times D}$ and the objective function becomes

$$\xi_X^{(m)}\left(\left((w_k,\mu_k,\Sigma_k)\right)_{k\in[K]},(p_{nk})_{n,k}\right) = \sum_{n=1}^{N}\sum_{k=1}^{K} p_{nk}^m \frac{\sqrt{|\Sigma_k|}}{w_k}\exp\left(\frac{1}{2}(x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k)\right)\ .$$

Note that though this cost function resembles the likelihood of a Gaussian mixture, it is not the likelihood of a mixture model. For more information, we refer to Höppner (1999).

4.4 More Related Work (The K-Means Problem)

For the K-means problem, there are numerous algorithms with performance guarantees.

4.4.1 The Bad News First

To get a coarse idea of the runtimes and the approximation guarantees of these algorithms, take note of the following facts: The K-means problem is NP-hard for fixed K (even for $K=2$) (Dasgupta, 2008), and it is also NP-hard for fixed D (even for $D=2$) (Mahajan et al., 2012). Recently, Lee et al. (2017) showed that it is even NP-hard to approximate K-means within a factor 1.0013. In other words, the problem is APX hard and there is no polynomial-time approximation scheme (PTAS), given that K and D are not considered to be constants. In case K is considered a constant, there is a PTAS for the K-means problem (e.g. Feldman et al. (2007)). In case D is considered a constant, to the best of our knowledge it is not known whether there is a PTAS.

4.4.2 (Few Practical) Approximation Algorithms

An overview of approximation algorithms for the K-means problem can be found in (Schmidt, 2014, p. 15), for instance. In the following, we just give a brief overview of some algorithms that are important in the remainder of this thesis. We state all runtimes with respect to unweighted data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$: First of all, Hasegawa et al. (1993) showed that there is a very simple 2-approximation algorithm with runtime $\mathcal{O}(|X|^{K+1}\cdot KD)$. Shortly after,

Inaba et al. (1994) showed that there is an exact algorithm with running time $\mathcal{O}(|X|^{DK+1})$. The algorithm of Matoušek (2000) yields a $(1+\epsilon)$-approximation in time $\mathcal{O}(|X|\log(|X|)^K\epsilon^{-2D})$ (their algorithm was later improved by Har-Peled and Mazumdar (2004)). Moreover, there is the famous K-means++ algorithm by Arthur and Vassilvitskii (2007). It is a randomized algorithm that computes, in expectation, an $\mathcal{O}(\log(K))$-approximation in time $\mathcal{O}(|X|KD)$. Aggarwal et al. (2009) show that the L-means cost of $L = \lceil 16(K+\sqrt{K})\rceil$ mean vectors that have been generated according to the K-means++ algorithm is, with constant probability, at most a constant factor 20 worse than the cost of an optimal solution to the K-means problem, which contains K mean vectors. Moreover, based on the same idea, they derive an algorithm that, with constant probability, computes a constant-factor approximation for the K-means problem and needs time $\mathcal{O}(|X|KD + \text{poly}(\log(|X|, K)))$. Recently, Wei (2016) improved this result and showed that it suffice to generate $\lceil 1.2 \cdot K\rceil$ means to obtain a factor of 10, with the same probability of success. Bachem et al. (2016) proposed a faster variant of the K-means++ algorithm that can be used for massive data sets and that still produces good clusterings in practise.

4.4.3 Clustering is Difficult – Except when It Is Not

When we apply a certain clustering algorithm, then we implicitly make assumptions: We assume that the given data is "clusterable" and we assume that the optimal clustering, which the clustering algorithm searches for, is meaningful to us. For an introduction to this topic, we refer to Ackerman and Ben-David (2009) and Ben-David and Reyzin (2014). Yet, there is no formalization that is commonly agreed upon.

For instance, Awasthi et al. (2010b) formalize assumptions on optimal solutions. Under their assumptions, the K-means problem simplifies significantly: They state that there is a polynomial-time approximation scheme (PTAS) for instances where a subset of $K-1$ means of an optimal K-means solution is at least a factor $(1+\alpha)$ costlier than a (complete) optimal K-means solution, for some constant α. This approach has some similarity to the so-called elbow method (Tibshirani et al., 2001), which is used to decide whether some number L is the "right" number of clusters. Yet, Ben-David (2015) criticize that these assumptions are much too strict. The authors themselves discuss their assumptions in (Awasthi et al., 2010a) in comparison to the assumptions made by Ostrovsky et al. (2006), for example.

Another example is the work of Tang and Monteleoni (2016). They present a new initialization algorithm for Lloyd's algorithm under a clusterability assumption. More precisely, their assumption is that, in the sought clustering, the mean vectors of two different clusters have a certain minimum distance, which depends on the K-means cost of the sought clustering and the number of points in each cluster.

4.4.4 Constraints and Side Information

The notion of clusterability, which we discussed in the last section, can be seen as a way to incorporate additional knowledge. Two approaches that incorporate knowledge in a more direct manner are known as semi-supervised clustering and constrained clustering. An example for semi-supervised clustering is the work of Wagstaff et al. (2001). They incorporate background knowledge on the input data set which takes the form of must-links and cannot-links that determine if two data points must or cannot be in the same cluster. An example for constraint clustering is the work of Ding and Xu (2015). They consider various versions of the K-means problem where the space of solutions is constrained. For instance, they consider the r-gather clustering problem where each cluster must contain at least r points.

4.5 Overview

In the following, we abstract from the classical fuzzy K-means problem. That is, we consider a generalized version where we replace the exponentiation p_{nk}^m of the soft assignment probabilities p_{nk} by the evaluation $r(p_{nk})$ of a fixed fuzzifier function r. We call the resulting problem the r-fuzzy K-means problem. The benefit of this approach is threefold: First, the abstraction helps to gain a better insight into the properties of this problem. Second, it improves the reusability of our results. Third, it helps to check whether our results generalize to the fuzzifier function proposed by Klawonn and Höppner (2003).

The following chapters are organized as follows.

Chapter 5 introduces the r-fuzzy K-means problem formally and discusses its basic properties. We discuss properties that a reasonable fuzzifier function r should have and identify additional useful properties.

Chapter 6 deals with two basic techniques that will be very helpful throughout the following chapters: Section 6.1 relates the r-fuzzy K-means cost function to the classical K-means cost function. Section 6.2 introduces the notion of negligible fuzzy clusters. Simply speaking, we show that fuzzy clusters with too small a weight can be ignored if we allow the total cost to worsen by a small factor.

Chapter 7 gives an overview of some simple algorithms for the r-fuzzy K-means problem. We show that there are large-factor approximation algorithms and that there is a simple but extremely slow $(1+\epsilon)$-approximation algorithm.

Chapter 8 shows how our results from Chapter 3 can be applied. We combine our soft-to-hard-cluster technique with the so-called superset sampling technique, which is well known from the K-means clustering problem.

Chapter 9 comprises some very technical results that are used in the following two chapters. We analyse a technique that has been introduced by Chen (2009) in a general way. In simple terms, this construction can be used to construct a small discrete search space, which we will do in Chapter 10, or to discretize the input points, which is a very helpful trick that we will use (in the proofs presented) in Chapter 12.

Chapter 10 proposes a $(1+\epsilon)$-approximation algorithm for the r-fuzzy K-means problem. It uses the technique from Chapter 9 to discretize the search space in such a way that the resulting discrete search space is guaranteed to contain a $(1+\epsilon)$-approximation.

Chapter 11 shows that the well-known Johnson-Lindenstrauss lemma can be applied to the r-fuzzy K-means problem. We show that it can be used to improve our results from Chapter 10. Moreover, we discuss further dimension reduction techniques.

Chapter 12 presents a coreset construction for the r-fuzzy K-means problem. A coreset is a small representation of a data set that preserves a certain property of the original data set. Again, we make use of the construction introduced by Chen (2009), which we analysed in Chapter 9.

Chapter 13 contains a summary of our techniques and main results. Furthermore, discuss our results and suggest directions for further work.

"The purpose of computation is insight, not numbers."

Richard Hamming[1]

Chapter 5

Basics

In this chapter, we introduce a generalized version of the fuzzy K-means problem where we replace the exponentiation of a soft assignment probability by the evaluation of a fuzzifier function r. We call this problem the r-fuzzy K-means problem. Our intention behind this abstraction is to gain a better insight into the properties of the problem, improve the reusability of our results, and to generalize our results.

Overview. In Section 5.1, we formalize the problem. In Section 5.2, we focus on the fuzzifier function r. We discuss necessary as well as additional useful properties. Moreover, we analyse the form that optimal soft clusterings take. Finally, in Section 5.3, we get back to the different fuzzifier functions that we already described in Section 4.1 and Section 4.3.2 and analyse their properties.

Publications. In this chapter, we present unpublished ongoing work.

5.1 Problem Definition

The r-fuzzy K-means problem corresponds to the classical fuzzy K-means problem where we replace the exponentiation of the soft assignments p_{nk} by the evaluation $r(p_{nk})$ of a fuzzifier function r.

5.1.1 Cost and Clusters

In the r-fuzzy K-means problem we want to minimize the following objective function:

Definition 5.1 (r-fuzzy K-means cost)**.** *For each $K \in \mathbb{N}$, $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, and function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, we let*

$$\phi_X^{(r)}\left((\mu_k)_{k\in[K]}, P\right) := \sum_{n=1}^{N}\sum_{k=1}^{K} r(p_{nk}) w_n \left\|x_n - \mu_k\right\|_2^2$$

be the r-fuzzy K-means cost of X with respect to the mean vectors $(\mu_k)_{k\in[K]} \subset \mathbb{R}^D$ and the soft K-clustering $P = (p_{nk})_{n\in[N],k\in[K]}$.

A soft K-clustering $(p_{nk})_{n,k} \in \Delta_{N,K-1}$ assigns the n-th data point to the k-th cluster with probability p_{nk} (Notation 2.7). We call the values $r(p_{nk})$ *fuzzified soft assignments*. Analogously, to the soft clusters $A_k^{(X,P)} = ((x_n, p_{nk} \cdot w_n))_{n\in[N]}$ of X given by P (Notation 2.7), we define r-fuzzy clusters as follows.

[1] Source: Hamming, Richard (1962). Numerical Methods for Scientists and Engineers. New York: McGraw-Hill. ISBN 0-486-65241-6.

Definition 5.2 (r-fuzzy cluster)**.** *For all $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $P = (p_{nk})_{n\in[N],k\in[K]} \in \Delta_{N,K-1}$, let*

$$r(P) := (r(p_{nk}))_{n\in[N],k\in[K]} \ .$$

For all $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ and $k \in [K]$, we call

$$A_k^{(X,r(P))} = ((x_n, r(p_{nk}) \cdot w_n)_{n\in[N]} \ .$$

the k-th r-fuzzy cluster of X given by P.

With this notation and our definitions from Section 2.3, we can identify

$$\phi_X^{(r)}((\mu_k)_{k\in[K]}, P) = \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,r(P))}, \mu_k\right) \quad \text{and} \quad \sum_{n=1}^{N} r(p_{nk}) w_n = \mathbf{w}\left(A_k^{(X,r(P))}\right) \ . \tag{5.1}$$

5.1.2 Induced Solutions

In this section, we consider an alternating optimization of the r-fuzzy K-means cost function.

Notation 5.3 (induced solutions)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $K \in \mathbb{N}$, and $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$.*
We call $C^* \subseteq \mathbb{R}^D$ r-fuzzy means of X induced by $P \in \Delta_{|X|,K-1}$ if

$$C^* \in \arg\min \left\{ \phi_X^{(r)}(C, P) \;\middle|\; C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D \right\} \ .$$

We call $P^* \in \Delta_{|X|,K-1}$ an r-fuzzy K-clustering of X induced by $C \subseteq \mathbb{R}^D$, $|C| = K$, if

$$P^* \in \arg\min \left\{ \phi_X^{(r)}(C, P) \;\middle|\; P \in \Delta_{|X|,K-1} \right\} \ .$$

If the respective induced solutions exist, we use the short notation

$$\phi_X^{(r)}(P) := \min \left\{ \phi_X^{(r)}\left(C^*, P\right) \;\middle|\; C^* = (\tilde{\mu}_k)_{k\in[K]} \subseteq \mathbb{R}^D \right\} \text{ and}$$
$$\phi_X^{(r)}(C) := \min \left\{ \phi_X^{(r)}\left(C, P^*\right) \;\middle|\; P^* \in \Delta_{|X|,K-1} \right\} \ .$$

For a fixed soft clustering, the induced r-fuzzy means can be computed easily:

Lemma 5.4 (induced r-fuzzy means)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $P \in \Delta_{|X|,K-1}$. Then, r-fuzzy means $(\mu_k)_{k\in[K]}$ of X induced by P exist. In particular, for all $k \in [K]$ with $\mathbf{w}(A_k^{(X,r(P))}) > 0$, the mean μ_k satisfies*

$$\mu_k = \mathbf{m}\left(A_k^{(X,r(P))}\right) \ . \tag{5.2}$$

Proof. The claim directly follows from (5.1) and Lemma 2.20. □

Given some means and an arbitrary black-box function r, it is obviously not clear how an induced r-fuzzy clustering can be computed. In the following, we focus on continuous functions r.

Lemma 5.5 (induced r-fuzzy clustering)**.** *Let $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ and let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a function. Let $C \subseteq \mathbb{R}^D$ be a vector of K means.*

1. *If r is continuous, C induces an r-fuzzy K-clustering.*
2. *The set of r-fuzzy clusterings that are induced by C is not necessarily singleton.*

Proof. Observe that the simplex $\Delta_{K-1} \subseteq \mathbb{R}^K$ is closed and bounded. Hence, the Cartesian product $(\Delta_{K-1})^N \subseteq (\mathbb{R}^K)^N$ is closed and bounded. Since r is continuous and bounded, the objective function $f_C\ (\Delta_{K-1})^N \to \mathbb{R}$ with $f_C(P) = \phi_X^{(r)}(C,P)$ (where we interpret $(\Delta_{K-1})^N$ as $\Delta_{N,K-1} \subseteq \mathbb{R}^{N\times K}$ on the righthand side) is continuous and bounded. From the extreme value theorem (Stewart, 2009, p. 964), we know that f_C attains a minimum on $(\Delta_{K-1})^N$. This yields Item 1.

Consider a set of mean vectors containing two means μ_k and μ_l, $k \neq l$, that coincide with one of the data points $x_n \in X$. Then we have $r(p_{nk})w_n \left\|x_n - \mu_k\right\|_2^2 + r(p_{nl})w_n \left\|x_n - \mu_l\right\|_2^2 = 0$ for *all* $p_{nk}, p_{nl} \in [0,1]$. This yields Item 2. □

Even though we know that induced r-fuzzy clusterings exist, we do not know how to compute them yet. We will we deal with this question in Section 5.2.6 and Section 5.3.

5.1.3 Approximation

Lemma 5.4 and Lemma 5.5 imply the existence of an optimal solution.

Definition 5.6 (optimal)**.** *For all $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $K \in \mathbb{N}$, and continuous $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, we denote the cost of an optimal solution to the r-fuzzy K-means problem with respect to X by*

$$\phi_{(X,K,r)}^{OPT} := \min\left\{\phi_X^{(r)}(C,P) \;\middle|\; C \subseteq \mathbb{R}^D \text{ with } |C| = K,\ P \in \Delta_{|X|,K-1}\right\} .$$

In the following, we aim to find an approximate solution.

Problem 5.7 ($(1+\epsilon)$-approximation)**.** *Given $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, a continuous function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $K \in \mathbb{N}$, and $\epsilon \in (0,1]$, the r-fuzzy K-means $(1+\epsilon)$-approximation problem is to find mean K vectors $C \subset \mathbb{R}^D$ and a soft K-clustering $P \in \Delta_{N,K-1}$ such that*

$$\phi_X^{(r)}(C,P) \leq (1+\epsilon) \cdot \phi_{(X,K,r)}^{OPT} .$$

Observe that this problem has the following reasonable properties: First, the approximation factor is shift invariant: If we shift the data points by adding the same constant vector $c \in \mathbb{R}^D$ to each point x_n, then the value of the objective function does not change. Second, the approximation factor is scale invariant: If we scale all points x_n and means μ_k by a constant factor $c > 0$, then the value of the objective function changes by the very same factor c. The same happens if we scale the weights w_n. In Part III, we will deal with an objective function that does *not* fulfill the latter property.

5.2 Fuzzifier Functions

In this section, we focus on the function r. First, we propose fundamental constraints and some additional useful constraints. After that, we aim to specify induced r-fuzzy clusterings further. In the next Section 5.3, we present some concrete fuzzifier functions r.

5.2.1 Definition

We claim that the function r should satisfy the following properties.

Definition 5.8 (fuzzifier)**.** *A function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a fuzzifier function if*

1. *r is continuous,*
2. *$r(0) = 0$ and $r(1) = 1$, and*
3. *r is strictly increasing.*

This definition describes some of the very basic properties of the function $r(p) = p^m$, $m \in \mathbb{N}$, which is used in the classical fuzzy K-means problem. In the next section, we show that these basic properties ensure that a fuzzy K-means clustering has some similarity to a K-means clustering.

5.2.2 Basic Properties

In this section, we show that an r-fuzzy clustering with a fuzzifier function r has still some similarity to a K-means clustering. We already discussed some of the following properties with respect to the classical fuzzy K-means problem in Section 4.2.1.

Locality. We demand that a fuzzifier function is strictly increasing. This helps to preserve a locality property, which we already described in Section 4.2.1:

Lemma 5.9 (locality)**.** *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, and $(x_n, w_n) \in \mathbb{R}^D \times \mathbb{R}_+$. Let $(p_{nk})_{k\in[K]} \in \Delta_{K-1}$ be an r-fuzzy clustering of (x_n, w_n) induced by C.*

Then, for all $l, k \in [K]$, we have

$$\left\|x_n - \mu_k\right\|_2 < \left\|x_n - \mu_l\right\|_2 \quad \Rightarrow \quad p_{nk} \geq p_{nl} \ .$$

Proof. Consider some $\mu_l, \mu_k \in C$ with $\left\|x_n - \mu_k\right\|_2 < \left\|x_n - \mu_l\right\|_2$. By definition, $(p_{nk})_{k\in[K]}$ minimizes the cost $\sum_{k=1}^K r(p_{nk}) w_n \left\|x_n - \mu_k\right\|_2^2$. Besides that, we know that $w_n > 0$. Hence, $r(p_{nk})\left\|x_n - \mu_k\right\|_2^2 + r(p_{nl})\left\|x_n - \mu_l\right\|_2^2 \leq r(p_{nl})\left\|x_n - \mu_k\right\|_2^2 + r(p_{nk})\left\|x_n - \mu_l\right\|_2^2$. This means that $r(p_{nk})\left(\left\|x_n - \mu_k\right\|_2^2 - \left\|x_n - \mu_l\right\|_2^2\right) \leq r(p_{nl})\left(\left\|x_n - \mu_k\right\|_2^2 - \left\|x_n - \mu_l\right\|_2^2\right)$. By assumption, we have $(\left\|x_n - \mu_k\right\|_2^2 - \left\|x_n - \mu_l\right\|_2^2) < 0$. Therefore, we know that $r(p_{nk}) \geq r(p_{nl})$. Since r is strictly increasing, we can conclude that $p_{nk} \geq p_{nl}$. This yields the claim. □

Hence, in an r-fuzzy K-means clustering, points are assigned more to those means that are closer than to those means that are farther away.

Cost of a Hard Clustering. By demanding that $r(0) = 0$ and $r(1) = 1$, we ensure that hard assignments are preserved *exactly* by the fuzzifier function.

Observation 5.10 (cost of a hard clustering)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ and let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an arbitrary but fixed fuzzifier function. Consider some means $C \subseteq \mathbb{R}^D$. Let Z be the K-means hard clustering of X induced by C (cf. Observation 4.4). Then, $\phi_X^{(r)}(C, Z) = \mathrm{km}_X(C)$.*

Monotonicity of Solutions. Simply speaking, the property that $r(0) = 0$ implies that we do not increase the cost when we add some more mean vectors to our solution.

Lemma 5.11 (monotonicity)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, and $C \subseteq \mathbb{R}^D$ a vector of mean vectors. Then,*

$$\forall \tilde{C} \subseteq C : \ \phi_X^{(r)}(\tilde{C}) \geq \phi_X^{(r)}(C) \ .$$

Proof. Let $\tilde{P} = (\tilde{p}_{nk})_{n,k}$ be the r-fuzzy $|\tilde{C}|$-clustering of X induced by $\tilde{C}$. Let P be the $|C|$-clustering of X that assigns (x_n, w_n) to $\mu_k \in C \setminus \tilde{C}$ with probability 0 and that assigns (x_n, w_n) to $\mu_k \in \tilde{C}$ with probability $\tilde{p}_{nk}$. Since r is a fuzzifier function, we have $r(0) = 0$. Hence, by construction, $\phi_X^{(r)}(\tilde{C}) = \phi_X^{(r)}(\tilde{C}, \tilde{P}) = \phi_X^{(r)}(C, P) \geq \phi_X^{(r)}(C)$. This yields the claim. □

Again, observe that the K-means cost function exhibits the same property. That is, by adding a mean to a set of means we do not increase the (fuzzy) K-means cost of an induced solution.

No Probabilities. Be aware that Definition 5.8 does *not* ensure that a fuzzified soft assignment $r(p_{nk})$ can still be thought of as a probability. There might be some $p \in [0,1]$ with $r(p) > p$. Hence, there might be a soft assignment $(p_k)_{k\in[K]} \in \Delta_{K-1}$ with $\sum_{k=1}^{K} r(p_k) > 1$.

5.2.3 Bounded Contribution

In a soft clustering, each point belongs to each cluster with some probability p_{nk}. In the r-fuzzy K-means cost function, these probabilities p_{nk} are fed to the fuzzifier function r. The resulting fuzzified probabilities $r(p_{nk})$ might be arbitrarily small. Even the sum $\sum_{k=1}^{K} r(p_{nk})$ can be arbitrarily close to zero. This means that the respective data point (x_n, w_n) might contribute an arbitrarily small share to the overall r-fuzzy K-means cost. In the remainder of this thesis, we sometimes require that each data point contributes a certain minimum share to the overall cost.

Definition 5.12. *For each fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $K \in \mathbb{N}$, we let*

$$\mathbf{c}_r^*(K) := \min \left\{ \sum_{k=1}^{K} r(p_k) \;\middle|\; (p_k)_{k\in[K]} \in \Delta_{K-1} \right\}$$

be the minimum contribution in an r-fuzzy K-clustering.

Note that this minimum always exists: For each fuzzifier function r, the function $g : \Delta_{K-1} \to \mathbb{R}_{\geq 0}$ with $g((p_k)_{k\in[K]}) = \sum_{k=1}^{K} r(p_k)$ is continuous and lower bounded by 0. Since the simplex $\Delta_{K-1} \subseteq \mathbb{R}^K$ is closed and bounded, $\mathbf{c}_r^*(K)$ exists (Stewart, 2009, p. 964).

We do not need to know this minimum contribution exactly, but we do need to know a lower bound on this value. This is why we consider the following property:

Definition 5.13. *Let $\mathbf{c}_r : \mathbb{N} \to (0,1]$. A fuzzifier function r is $\mathbf{c}_r$-contribution-bounded if*

$$\forall K \in \mathbb{N}: \quad 0 < \mathbf{c}_r(K) \leq \mathbf{c}_r^*(K) \quad \text{and} \quad \mathbf{c}_r(K) \geq \mathbf{c}_r(K+1) .$$

We stress the fact that whenever we mention $\mathbf{c}_r$-contribution-bounded functions, we implicitly assume that $\mathbf{c}_r$ is a function $\mathbf{c}_r : \mathbb{N} \to (0,1]$.

To check that this definition is reasonable, we show that every fuzzifier function r is $\mathbf{c}_r^*$-contribution-bounded.

Lemma 5.14. *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function. Then,*

$$\forall K \in \mathbb{N}: \quad 0 < \mathbf{c}_r^*(K) \leq 1 \quad \text{and} \quad \mathbf{c}_r^*(K) \geq \mathbf{c}_r^*(K+1) .$$

Proof. First, we prove that $\mathbf{c}_r^*(K) \in [0,1]$. Consider an arbitrary but fixed $(p_k)_{k\in[K]} \in \Delta_{K-1}$. Observe that, since for all $(p_k)_{k\in[K]} \in \Delta_{K-1}$ we have $\sum_{k=1}^{K} p_k = 1$, there exists a $k \in [K]$ with $p_k > 0$. Since a fuzzifier function is strictly increasing, we have $r(p_k) > 0$. Besides that, for all $l \neq k$, we have $r(p_l) \geq 0$. Hence, $\mathbf{c}_r^*(K) > 0$. Next, consider some $p \in \Delta_{K-1} \cap [0,1]^K$. Observe that $\sum_{k=1}^{K} r(p_k) = 1$ since a fuzzifier function satisfies $r(0) = 0$ and $r(1) = 1$. Hence, $\mathbf{c}_r^*(K) \leq 1$. This yields the first part of the claim.

To prove the second part of the claim, consider some $(p_k)_{k\in[K]} \in \Delta_{K-1}$. Let $(\tilde{p}_k)_{k\in[K+1]} \in \Delta_K$ with $\tilde{p}_k = p_k$ for all $k \in [K]$ and $p_{K+1} = 0$. Since r is a fuzzifier function, we have $r(p_{K+1}) = r(0) = 0$. Therefore, $\sum_{k=1}^{K} r(p_k) = \sum_{k=1}^{K+1} r(\tilde{p}_k)$. Hence, $\left\{\sum_{k=1}^{K} r(p_k) \;\middle|\; (p_k)_{k\in[K-1]} \in \Delta_{K-1}\right\} \subseteq \left\{\sum_{k=1}^{K+1} r(p_k) \;\middle|\; (p_k)_{k\in[K]} \in \Delta_K\right\}$ This yields the claim. □

We do not need to know the function $\mathbf{c}_r^*$ exactly. As said before, we will need to be able to compute a lower bound $\mathbf{c}_r(K)$ in advance. This goes without saying that $\mathbf{c}_r(K)$ should be as large as possible.

5.2.4 Bounded Increase

Assume that we know that two soft assignments p_{nk} and p_{nl} are very similar. More precisely, assume that there is a small $\epsilon > 0$ such that $p_{nl} \leq (1+\epsilon)p_{nk}$. The cost function $\phi_X^{(r)}$ depends on the fuzzified probabilities $r(p_{nk})$. So we can make use of the fact that $p_{nl} \leq (1+\epsilon)p_{nk}$ only if we can deduce something about the relation between the fuzzified probabilities $r(p_{nl})$ and $r(p_{nk})$. This leads us to the following property:

Definition 5.15. *Let $\mathbf{i}_r \in [1,\infty)$. A function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is $\mathbf{i}_r$-increase-bounded if*

$$\forall p \in [0,1]\ \forall \epsilon \in [0,1] :\ r\left(\left(1+\frac{\epsilon}{\mathbf{i}_r}\right)\cdot p\right) \leq (1+\epsilon)\cdot r(p)\ .$$

Hence, if r is an $\mathbf{i}_r$-increase-bounded fuzzifier function, then for all $p,q \in [0,1]$ with $q \leq (1+\epsilon/\mathbf{i}_r)p$ we have $r(q) \leq (1+\epsilon)r(p)$. We stress the fact that whenever we talk about $\mathbf{i}_r$-increase-bounded functions, we implicitly assume that $\mathbf{i}_r$ is some value with $\mathbf{i}_r \in [1,\infty)$.

Let us briefly discuss this property in the context of continuity and derivatives. First, consider a continuous function r. By definition, for each $p \in [0,1]$, there exists some $\delta_p > 0$ such that for all $p' \in [1 \pm \delta_p]$ we have $r(p') \in [1 \pm \epsilon]\cdot r(p)$. In particular, $r(p+\delta_p) \leq (1+\epsilon)\cdot r(p)$. Definition 5.15 demands that $\delta_p := \frac{\epsilon}{\mathbf{i}_r} p$ fulfills the latter inequality. Second, assume that r is differentiable and that its first derivative r' is strictly increasing. Then we know that $r\left(\left(1+\frac{\epsilon}{\mathbf{i}_r}\right)\cdot p\right) \leq r(p) + \epsilon \cdot \frac{p}{\mathbf{i}_r}\cdot r'\left(\left(1+\frac{\epsilon}{\mathbf{i}_r}\right)\cdot p\right)$. So if $r'\left(\left(1+\frac{\epsilon}{\mathbf{i}_r}\right)\cdot p\right) < \frac{\mathbf{i}_r}{p} r(p)$, then we know that r is increase-bounded. To sum up, if r does not increase too fast, then it is increase-bounded.

5.2.5 Reducing Probabilities

Simply speaking, the following property ensures that fuzzified probabilities can still be thought of as probabilities.

Definition 5.16. *A function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is [0,1]-reducing if*

$$\forall p \in [0,1] :\ r(p) \leq p\ .$$

Observe that, if $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a [0,1]-reducing function and $P = (p_{nk})_{n,k} \in \Delta_{N,K-1}$, then $r(P) = (r(p_{nk}))_{n,k}$ is a probabilistic membership matrix (Definition 3.1). This means that we can apply our soft-to-hard clustering technique from Theorem 3.21 to r-fuzzy clusters if r is [0,1]-reducing. We will pursue this idea in Chapter 8.

5.2.6 Induced r-Fuzzy Clusterings

Unfortunately, it is not enough to know that an induced r-fuzzy clustering exists. We have to be able to compute an induced r-fuzzy clustering in reasonable time. In this section, we aim to specify the form that an induced r-fuzzy clustering takes, to the extent possible, for certain classes of fuzzifier functions.

Independencies

First and foremost, take note of the following useful observations.

Lemma 5.17 ($|X|$ independent observations)**.** *Let $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, and let $P = (p_{nk})_{n\in[N],k\in[K]} \in \Delta_{N,K-1}$.*

P is an r-fuzzy clustering of X induced by C if and only if, for each $n \in [N]$ with $w_n > 0$, $(p_{nk})_{k\in[K]}$ is a vector that minimizes

$$\sum_{k=1}^{K} r(p_{nk}) \left\|x_n - \mu_k\right\|_2^2$$

subject to $(p_{nk})_{k\in[K]} \in \Delta_{K-1}$.

Proof. Consider an arbitrary but fixed r-fuzzy clustering $P = (p_{nk})_{n,k}$ of X induced by C. That is, $(p_{nk})_{n,k}$ minimizes $\phi_X^{(r)}(C,(p_{nk})_{n,k}) = \sum_{n=1}^{N} w_n \cdot \left(\sum_{k=1}^{K} r(p_{nk}) \left\|x_n - \mu_k\right\|_2^2\right)$ subject to: $(p_{nk})_{k\in[K]} \in \Delta_{K-1}$ for each $n \in [N]$. Hence, for each $n \in [N]$ with $w_n > 0$, $(p_{nk})_{k\in[K]}$ is a vector that minimizes $\sum_{k=1}^{K} r(p_{nk}) \left\|x_n - \mu_k\right\|_2^2$ subject to $(p_{nk})_{k\in[K]} \in \Delta_{K-1}$. For each $n \in [N]$ with $w_n = 0$, we have $w_n \cdot \left(\sum_{k=1}^{K} r(\tilde{p}_{nk}) \left\|x_n - \mu_k\right\|_2^2\right) = 0$ for all $(\tilde{p}_{nk})_{k\in[K]} \in \Delta_{K-1}$. □

Corollary 5.18 (independence of positive weights)**.** *Let $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $x \in \mathbb{R}^D$ and $v, w \in \mathbb{R}_+$ with $v \neq w$. An r-fuzzy clustering of (x, v) induced by C is also an r-fuzzy clustering of a data point (x, w) induced by C.*

A Vital Assumption

The algorithms presented in the following sections heavily rely on the assumption that, given K means, an induced r-fuzzy K-clustering can be computed.

Assumption 5.19 (computation of an induced r-fuzzy clustering)**.** *There is an algorithm that, given a point $x \in \mathbb{R}^D$, means $C \subseteq \mathbb{R}^D$ and a fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, computes an r-fuzzy K-clustering of a data point (x, w) with weight $w \in \mathbb{R}_{\geq 0}$ induced by C in time*

$$D \cdot \mathbf{t}_r(|C|) \,,$$

for some $\mathbf{t}_r(|C|) \in \Omega(|C|)$ that only *depends on $|C|$ and r.*

In the following, we will not explicitly point out whether an algorithm depends on this assumption. However, if it does, then the factor $\mathbf{t}_r(K)$ will appear in the corresponding runtime bound.

Probability Zero

In contrast to a classical fuzzy K-means clustering, an induced r-fuzzy clustering might possibly assign a data point to some of the clusters with probability zero. For example, the fuzzifier function s_β, which we presented in Section 4.3.2, exhibits this property. Nonetheless, assignments with probability zero behave as to be expected:

Corollary 5.20 (monotonicity)**.** *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, and a data point $(x_n, w_n) \in \mathbb{R}^D \times \mathbb{R}_+$. Let $(p_{nk})_{k\in[K]} \in \Delta_{K-1}$ be an r-fuzzy clustering of (x_n, w_n) induced by C. If $p_{nk} = 0$ for some $k \in [K]$, then for all $l \in [K]$ with $\left\|x_n - \mu_l\right\|_2 > \left\|x_n - \mu_k\right\|_2$, we have $p_{nl} = 0$.*

Proof. Apply Lemma 5.9. □

First-Order Optimality Conditions

We can describe an induced r-fuzzy clustering via the first-order optimality condition of the cost function. This approach has also been pursued by Klawonn and Höppner (2003) and Klawonn (2004). In contrast to them, we formalize all results, consider arbitrary fuzzifier functions, and provide complete proofs.

As explained in the previous section, it is possible that an r-fuzzy clustering $P = (p_{nk})_{n,k}$ induced by C assigns a data point to some of the clusters with probability zero. In other words, there might be a soft assignment $(p_{nk})_{k\in[K]}$ of some point x_n that does not lie in the interior (but on the border) of the simplex Δ_{K-1}. In this case, P does not necessarily satisfy the first-order optimality conditions of the function $\phi_X^{(r)}(C, \cdot)$. For this reason, the following lemma takes a slightly complicated form.

For each data point (x_n, w_n), consider a set $I_n \subset [K]$ that determines which assignments are *not* fixed to probability zero. That is, for all $k \notin I_n$, we set $p_{nk} := 0$. Then we use the first-order optimality conditions of the cost function to determine the values $p_{nk} \in \mathbb{R}$ with $k \in I$ and $\sum_{k \in I} p_{nk} = 1$ that minimize the cost function. If I_n indicates exactly the set of non-zero assignments of an induced r-fuzzy clustering of (x_n, w_n), then these values p_{nk} are indeed the probabilities of an induced r-fuzzy clustering.

Before we explain the relevancy of this approach, take a look at the formal result:

Lemma 5.21. *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a differentiable function whose first derivative r' is invertible over the interval $[0,1]$ and satisfies $r'((0,1]) > 0$. Let $x \in \mathbb{R}^D$ and $C = (\mu_k)_{k \in [K]} \subseteq \mathbb{R}^D$ where $x \neq \mu_k$ for all $k \in [K]$.*

Fix some set $I \subset [K]$. Consider the function g_I given by

$$g_I : \mathbb{R}^{|I|} \to \mathbb{R}$$
$$(\tilde{p}_k)_{k \in I} \mapsto \sum_{k \in I} r(\tilde{p}_k) \left\| x - \mu_k \right\|_2^2 - \lambda \left(\sum_{k \in I} \tilde{p}_k - 1 \right),$$

where the variable λ denotes a Lagrange multiplier.

The function has a unique extremum $(p_k)_{k \in I} \in \mathbb{R}^{|I|}$ with the following properties:

1. *For all $k \in I$, p_k is the solution to*

$$r'(p_k) \left\| x - \mu_k \right\|_2^2 - \lambda = 0 \tag{5.3}$$

where $\lambda \in \mathbb{R}$ is a constant that satisfies

$$1 = \sum_{k \in I} \left(r' \right)^{-1} \left(\frac{\lambda}{\left\| x - \mu_k \right\|_2^2} \right). \tag{5.4}$$

2. *If $p_k > 0$ for all $k \in I$, then $(p_k)_{k \in I}$ is the global minimum of g_I.*

3. *For all $k, l \in I$, we have*

$$r(p_k) \left\| x - \mu_k \right\|_2^2 = r(p_l) \left\| x - \mu_l \right\|_2^2.$$

Proof. First, we prove Item 1. Set the first derivative in the direction of $\tilde{p}_k$ equal to zero:

$$\frac{\partial}{\partial \tilde{p}_k} g_I \left((\tilde{p}_k)_{,k \in I} \right) = r'(\tilde{p}_k) \left\| x - \mu_k \right\|_2^2 - \lambda = 0. \tag{5.5}$$

Since r' is invertible over the interval $[0,1]$, we can solve this equation for $\tilde{p}_k$. This yields (5.3). Observe that $\sum_{k \in I} p_k = 1$. This yields (5.4).

Next, we prove Item 2. By assumption, $x \neq \mu_k$. Moreover, in Item 2, we assume that $p_k > 0$ for all $k \in I$. Since $r'((0,1]) > 0$ by assumption, we can conclude that $r'(p_k) > 0$ for all $k \in I$. Hence,

$$\frac{\partial}{\partial \tilde{p}_k \partial \tilde{p}_{nl}} g_I((p_k)_{k \in I}) = \begin{cases} r'(p_k) \left\| x - \mu_k \right\|_2^2 > 0 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

for all $k, l \in I$. Hence, the Hessian matrix of g is positive definite (Magnus and Neudecker, 1999, pp. 15). Due to (Magnus and Neudecker, 1999, pp. 123,pp. 131), this yields the correctness of Item 2.

Finally, we prove Item 3. Setting (5.5) to zero gives $\lambda = r(p_k) \left\| x - \mu_k \right\|_2^2$ where λ is independent of k. Combining these equalities for all $k \in I$ with respect to the same $n \in [N]$ yields the claim. □

Lemma 5.21 gives us a hint on how useful fuzzifier functions should be defined. Klawonn and Höppner (2003) and Klawonn (2004) use these results, especially the third claim of the lemma, to derive alternative fuzzifier functions. We will review these alternative functions in Section 5.3.

Efficient Use of the First-Order Optimality Conditions

So far, Lemma 5.21 does not tell us anything about "true" non-zero assignments. Obviously, it is extremely time consuming to determine all the soft clusterings specified by Lemma 5.21 with respect to each possible assumption on the non-zero assignments (per point!) and to find the best soft clustering among all of them. To be able to find the induced r-fuzzy clustering efficiently, we have to characterize the "true" non-zero assignments further. This is what the following lemma does.

Lemma 5.22. *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a differentiable function whose first derivative r' is invertible over the interval $[0,1]$ and satisfies $r'((0,1]) > 0$. Consider some $(x,w) \in \mathbb{R}^D \times \mathbb{R}_+$ and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ where $x \neq \mu_k$ for all $k \in [K]$. Let $(p_k)_{k\in[K]}$ be an r-fuzzy clustering of (x,w) induced by C. Let*

$$I := \{k \in [K] \mid p_k > 0\} \ .$$

Let π be a permutation on $[K]$ such that

$$\|x - \mu_{\pi(1)}\|_2 \leq \|x - \mu_{\pi(2)}\|_2 \leq \cdots \leq \|x - \mu_{\pi(K)}\|_2 \ .$$

Then, the following three properties hold true:

1. *There exists some $l \in [K]$ such that $I = \{\pi(q) \in [K] \mid q \leq l\}$.*
2. *For each $k \in I$, p_k is a solution to the equation given in Item 1 from Lemma 5.21 (with respect to I).*
3. *Consider some $I' \subset I$. Let $\hat{P} = (\hat{p}_k)_{k\in I'}$ be the vector where each $\hat{p}_k$ is the solution to the corresponding equation given in Item 1 from Lemma 5.21 with respect to I' (instead of I). Then, $\phi^{(r)}_{(x,w)}(C,\hat{P}) \geq \phi^{(r)}_{(x,w)}(C,P)$.*

Proof. The first part of the claim is a consequence of Corollary 5.20. The second part of the claim is a consequence of Lemma 5.21. To see that the third part of the claim holds true, consider functions g_I and $g_{I'}$ as defined in Lemma 5.21. The condition $I' \subset I$ can be interpreted as a reduction of the space of possible solutions: Each point p' from the input domain of $g_{I'}$ can be interpreted as point p from the input domain of g_I (with additional coordinates that are set to 0) with $g_{I'}(p') = g_I(p)$. Hence, a minimum of $g_{I'}$ cannot induce a better solution than a minimum of g_I. This observation yields the claim. □

This last lemma shows us a way of applying Lemma 5.21 in an efficient way: We do *not* have to consider all possible assumptions on the non-zero assignments. From Lemma 5.22, we see that we just have to sort the distances and evaluate the soft assignments given by Lemma 5.21 in the right way. We apply this approach in the next Section 5.3.

5.3 Special Cases

After having discussed the r-fuzzy K-means problem and fuzzifier functions in general, we now turn to examine the concrete fuzzifier functions that we described in Section 5.2. Table 5.1 gives an overview of the results that we explain in the remainder of this section.

Recall from Section 5.1.2 that induced r-fuzzy means are easy to compute, while the computation of an induced r-fuzzy clustering heavily depends on the chosen fuzzifier function. Hence, in the following, we focus on the latter.

r	parameter	description	reducing	$\mathbf{c}_r(K)$	$\mathbf{i}_r$	$\mathbf{t}_r(K)$
id	–	identity	✓	1	1	$\mathcal{O}(K)$
s_β	$\beta \in [0,1]$	quadratic-linear	✓	$\frac{1-\beta}{(1+\beta)\cdot K} + \frac{2\beta}{1+\beta}$	4	$\mathcal{O}(K \log(K))$
p_m	$m \in [1,\infty)$	m-th power	✓	$1/K^{m-1}$	$4m$	$\mathcal{O}(K)$
e_γ	$\gamma \in \mathbb{R}_+$	exponential	✓	$\frac{\gamma}{e^\gamma - 1}$	–	$\mathcal{O}(K \log(K))$

Table 5.1: Overview of the properties of some fuzzifier functions.

5.3.1 Identity – K-Means

The id-fuzzy K-means problem with the identity function id corresponds to the classical K-means problem, as we already noted in Section 4.2.1:

Observation 5.23. *Consider* $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ *and* $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$. *Then, each hard clustering* $Z = (z_{nk})_{n\in[N],k\in[K]}$ *that satisfies*

$$\forall n \in [N]\ \forall k \in [K]\colon\ z_{nk} = 1 \ \Rightarrow\ k \in \arg\min\left\{\left\|x_n - \mu_l\right\|_2 \,\middle|\, l \in [K]\right\}$$

minimizes $\phi_X^{(\mathrm{id})}(C,\cdot)$ *with respect to all soft* K*-clusterings of* X.

Hence, an induced r-fuzzy K-clustering of a data set X can be computed in time $\mathcal{O}(|X|\cdot K\cdot D)$. That is, $\mathbf{t}_r(K) = \Theta(K)$ (Assumption 5.19). Moreover, the identity function id is clearly a fuzzifier function that is $[0,1]$-reducing, 1-increase-bounded, and $\mathbf{c}^*_{\mathrm{id}}$-contribution-bounded with $\mathbf{c}^*_{\mathrm{id}}(K) = 1$ for all $K \in \mathbb{N}$.

5.3.2 Power Function – Classical Fuzzy K-Means

The classical fuzzy K-means problem with fuzzifier $m \in (1,\infty)$ coincides with the p_m-fuzzy K-means problem where p_m is defined as follows.

Definition 5.24. *For each* $m \in [1,\infty)$, *we let* $\mathrm{p}_m : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *be the function with*

$$\mathrm{p}_m(p) = p^m\ .$$

Lemma 5.25 (properties of p_m)**.** *For all* $m \in [1,\infty)$, p_m *satisfies the following properties:*

- p_m *is a* $[0,1]$*-reducing fuzzifier function.*
- p_m *is* $(4m)$*-increase-bounded.*
- *If* $m \in \mathbb{N}$, *then* p_m *is* $(2m)$*-increase-bounded.*
- p_m *is* $\mathbf{c}_{\mathrm{p}_m}$*-contribution-bounded with* $\mathbf{c}^*_{\mathrm{p}_m}(K) = 1/K^{m-1}$ *for all* $K \in \mathbb{N}$.

Proof. Clearly, since $m \geq 1$, p_m is a $[0,1]$-reducing fuzzifier function. Consider arbitrary but fixed $p, \epsilon \in [0,1]$. For all $c \in \mathbb{N}$, we have

$$\mathrm{p}_m\left(\left(1 + \frac{\epsilon}{c\cdot m}\right)p\right) = \left(1 + \frac{\epsilon}{c\cdot m}\right)^m \cdot p^m = \left(1 + \frac{\epsilon}{c\cdot m}\right)^m \cdot \mathrm{p}_m(p)\ .$$

For $m \in \mathbb{N}$ and $c = 2$, we have $\left(1 + \frac{\epsilon}{2m}\right)^m \leq (1+\epsilon)$ due to Lemma A.1. This yields the third claim. For $m \in [1,\infty)$ and $c = 4$, we can bound

$$\left(1 + \frac{\epsilon}{4m}\right)^m \leq \left(1 + \frac{\epsilon}{4\lfloor m\rfloor}\right)^{\lfloor m\rfloor + 1} \leq \left(1 + \frac{\epsilon}{4\lfloor m\rfloor}\right)^{2\lfloor m\rfloor} \leq 1 + \epsilon\ ,$$

where the last inequality is again due to Lemma A.1. This yields the second claim.

Now consider an arbitrary $(p_k)_{k\in[K]} \in \Delta_{K-1}$. We can apply Hölder's inequality (Hardy et al., 1952, p.21) and bound

$$\left(\sum_{k=1}^{K} p_k^m\right)^{1/m} \cdot K^{(m-1)/m} = \left(\sum_{k=1}^{K} p_k^m\right)^{1/m} \cdot \left(\sum_{k=1}^{K} 1^{m/(m-1)}\right)^{(m-1)/m} \geq \sum_{k=1}^{K} p_k = 1 \,.$$

Hence, $\sum_{k=1}^{K} p_k^m \geq K^{m-1}$. In particular, if $p_k = 1/K$ for all $k \in [K]$, then $\sum_{k=1}^{K} \mathrm{p}_k^m = 1/K^{m-1}$. Hence, $\mathbf{c}^*_{\mathrm{p}_m}(K) = 1/K^{m-1}$. This yields the last claim. □

As already explained in Section 4.1, a p_m-induced fuzzy K-means clustering can be determined efficiently. For an illustration of an induced p_m-fuzzy clustering (for different values of m) we refer back to Figure 4.5 in Section 4.3.2.

Lemma 5.26 (induced p_m-fuzzy clustering)**.** *Let $X = ((x_n, w_n))_{n\in[N]}$, $C = (\mu_k)_{k\in[K]} \subset \mathbb{R}^D$, and $m \in (1,\infty)$. The p_m-fuzzy clustering $(p_{nk})_{n,k}$ of X induced by C can be computed in time $\mathcal{O}(|X| \cdot K \cdot D)$ (i.e., $\mathbf{t}_r(K) = \Theta(K)$ with Assumption 5.19). In particular, for all $n \in [N]$ with $\forall l \in [K] : x_n \neq \mu_l$, we have*

$$\forall k \in [K] : \; p_{nk} = \frac{\left\|x_n - \mu_k\right\|_2^{-\frac{2}{m-1}}}{\sum_{l=1}^{K} \left\|x_n - \mu_l\right\|_2^{-\frac{2}{m-1}}} > 0 \,.$$

Proof. Use the method of Lagrange multipliers to ensure that $\sum_{k=1}^{K} p_{nk} = 1$ for all $n \in [N]$. An analysis of the first-order optimality conditions of the resulting objective function yields the claim. □

5.3.3 Quadratic-Linear – Between K-Means and Fuzzy K-Means

Klawonn and Höppner (2003) propose a mixture of two fuzzifier functions: the identity function id, which leads to hard K-means clustering, and the square function p_2, which leads to the classical fuzzy K-means clustering with fuzzifier $m = 2$.

Definition 5.27. *For each $\beta \in [0,1]$, we let $\mathrm{s}_\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the function with*

$$\mathrm{s}_\beta(p) = \frac{1-\beta}{1+\beta} \cdot p^2 + \frac{2\beta}{1+\beta} \cdot p \,.$$

For $\beta = 0$, we obtain the power function $\mathrm{s}_0 = \mathrm{p}_2$. For $\beta = 1$, we obtain the identity function $\mathrm{s}_1 = \mathrm{id} = \mathrm{p}_1$.

Lemma 5.28 (properties of s_β)**.** *For all $\beta \in [0,1]$, s_β satisfies the following:*

- *s_β is a $[0,1]$-reducing fuzzifier function.*
- *s_β is 4-increase-bounded.*
- *s_β is $\mathbf{c}_{\mathrm{s}_\beta}$-contribution-bounded with $\mathbf{c}_{\mathrm{s}_\beta}(K) = \frac{1-\beta}{(1+\beta)\cdot K} + \frac{2\beta}{1+\beta}$ for all $K \in \mathbb{N}$.*

Proof. Observe that s_β is a convex combination of the two functions p_2 and p_1. With Lemma 5.26, we can conclude that s_β is a $[0,1]$-reducing, $\left(\frac{1-\beta}{1+\beta} \cdot \mathbf{c}^*_{\mathrm{p}_2} + \frac{2\beta}{1+\beta} \cdot \mathbf{c}^*_{\mathrm{p}_1}\right)$-contribution-bounded, and 4-increase-bounded fuzzifier function. With Lemma 5.25 the claim follows. □

Next, we show how an induced s_β-means clustering can be computed. Our results coincide with the results of Klawonn (2004) and Klawonn and Höppner (2003). However, neither of these publications contains a complete proof of correctness. An illustration of induced s_β-fuzzy clusterings for different values of β can be found in Figure 4.6.

Lemma 5.29. *Given $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $\beta \in [0,1]$, and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, Algorithm 2 computes an s_β-means clustering of X induced by C in time*

$$\mathcal{O}\left(|X| \cdot K(D + \log(K))\right) .$$

With our notation from Assumption 5.19, we can write $\mathbf{t}_r(K) = K\log(K)$.

Proof. The correctness follows from Lemma 5.21, Lemma 5.22, and the next Lemma 5.30. There is only one little trick: When checking whether $p_{nk} > 0$, we make use of the fact that the distances are sorted in ascending order and that the bound $(1/\beta) + k - 1$ increases if k increases. □

Algorithm 2 Induced s_β-Means Clustering

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $\beta \in [0,1]$, and $C = (\mu_k)_{k\in[K]} \in \mathbb{R}^D$

$P := (0)_{n,k}$
for all $n \in [N]$ **do** ▷ consider the n-th point
  $s := 0$
  for all $k \in [K]$ **do** ▷ compute and sort the distances
    $d_{nk} := \left\|x_n - \mu_k\right\|_2^2$
  Compute a permutation π on $[K]$ such that $d_{n\pi(1)} \leq d_{n\pi(2)} \leq \cdots \leq d_{n\pi(K)}$.
  if $d_{n\pi(1)} = 0$ or $w_n = 0$ **then** ▷ $x_n = \mu_k$ for some $k \in [K]$
    $p_{n\pi(1)} := 1$
  else
    $l := 0$ ▷ initialize search
    $s_0 := 0$
    for all $k = 1, \ldots, K$ **do**
      $s_k := s_k + 1/d_{n\pi(k)}$ ▷ $s_k = \sum_{h=1}^{k} \left(1/\left\|x_n - \mu_{\pi(h)}\right\|_2^2\right)$
      if $d_{n\pi(k)} \cdot s_k < (1/\beta) + k - 1$ **then** ▷ $p_{n\pi(k)} > 0$? (see (5.7))
        $l := k$
    for all $k = 1, \ldots, K$ **do**
      Set

$$p_{n\pi(k)} := \begin{cases} \frac{1}{1-\beta}\left((1 + (l-1)\cdot\beta)\cdot\left(d_{n\pi(k)}\cdot s_l\right)^{-1} - \beta\right) & \text{if } k \leq l \\ 0 & \text{otherwise} \end{cases}$$

return P

Lemma 5.30. *Fix some $\beta \in [0,1]$. Let $x \in \mathbb{R}^D$ and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ where $x \neq \mu_k$ for all $k \in [K]$. Fix some set $I \subset [K]$. Consider the function*

$$g_I : \mathbb{R}^{|I|} \to \mathbb{R} \ , \ (\tilde{p}_k)_{k\in I} \mapsto \sum_{k\in I} \mathrm{s}_\beta(\tilde{p}_k)\left\|x - \mu_k\right\|_2^2 - \lambda\left(\sum_{k\in I}\tilde{p}_k - 1\right) ,$$

where the variable λ denotes a Lagrange multiplier.

Then, g_I has a unique extremum $(p_k)_{k\in[K]}$ where

$$p_k = \frac{1}{1-\beta}\left(\left(1 + \left(|I| - 1\right)\cdot\beta\right)\cdot\left(\sum_{l\in I}\frac{\left\|x - \mu_k\right\|_2^2}{\left\|x - \mu_l\right\|_2^2}\right)^{-1} - \beta\right) \tag{5.6}$$

for all $k \in I$.

In particular, for all $k \in I$, we have $p_k > 0$ if and only if

$$\sum_{l \in I} \frac{\|x - \mu_k\|_2^2}{\|x - \mu_l\|_2^2} < \frac{1}{\beta} + |I| - 1 \,. \tag{5.7}$$

Proof. Consider s_β as a function $\mathrm{s}_\beta : \mathbb{R} \to \mathbb{R}$. s_β is continuous and its first derivative

$$\mathrm{s}'_\beta(p) = 2\frac{1-\beta}{1+\beta}p + \frac{2\beta}{1+\beta}$$

is strictly increasing. Hence, we can apply Lemma 5.21. Observe that

$$\begin{aligned} 0 = \mathrm{s}'_\beta(p_k)\|x - \mu_k\|_2^2 - \lambda \Leftrightarrow 0 &= \left(2\frac{1-\beta}{1+\beta}p_k + \frac{2\beta}{1+\beta}\right)\|x - \mu_k\|_2^2 - \lambda \\ \Leftrightarrow p_k &= \frac{1+\beta}{2(1-\beta)}\left(\frac{\lambda}{\|x - \mu_k\|_2^2} - \frac{2\beta}{1+\beta}\right) = \frac{1+\beta}{2(1-\beta)} \cdot \frac{\lambda}{\|x - \mu_k\|_2^2} - \frac{\beta}{1-\beta} \,. \end{aligned}$$

Hence, we have

$$\begin{aligned} 1 = \sum_{l \in I} p_k \Leftrightarrow 1 &= \sum_{l \in I}\left(\frac{1+\beta}{2(1-\beta)} \cdot \frac{\lambda}{\|x - \mu_l\|_2^2} - \frac{\beta}{1-\beta}\right) \\ \Leftrightarrow 1 &= \frac{1+\beta}{2(1-\beta)} \cdot \lambda \cdot \left(\sum_{l \in I} \frac{1}{\|x - \mu_l\|_2^2}\right) - |I| \cdot \frac{\beta}{1-\beta} \\ \Leftrightarrow \lambda &= \frac{2(1-\beta)}{1+\beta} \cdot \left(1 + \frac{|I|\beta}{1-\beta}\right) \cdot \left(\sum_{l \in I} \frac{1}{\|x - \mu_l\|_2^2}\right)^{-1} \,. \end{aligned}$$

Combining these equalities gives

$$p_k = \left(1 + \frac{|I|\beta}{1-\beta}\right) \cdot \left(\sum_{l \in I} \frac{\|x - \mu_k\|_2^2}{\|x - \mu_l\|_2^2}\right)^{-1} - \frac{\beta}{1-\beta} = \frac{1}{1-\beta}\left(\left(1 + (|I| - 1) \cdot \beta\right) \cdot \left(\sum_{l \in I} \frac{\|x - \mu_k\|_2^2}{\|x - \mu_l\|_2^2}\right)^{-1} - \beta\right) \,.$$

This yields the claim. □

5.3.4 Exponential Fuzzifier

Klawonn (2004) also proposed the following exponential fuzzifier function.

Definition 5.31. *For all $\gamma \in \mathbb{R}_+$, we let $\mathrm{e}_\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the function with*

$$\mathrm{e}_\gamma(p) = \frac{e^{\gamma p} - 1}{e^\gamma - 1} \,.$$

Klawonn (2004) already mentions some properties of this function, such as the fact that for $\gamma \to 0$ one obtains a hard clustering objective, but he provides no proofs. We start by formalizing and proving this observation:

Lemma 5.32 (similarity to K-means)**.** *For $\gamma \to 0^+$ with $\gamma \neq 0$, we have $\mathrm{e}_\gamma \to \mathrm{id}$ pointwise.*

Proof. On the one hand, in Lemma 5.33, we show that $\mathrm{e}_\gamma(p)$ is $[0,1]$-reducing. Hence, $\mathrm{e}_\gamma(p) \leq p$ for all $p \in [0,1]$. On the other hand, we can upper bound e_γ as follows. Recall the fact that the exponential function can be expressed as an infinite power series (Stewart, 2009, p. 772). Thus, $\mathrm{e}_\gamma(p) = \frac{e^{\gamma p}-1}{e^\gamma - 1} \geq \frac{(1+\gamma p)-1}{e^\gamma - 1} = \frac{\gamma}{e^\gamma - 1} \cdot p$. It is well known that $\lim_{\gamma \to 0, \gamma \neq 0} \frac{\gamma}{e^\gamma - 1} = 1$ (Stewart, 2009, p. 397). The squeeze theorem (Stewart, 2009, p. A42) yields the claim. □

Next, we check the properties of the exponential fuzzifier function. It is not surprising that this function turns out to be *not* increase-bounded.

Lemma 5.33 (properties of e_γ)**.** *For all $\gamma \in \mathbb{R}_+$, e_γ satisfies the following:*

- e_γ *is a* $[0,1]$*-reducing fuzzifier function.*
- e_γ *is* $\mathbf{c}_{\mathsf{e}_\gamma}$*-contribution-bounded with* $\mathbf{c}_{\mathsf{e}_\gamma}(K) = \frac{\gamma}{e^\gamma - 1}$ *for all* $K \in \mathbb{N}$*.*
- e_γ *is* not *increase-bounded, i.e., there* does not *exist a constant* $\mathbf{i}_{\mathsf{e}_\gamma} \in [1,\infty)$ *such that* e_γ *is* $\mathbf{i}_{\mathsf{e}_\gamma}$*-increase-bounded.*

Proof. It is easy to check that e_γ is a fuzzifier function. We omit the details. Next, we show that e_γ is $[0,1]$-reducing. Observe that e_γ is strictly convex since $\mathsf{e}_\gamma''(p) = \gamma^2 e^{\gamma p}/(e^\gamma - 1) > 0$ for all $p \in [0,1]$. Together with the fact that $\mathsf{e}_\gamma(0) = 0$ and $\mathsf{e}_\gamma(1) = 1$, this yields the second claim.

To prove the second claim, consider an arbitrary but fixed $(p_k)_{k\in[K]} \in \Delta_{K-1}$. We can write $\sum_{k=1}^K \mathsf{e}_\gamma(p_k) = \frac{1}{e^\gamma - 1}\left(-K + \sum_{k=1}^K e^{\gamma p_k}\right)$. Using the fact that the exponential function can be expressed as an infinite power series (Stewart, 2009, p. 772), we can bound $\sum_{k=1}^K e^{\gamma p_k} \geq \sum_{k=1}^K \left(1 + \gamma p_k\right) = K + \gamma$. This yields the second claim.

Finally, we check that e_γ is not increase-bounded. Consider arbitrary $p, \epsilon, \epsilon' \in (0,1)$ and $\gamma \in \mathbb{R}_+$. Towards a contradiction, assume that $\mathsf{e}_\gamma\left((1+\epsilon')p\right) \leq (1+\epsilon)\cdot \mathsf{e}_\gamma(p)$. Then,

$$
\begin{aligned}
&\mathsf{e}_\gamma\left((1+\epsilon')p\right) \leq (1+\epsilon)\cdot \mathsf{e}_\gamma(p) \\
\Leftrightarrow\ &\frac{e^{\gamma(1+\epsilon')p} - 1}{e^\gamma - 1} \leq (1+\epsilon)\frac{e^{\gamma p} - 1}{e^\gamma - 1} \\
\Leftrightarrow\ &e^{\gamma(1+\epsilon')p} - 1 \leq (1+\epsilon)\left(e^{\gamma p} - 1\right) \\
\Leftrightarrow\ &e^{\gamma(1+\epsilon')p} - 1 \leq e^{\gamma p} - 1 + \epsilon\left(e^{\gamma p} - 1\right) \\
\Leftrightarrow\ &e^{\gamma(1+\epsilon')p} - e^{\gamma p} \leq \epsilon\left(e^{\gamma p} - 1\right) \\
\Leftrightarrow\ &e^{(1+\epsilon')} - 1 \leq \epsilon\left(1 - e^{-\gamma p}\right) \qquad \text{(where } e^{-\gamma p} \in (0,1)\text{)} \\
\Rightarrow\ &1.7 < e^{1+0} - 1 \leq e^{(1+\epsilon')} - 1 \leq \epsilon\left(1 - e^{-\gamma p}\right) \leq \epsilon,
\end{aligned}
$$

which contradicts the fact that $\epsilon \in (0,1)$. This yields the claim. □

Nonetheless, we can compute an induced e_γ-fuzzy clustering in reasonable time, using an algorithm that proceeds similarly to Algorithm 2. Figure 5.1 illustrates an induced e_γ-fuzzy clustering and the resulting cost function.

Lemma 5.34. *There is an algorithm that, given $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right), \gamma \in \mathbb{R}_+$, and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, computes an e_γ-fuzzy clustering of X induced by C in time*

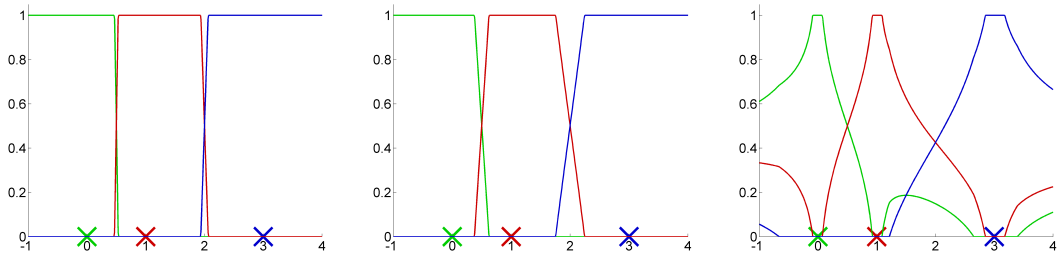
$$\mathcal{O}\left(NK(D + \log(K))\right) .$$

With our notation from Assumption 5.19, we can write $\mathbf{t}_r(K) = K\log(K)$.

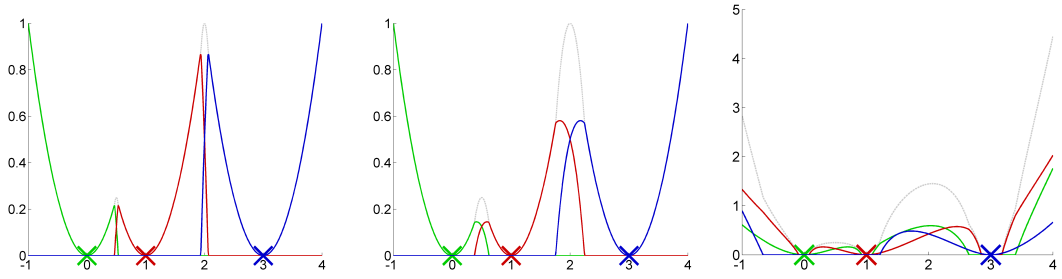
Proof. The algorithm proceeds similarly to Algorithm 2: Instead of evaluating p_{nk} according to Lemma 5.30, it uses the formula from Lemma 5.35. Instead of making use of (5.7), it makes use of (5.9). The correctness of this algorithm follows from Lemma 5.21, Lemma 5.22, and the next Lemma 5.35. □

Lemma 5.35. *Fix some $\gamma \in \mathbb{R}_+$. Let $x \in \mathbb{R}^D$ and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ where $x \neq \mu_k$ for all $k \in [K]$. Fix some set $I \subset [K]$. Consider the function*

$$g_I : \mathbb{R}^{|I|} \to \mathbb{R}\ ,\ (\tilde{p}_k)_{k\in I} \mapsto \sum_{k\in I} \mathsf{e}_\gamma(\tilde{p}_k)\left\|x - \mu_k\right\|_2^2 - \lambda\left(\sum_{k\in I} \tilde{p}_k - 1\right) ,$$



(a) soft assignment ($\gamma = 0.25$) (b) soft assignment ($\gamma = 1$) (c) soft assignment ($\gamma = 5$)



(d) cost ($\gamma = 0.25$) (e) cost ($\gamma = 1$) (f) cost ($\gamma = 5$)

Figure 5.1: Illustration of the impact of the fuzzifier function e_γ with $\gamma \in \{0.25, 1, 5\}$:
We are given the 3 mean points $\mu_1 = 0$, $\mu_2 = 1$, and $\mu_3 = 3$ in the one-dimensional space $\mathbb{R}$. For each data point $x_n \in [-1, 4]$ and each $k \in [3]$, we evaluate the optimal soft assignments p_{nk} for the given means and the resulting cost $\mathrm{e}_\gamma(p_{nk}) \left\| x_n - \mu_k \right\|_2^2$ per cluster.
Figure 5.1a through Figure 5.1c depict the optimal soft assignments, while Figure 4.5d through Figure 4.5f depict the resulting costs per cluster and the overall cost (gray).

where the variable λ denotes a Lagrange multiplier.

Then, g_I has a unique extremum $(p_k)_{k \in [K]}$ where

$$p_k = \frac{1}{|I|} \left(1 - \frac{1}{\gamma} \sum_{l \in I} \ln \left(\frac{\left\| x - \mu_k \right\|_2^2}{\left\| x - \mu_l \right\|_2^2} \right) \right) . \tag{5.8}$$

In particular, for all $k \in I$, we have $p_k > 0$ if and only if

$$\sum_{l \in I} \ln \left(\frac{\left\| x - \mu_k \right\|_2^2}{\left\| x - \mu_l \right\|_2^2} \right) < \gamma . \tag{5.9}$$

Proof. Consider e_γ as a function $\mathrm{e}_\gamma : \mathbb{R} \to \mathbb{R}$. Since e_γ is continuous and its first derivative

$$\mathrm{e}'_\gamma(p) = \gamma \frac{e^{\gamma p} - 1}{e^\gamma - 1}$$

is strictly increasing, we can apply Lemma 5.21.

Observe that

$$\begin{aligned}
& 0 = \mathrm{e}'_\gamma(p_k) \left\| x - \mu_k \right\|_2^2 - \lambda \\
\Leftrightarrow\ & 0 = \gamma \frac{e^{\gamma p_k} - 1}{e^\gamma - 1} \left\| x - \mu_k \right\|_2^2 - \lambda \\
\Leftrightarrow\ & e^{\gamma p_k} = \lambda \cdot \frac{e^\gamma - 1}{\gamma \cdot \left\| x - \mu_k \right\|_2^2} \\
\Leftrightarrow\ & p_k = \frac{1}{\gamma} \ln \left(\lambda \cdot \frac{e^\gamma - 1}{\gamma \cdot \left\| x - \mu_k \right\|_2^2} \right) = \frac{1}{\gamma} \left(\ln(\lambda) + \ln \left(\frac{e^\gamma - 1}{\gamma} \right) - \ln \left(\left\| x - \mu_k \right\|_2^2 \right) \right) .
\end{aligned}$$

Then, we have

$$\begin{aligned}
1 = \sum_{l \in I} p_k &\Leftrightarrow 1 = \sum_{l \in I} \frac{1}{\gamma} \left(\ln(\lambda) + \ln\left(\frac{e^\gamma - 1}{\gamma} \right) - \ln\left(\gamma \cdot \left\| x - \mu_l \right\|_2^2 \right) \right) \\
&\Leftrightarrow 1 = \frac{|I|}{\gamma} \left(\ln(\lambda) + \ln\left(\frac{e^\gamma - 1}{\gamma} \right) \right) - \frac{1}{\gamma} \sum_{l \in I} \ln\left(\left\| x - \mu_l \right\|_2^2 \right) \\
&\Leftrightarrow \frac{|I|}{\gamma} \ln(\lambda) = 1 + \frac{1}{\gamma} \sum_{l \in I} \ln\left(\left\| x - \mu_l \right\|_2^2 \right) - \frac{|I|}{\gamma} \ln\left(\frac{e^\gamma - 1}{\gamma} \right) \\
&\Leftrightarrow \ln(\lambda) = \frac{\gamma}{|I|} + \frac{1}{|I|} \sum_{l \in I} \ln\left(\left\| x - \mu_l \right\|_2^2 \right) - \ln\left(\frac{e^\gamma - 1}{\gamma} \right) .
\end{aligned}$$

Combining these equalities gives

$$\begin{aligned}
p_k &= \frac{1}{\gamma} \left(\frac{\gamma}{|I|} + \frac{1}{|I|} \left(\sum_{l \in I} \ln\left(\left\| x - \mu_l \right\|_2^2 \right) \right) - \ln\left(\left\| x - \mu_k \right\|_2^2 \right) \right) \\
&= \frac{1}{\gamma} \left(\frac{\gamma}{|I|} + \frac{1}{|I|} \sum_{l \in I} \left(\ln\left(\left\| x - \mu_l \right\|_2^2 \right) - \ln\left(\left\| x - \mu_k \right\|_2^2 \right) \right) \right) \\
&= \frac{1}{|I|} \left(1 + \frac{2}{\gamma} \sum_{l \in I} \ln\left(\frac{\left\| x - \mu_l \right\|_2}{\left\| x - \mu_k \right\|_2} \right) \right) \\
&= \frac{1}{|I|} \left(1 - \frac{1}{\gamma} \sum_{l \in I} \ln\left(\frac{\left\| x - \mu_k \right\|_2^2}{\left\| x - \mu_l \right\|_2^2} \right) \right) .
\end{aligned}$$

This yields the claim. □

"Like my old skleenball coach used to say, 'Find out what you don't do well, then don't do it.'"

Alf

Chapter 6

Two Key Properties

In this chapter, we present two basic techniques that help us to analyse the r-fuzzy K-means problem: First, we relate the r-fuzzy K-means cost function and the K-means cost function. Second, we transfer the notion of empty hard clusters, which can effectively be ignored in a hard clustering, to an r-fuzzy K-means clustering. Thereby, we effectively reduce the problem of approximating an optimal solution to the problem of approximating the best solution where each cluster has a certain non-negligible weight.

Publication. In this chapter, we generalize properties of the classical fuzzy K-means problem that we also identified in (Blömer et al., 2016, Lemma 1+2).

6.1 Relation to the K-Means Cost Function

There is a coarse yet useful relation between the objective functions of the K-means and an r-fuzzy K-means problem:

Lemma 6.1 (Relation to K-Means)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $K \in \mathbb{N}$, and let $C \subset \mathbb{R}^D$ with $|C| \geq 1$.*

Then,

$$\phi_X^{(r)}(C) \leq \mathrm{km}_X(C) .$$

If r is $\mathbf{c}_r$-contribution-bounded, then

$$\phi_X^{(r)}(C) \geq \mathbf{c}_r(|C|) \cdot \mathrm{km}_X(C) .$$

Proof. First, consider the lower bound. Write $X = ((x_n, w_n))_{n \in [N]}$. Let $(p_{nk})_{n,k}$ be the r-fuzzy clustering of X induced by $C = (\mu_k)_{k \in [K]}$. Then,

$$\phi_X^{(r)}(C) \geq \sum_{n=1}^{N} \left(\sum_{k=1}^{K} r(p_{nk}) \right) w_n \min\left\{ \left\| x_n - \mu_l \right\|_2 \,\middle|\, l \in [K] \right\} \geq \mathbf{c}_r(K) \cdot \mathrm{km}_X(C) .$$

Now consider the upper bound. Let $Z \in \{0,1\}^{N \times K} \cap \Delta_{N,K-1}$ be the K-means clustering (i.e., the id-fuzzy clustering) of X induced by C. Since r is a fuzzifier, we have $r(0) = 0$ and $r(1) = 1$. Hence,

$$\phi_X^{(r)}(C) \leq \phi_X^{(r)}(C, Z) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\| x_n - \mu_k \right\|_2^2 = \mathrm{km}_X(C) .$$

□

Observe that we cannot expect a lower bound that does not depend on an additional constraint, such as the contribution-bounded property of the fuzzifier function. Without additional constraints, the r-fuzzy K-means cost $\phi^{(r)}_{(x_n,w_n)}(C,P) = w_n \cdot \sum_{k=1}^{N} r(p_{nk}) \cdot \left\|x_n - \mu_k\right\|_2^2$ of a point (x_n,w_n) with respect to a fixed set of means $C = (\mu_k)_{k\in[K]}$ might be arbitrarily close to zero because the fuzzified probabilities $(r(p_{nk}))_{k\in[K]}$ might be arbitrarily small. We preclude this possibility by demanding that r is contribution-bounded.

6.2 Negligible Clusters

Given a soft K-clustering P of a data set X, we think of a cluster as negligible if not even a single data point (x_n,w_n) in X supports this cluster sufficiently.

Definition 6.2 (($\mathbf{i}_r,K,\epsilon$)-negligible cluster)**.** *Let $\mathbf{i}_r \in [1,\infty)$, $\epsilon \in (0,1]$, and $K \in \mathbb{N}$. Consider a soft L-clustering $P = (p_{nl})_{n\in[N],l\in[L]} \in \Delta_{N,L-1}$ with $L \geq K$.*

The l-th cluster given by P is ($\mathbf{i}_r,K,\epsilon$)-negligible *if*

$$\forall n \in [N]\colon\ p_{nl} \leq \frac{\epsilon}{2 \cdot \mathbf{i}_r \cdot K^2}\ .$$

For each solution that contains such a negligible cluster, there exists a solution with similar cost where none of the clusters is negligible.

Theorem 6.3 (remove negligible clusters)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $K \in \mathbb{N}$, and let $r\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an $\mathbf{i}_r$-increase-bounded fuzzifier function.*

For each $C \subset \mathbb{R}^D$ with $|C| = K$ there exists $\tilde{C} \subseteq C$ with

$$\phi^{(r)}_X(\tilde{C}) \ \leq\ (1+\epsilon) \cdot \phi^{(r)}_X(C)$$

and there exists an r-fuzzy L-clustering $\tilde{P} = (\tilde{p}_{nl})_{n\in[N],l\in[L]}$ of X induced *by $\tilde{C}$ that has no* ($\mathbf{i}_r,K,\epsilon$)*-negligible clusters.*

This observation resembles the notion of empty clusters in a K-means hard clustering: If a hard cluster is empty, then no point from the data set is assigned to this cluster (i.e., supports it), and we can remove its mean vector from the solution without increasing the K-means cost. Nonetheless, there is also an aspect that is different: We cannot preclude the possibility that an optimal r-fuzzy K-means solution contains a negligible cluster.

Theorem 6.3 is a direct consequence of the following constructive results.

Algorithm 3 Remove Negligible Clusters

Require: $X = ((x_n,w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $\epsilon \in (0,1]$, $r\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{i}_r \in [1,\infty)$, and $C = (\mu_k)_{k\in[K]} \in \mathbb{R}^D$

$\tilde{\epsilon} := \epsilon/(2K)$
$I := \emptyset$
$P_{-I} :=$ r-fuzzy K-clustering of X induced by C
$l := 0$
while $l < K - |I|$ **do**
  $l := l + 1$ ▷ Consider the next cluster
  if $\forall n \in [N]\colon\ p_{nl} < \tilde{\epsilon}/(\mathbf{i}_r K)$ **then** ▷ Is the support of the l-th cluster low?
    $I := I \cup \{l\}$ ▷ Remove mean μ_l from C_{-I}
    $C_{-I} := \left(\mu_k\right)_{k\in[K]\setminus I}$
    $P_{-I} :=$ r-fuzzy $(|K| - |I|)$-clustering of X induced by C_{-I}
    $l := 0$ ▷ Restart search
return (C_{-I}, P_{-I})

Lemma 6.4 (remove all negligible clusters)**.** *Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $K \in \mathbb{N}$, an $\mathbf{i}_r$-increase-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the value $\mathbf{i}_r \in [1, \infty)$, $\epsilon \in (0, 1]$, and means $C \in \mathbb{R}^D$ with $|C| = K$, Algorithm 3 computes a solution $(\tilde{C}, \tilde{P})$ such that $\tilde{C} \subseteq C$, $L := \left|\tilde{C}\right| > 0$, $\tilde{P} = (\tilde{p}_{nl})_{n,l}$ is an r-fuzzy L-clustering of X induced by $\tilde{C}$,*

$$\phi_X^{(r)}(\tilde{P}) \;\leq\; \phi_X^{(r)}(\tilde{C}) = \phi_X^{(r)}(\tilde{C}, \tilde{P}) \;\leq\; (1+\epsilon) \cdot \phi_X^{(r)}(C) \;,$$

and

$$\forall l \in [L] : \; \exists n \in [N] : \; \tilde{p}_{nl} \geq \frac{\epsilon}{2\mathbf{i}_r K^2} \;.$$

The algorithms' runtime is $\mathcal{O}(|X| \cdot D \cdot K \cdot \mathbf{t}_r(K))$.

In order to prove this result, let us start by considering the removal of a single mean vector whose corresponding cluster is negligible.

Lemma 6.5 (remove a single negligible cluster)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$, $\epsilon \in (0, 1]$, an $\mathbf{i}_r$-increase-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and let P be a soft K-clustering $P = (p_{nk})_{n \in [N], k \in [K]}$ with $K \geq 2$.*

Assume that for some $l \in [K]$ we have

$$\forall n \in [N] : \; p_{nl} \leq \frac{\epsilon}{\mathbf{i}_r K} \;.$$

Then, then there exists a soft $(K-1)$-clustering $\tilde{P} = (\tilde{p}_{nk})_{k \in [K] \setminus \{l\}, n \in [N]} \in \Delta_{N, K-2}$ such that

$$\forall n \in [N] : \; \forall k \in [K] \setminus \{l\} : \; \tilde{p}_{n,k} \geq p_{nk}$$

and such that for all $(\mu_k)_{k \in [K]} \subset \mathbb{R}^D$ it holds

$$\phi_X^{(r)}\left(\left(\mu_k\right)_{k \in [K] \setminus \{l\}}, \tilde{P}\right) \leq \; (1+\epsilon) \cdot \phi_X^{(r)}\left((\mu_k)_{k \in [K]}, P\right) \;.$$

Proof. Write $X = ((x_n, w_n))_{n \in [N]}$. Assume that for some $l \in [L]$ we have $\forall n \in [N] : \; p_{nl} \leq \frac{\epsilon}{\mathbf{i}_r K}$. Consider an arbitrary $n \in [N]$. Since $\sum_{k=1}^K p_{nk} = 1$ and $p_{nl} \leq \frac{1}{\mathbf{i}_r K} \leq \frac{1}{K}$, there exists some $k(n) \in [K] \setminus \{l\}$ such that $p_{nk(n)} \geq \frac{1}{K}$. Hence,

$$p_{nl} \leq \frac{\epsilon}{\mathbf{i}_r K} \leq \frac{\epsilon}{\mathbf{i}_r} \cdot p_{nk(n)} \;. \tag{6.1}$$

Let $\tilde{P} = (\tilde{p}_{nk})_{k \in [K] \setminus \{l\}, n \in [N]}$ be the soft clustering where, for all $n \in [N]$, we have $\tilde{p}_{nk} = p_{nk}$ for all $k \in [K] \setminus \{l, k(n)\}$ and $\tilde{p}_{n\,k(n)} = p_{n\,k(n)} + p_{nl}$. Let $C = (\mu_k)_{k \in [K]} \subseteq \mathbb{R}^D$ be an arbitrary but fixed set of means. Let

$$\begin{aligned}
C_{-l} &:= \left(\mu_k\right)_{k \in [K] \setminus \{l\}} \;, \\
\bar{\phi}_{l,k(\cdot)} &:= \sum_{n=1}^{N} \sum_{o \in [K] : \; o \neq l \wedge o \neq k(n)} r(p_{no}) w_n \left\| x_n - \mu_o \right\|_2^2 \;, \text{ and} \\
\phi_l &:= \sum_{n=1}^{N} r(p_{nl}) w_n \left\| x_n - \mu_l \right\|_2^2 \;.
\end{aligned}$$

Then,

$$\begin{aligned}
\phi_X^{(r)}(C_{-l},\tilde{P}) &= \sum_{n=1}^{N} r(p_{nk(n)}+p_{nl})w_n \left\|x_n-\mu_{k(n)}\right\|_2^2+\bar{\phi}_{l,k(\cdot)} \\
&\leq \sum_{n=1}^{N} r\left(\left(1+\frac{\epsilon}{\mathbf{i}_r}\right)p_{nk(n)}\right)w_n \left\|x_n-\mu_{k(n)}\right\|_2^2+\bar{\phi}_{l,k(\cdot)} && \text{(Eq. (6.1), } r \text{ is increasing)} \\
&\leq \sum_{n=1}^{N} (1+\epsilon)\cdot r(p_{nk(n)})w_n \left\|x_n-\mu_{k(n)}\right\|_2^2+\bar{\phi}_{l,k(\cdot)} && (r \text{ is } \mathbf{i}_r\text{-increase-bounded}) \\
&\leq \sum_{n=1}^{N} (1+\epsilon)\cdot r(p_{nk(n)})w_n \left\|x_n-\mu_{k(n)}\right\|_2^2+\bar{\phi}_{l,k(\cdot)}+\phi_l && (\phi_l \geq 0) \\
&\leq (1+\epsilon)\cdot\phi_X^{(r)}(C,P)\,, && (\bar{\phi}_{l,k(\cdot)},\phi_l \geq 0)
\end{aligned}$$

which yields the claim. □

Now we could try to remove all (currently) negligible clusters at once. However, we want to guarantee that the resulting *induced* r-fuzzy clustering has no negligible clusters. As we do not know how the induced clustering changes when we remove the mean vector of a negligible cluster, we cannot[1] preclude the possibility that we have to repeat this removal K times. Therefore, we remove clusters one by one as described in Algorithm 3.

Proof of Lemma 6.4. Observe that a soft 1-clustering of N elements is always equal to $\mathbb{1}_N$. Hence, by construction, the algorithm returns a set of means C_{-I} with $|C_{-I}| \geq 1$ such that the support of each cluster given by P_{-I} is at least $\tilde{\epsilon}/(\mathbf{i}_r K)$.

Assume that, at some point during its execution, the algorithm removes mean μ_k from the current set of means C_{-I}. Denote the resulting set of means by $C_{-I,k}$. Since $\frac{\tilde{\epsilon}}{\mathbf{i}_r\cdot K} \leq \frac{\tilde{\epsilon}}{\mathbf{i}_r\cdot(K-|I|)} = \frac{\tilde{\epsilon}}{\mathbf{i}_r\cdot|C_{-I}|}$ (cf. Algorithm 3), Lemma 6.5 implies that

$$\phi_X^{(r)}(C_{-I,k},\tilde{P}) \leq (1+\tilde{\epsilon})\cdot\phi_X^{(r)}(C_{-I})\,.$$

Now consider the final index set I. By applying this argument repeatedly to each index that was added to I, we obtain the desired approximation factor: We have

$$\phi_X^{(r)}(C_{-I}) \leq (1+\tilde{\epsilon})^K\cdot\phi_X^{(r)}(C) \leq (1+\epsilon)\cdot\phi_X^{(r)}(C)\,,$$

where the last inequality is due to Lemma A.1 and the fact that $\tilde{\epsilon} = \epsilon/2K$.

For the runtime, observe that the algorithm removes at most K means. After a mean has been removed, we have to compute a new r-fuzzy clustering of X which needs time $\mathcal{O}(|X|\cdot D\cdot \mathbf{t}_r(K))$. Between the removal of two means, we have to check whether a cluster has too low a support, which needs time $\mathcal{O}(|X|)$ (and is done at most K times). Thus, the overall runtime is $\mathcal{O}(K\cdot(|X|\cdot D\cdot\mathbf{t}_r(K)+|X|\cdot K)) \subseteq \mathcal{O}(|X|\cdot D\cdot K\mathbf{t}_r(K))$. □

[1] Well, except for the classical fuzzy K-means problem where we know the form of the optimal soft assignments exactly. However, this possible speedup will make no difference for the main results of this thesis.

Chapter 7

Baselines

In this chapter, we derive some simple algorithms for the r-fuzzy K-means problem with performance guarantees. We pursue the following three ideas: First, we restrict the set of mean vectors to the finite set of points $\{x_n \mid n \in [N]\}$ that contains only points from the given data set $X = ((x_n, w_n))_{n\in[N]}$. We perform an exhaustive search through all solutions induced by means from this set. This approach is exactly the same as the approach by Hasegawa et al. (1993) for the K-means problem. Second, we restrict the set of soft clusterings to a finite set of soft clusterings whose single soft assignment probabilities p_{nk} are rational values $i_{nk}/B \in \mathbb{Q}$ with a bounded denominator $B \in \mathbb{N}$. Again, we perform an exhaustive search through all solutions induced by soft clusterings from this set. Third, we analyse the use of a K-means approximation algorithm.

Overview. We summarize our contribution in Section 7.1. We do not present related work as, to the best of our knowledge, prior to Blömer et al. (2016), there have been no algorithms with approximation guarantees for a r-fuzzy K-means problem with $r \neq \mathrm{id}$. For an overview of work related to the K-means problem, we refer back to Section 4.4. In Section 7.2 and Section 7.3, we present our exhaustive searches through reduced sets of possible means and soft clusterings, respectively. In Section 7.4, we consider K-means approximation algorithms.

7.1 Contribution

First, we show that there is a 2-approximation algorithm for the r-fuzzy K-means problem that runs in time $\mathcal{O}(|X|^K \cdot D)$. Unlike the other algorithms, it works for for *arbitrary* fuzzifier functions. This result is an analogon of the results of Hasegawa et al. (1993).

Second, we show that there is a $(1+\epsilon)$-approximation algorithm for the r-fuzzy K-means problem with $\mathbf{i}_r$-increase-bounded fuzzifier functions r. Its runtime is linear in the dimension, but exponential in the number of data points $|X|$ and the number of clusters K.

Third, we show that every α-approximation for the K-means problem also induces an $\left(\alpha \cdot \mathbf{c}_r(K)^{-1}\right)$-approximation for the corresponding r-fuzzy K-means problem if the given fuzzifier function r is $\mathbf{c}_r$-contribution-bounded. For example, this implies that the algorithm by Matoušek (2000) is a constant factor approximation algorithm for the e_γ-fuzzy K-means problem with runtime $\mathcal{O}(|X|\log(|X|)^K \epsilon^{-2D})$ (for unweighted data sets). For an overview of all of our approximation algorithms we refer to Chapter 13.

7.2 2-Approximation Algorithm

The following result is an analogon of the results of Hasegawa et al. (1993) for the K-means problem: By testing all vectors of means that consist of points from the given point set X,

we can find a 2-approximation to the r-fuzzy K-means problem. The key to this result is the observation that the r-fuzzy K-means cost function can be expressed via pairwise distances between points (see Corollary 2.23).

Algorithm 4 Exhaustive Search for Means

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$

```
C* := (); φ* := ∞
for all C ⊆ X with |C| = K do
    if φ_X^(r)(C) < φ* then
        φ* := φ_X^(r)(C); C* := C
return C*
```

Theorem 7.1. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, and a fuzzifier function r, Algorithm 4 computes K means $C \subseteq \mathbb{R}^D$ such that*

$$\phi_X^{(r)}(C) \leq 2 \cdot \phi_{(X,K,r)}^{OPT} .$$

The algorithms' runtime is

$$\mathcal{O}\left(|X|^{K+1} \cdot D \cdot \mathbf{t}_r(K)\right) .$$

Proof. The correctness is a consequence of the following Lemma 7.2. For the runtime, observe that there are at most $|X|^K$ different vectors C of length K with elements from X. Evaluating the cost of a solution induced by such vector C needs time $\mathcal{O}(|X| D \mathbf{t}_r(K) + |X| KD) = \mathcal{O}(|X| D \mathbf{t}_r(K))$ since $\mathbf{t}_r(K) \in \Omega(K)$ (Assumption 5.19). □

We stress the fact that this result does *not* impose any additional constraints on the fuzzifier function, except for the constraint that we need to be able to compute induced r-fuzzy clusterings (Assumption 5.19). This result is applicable for *all* the fuzzifier functions that we presented in Section 5.3. For each of these fuzzifier functions, the algorithm computes a 2-approximation to the r-fuzzy K-means problem in running time $\mathcal{O}(|X|^{K+1} \cdot DK \log(K))$.

Lemma 7.2. *Let $X = ((x_n, w_n))_{n \in [N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, and let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function. Let $P = (p_{nk})_{n \in [N], k \in [K]}$ be a soft K-clustering, and let $C = (\mu_k)_{k \in [K]}$ be the r-fuzzy means of X induced by P. Set $\tilde{C} := (\tilde{\mu}_k)_{k \in [K]} \subseteq X$ where $\tilde{\mu}_k \in \arg\min\left\{ \left\| x_n - \mu_k \right\|_2 \,\middle|\, x_n \in X \right\}$ for each $k \in [K]$. Then, $\phi_X^{(r)}(\tilde{C}) \leq 2 \cdot \phi_X^{(r)}(P)$.*

Proof. This proof is an analogon of the proof of Hasegawa et al. (1993). Recall that

$$\phi_X^{(r)}(\tilde{C}) \leq \phi_X^{(r)}(\tilde{C}, P) = \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X, r(P))}, \tilde{\mu}_k\right) .$$

Then, with Lemma 2.20, we can conclude

$$\begin{aligned} \mathbf{d}\left(A_k^{(X,r(P))}, \tilde{\mu}_k\right) &= \mathbf{w}\left(A_k^{(X,r(P))}\right) \left\| \tilde{\mu}_k - \mathbf{m}\left(A_k^{(X,r(P))}\right) \right\|_2 + \mathbf{d}\left(A_k^{(X,r(P))}\right) \\ &= \sum_{n=1}^{N} r(p_{nk}) w_n \left(\left\| \tilde{\mu}_k - \mathbf{m}\left(A_k^{(X,r(P))}\right) \right\|_2^2 + \left\| \mathbf{m}\left(A_k^{(X,r(P))}\right) - x_n \right\|_2^2 \right) . \end{aligned}$$

From Lemma 5.4 and the definition of C, we know that $\mu_k = \mathbf{m}\left(A_k^{(X,r(P))}\right)$. By definition of $\tilde{C}$, we have $\left\| \tilde{\mu}_k - \mu_k \right\|_2 \leq \left\| x_n - \mu_k \right\|_2$ for all $n \in [N]$. Hence,

$$\begin{aligned} \mathbf{d}\left(A_k^{(X,r(P))}, \tilde{\mu}_k\right) &= \sum_{n=1}^{N} r(p_{nk}) w_n \left(\left\| \tilde{\mu}_k - \mu_k \right\|_2^2 + \left\| \mu_k - x_n \right\|_2^2 \right) \\ &\leq 2 \sum_{n=1}^{N} r(p_{nk}) w_n \left\| x_n - \mu_k \right\|_2^2 = 2\phi_X^{(r)}(C, P) = 2\phi_X^{(r)}(P) . \end{aligned}$$

□

7.3 $(1+\epsilon)$-Approximation Algorithm

Consider an arbitrary soft clustering $P = (p_{nk})_{n,k}$ and a distorted version $\tilde{P} = (\tilde{p}_{nk})_{n,k}$ thereof where $\forall n,k:\ \tilde{p}_{nk} \leq (1+\epsilon/\mathbf{i}_r)p_{nk}$ for some $\mathbf{i}_r \in [1,\infty)$. If r is an $\mathbf{i}_r$-increase-bounded fuzzifier, then we know that the r-fuzzy K-means cost of $\tilde{P}$ is at most a factor $(1+\epsilon)$ larger than the cost of P. In the following, this observation helps us to construct a set of soft clusterings Δ such that, for each possible soft clustering P, there is a soft clustering $\tilde{P} \in \Delta$ whose cost is at most a factor $(1+\epsilon)$ larger than the cost of P. In particular, there is some soft clustering $\tilde{P} \in \Delta$ whose cost is close to the cost of an optimal solution. Hence, by an exhaustive search through the set Δ, we can find a $(1+\epsilon)$-approximation.

Algorithm 5 Exhaustive Search for a Clustering

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$

1: $P^* := (); \phi^* := \infty$
2: $B := \left\lceil \frac{K \cdot \mathbf{i}_r}{\epsilon} \right\rceil$
3: **for all** soft K-clusterings $P = (i_{nk}/B)_{n,k}$ of X, where $\forall n \in [N] \forall k \in [K]:\ i_{nk} \in [B]$ **do**
4: **if** $\phi_X^{(r)}(P) < \phi^*$ **then**
5: $P^* := P; \phi^* := \phi_X^{(r)}(P)$
6: **return** P^*

Theorem 7.3. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, an $\mathbf{i}_r$-increase-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the value $\mathbf{i}_r \in [1,\infty)$, $\epsilon \in [0,1]$, Algorithm 5 computes a soft K-clustering P of X such that*

$$\phi_X^{(r)}(P) \leq (1+\epsilon)\,\phi_{(X,K,r)}^{OPT} .$$

The algorithms' runtime is

$$2^{\mathcal{O}(K|X|\cdot\log(\mathbf{i}_r/\epsilon))} \cdot D .$$

Proof. Write $X = ((x_n, w_n))_{n\in[N]}$. As in the algorithm, let $B := \lceil K \cdot \mathbf{i}_r/\epsilon \rceil$. For each $n \in [N]$ and $k \in [K]$, let $\hat{i}_{nk} := \left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) p_{nk} B$. Then, for all $n \in [N]$, we have $\sum_{k=1}^K \hat{i}_{nk} = \left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) B \geq B$ and

$$\sum_{k=1}^K \frac{\lfloor \hat{i}_{nk} \rfloor}{B} = \sum_{k=1}^K \frac{\hat{i}_{nk} - (\hat{i}_{nk} - \lfloor \hat{i}_{nk} \rfloor)}{B} = \sum_{k=1}^K \left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) p_{nk} - \sum_{k=1}^K \frac{\hat{i}_{nk} - \lfloor \hat{i}_{nk} \rfloor}{B} \geq \left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) - K \cdot \frac{1}{B} \geq 1 ,$$

where in the last inequality we use that $B \geq K \cdot \mathbf{i}_r/\epsilon$.

Hence, there exist values $i_{nk} \in \mathbb{N}_0$ such that $i_{nk} \leq \lfloor \hat{i}_{nk} \rfloor$ and $\sum_{k=1}^K i_{nk} = B$. Set $\tilde{P} := \left(\frac{i_{nk}}{B}\right)_{n,k}$. Let $C = (\mu_k)_{k\in[K]} \subset \mathbb{R}^D$ be the r-fuzzy means of X induced by P. Then,

$$\begin{aligned}
\phi_X^{(r)}(\tilde{P}) &\leq \phi_X^{(r)}(C, \tilde{P}) \\
&= \sum_{n=1}^N \sum_{k=1}^K r\left(\frac{i_{nk}}{B}\right) w_n \left\|x_n - \mu_k\right\|_2^2 \\
&\leq \sum_{n=1}^N \sum_{k=1}^K r\left(\frac{\hat{i}_{nk}}{B}\right) w_n \left\|x_n - \mu_k\right\|_2^2 && (i_{nk} \leq \hat{i}_{nk} \text{ and } r \text{ is increasing}) \\
&= \sum_{n=1}^N \sum_{k=1}^K r\left(\left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) p_{nk}\right) w_n \left\|x_n - \mu_k\right\|_2^2 && (\hat{i}_{nk} = \left(1 + \frac{\epsilon}{\mathbf{i}_r}\right) p_{nk} B) \\
&\leq \sum_{n=1}^N \sum_{k=1}^K (1+\epsilon) \cdot r(p_{nk}) w_n \left\|x_n - \mu_k\right\|_2^2 && (\mathbf{i}_r\text{-increase-bounded}) \\
&= (1+\epsilon) \cdot \phi_X^{(r)}(C) .
\end{aligned}$$

This yields the first part of the claim.

For the runtime, apply the well-known stars and bars method: The number of K tuples of non-negative integers whose sum is B is equal to the number of sets of size $K-1$ whose elements are taken from a set of size $B+K-1$, i.e., $\binom{B+K-1}{K-1}$. Hence, the total number of clusterings considered by the algorithm is $\binom{B+K-1}{K-1}^N$. Recall that, $\mathbf{i}_r K/\epsilon \leq B \leq \mathbf{i}_r K/\epsilon + 1$. Moreover, for all $n \in \mathbb{N}$ and $k \in [n]$, we have $\binom{n}{k} \leq \left(\frac{e \cdot n}{k}\right)^k$ (Cormen et al., 2001, p. 1097). Hence,

$$\binom{B+K-1}{K-1}^N \leq \left(\frac{(B+K-1)\cdot e}{K-1}\right)^{(K-1)N} \leq \left(\left(\frac{B}{K-1}+1\right)\cdot e\right)^{KN} \leq \left(\frac{B}{K}\cdot 3e\right)^{KN} \in \left(\frac{\mathbf{i}_r}{\epsilon}\right)^{\mathcal{O}(KN)} .$$

Evaluating the r-fuzzy means induced by P (see Lemma 5.4) and the cost of the resulting induced solution needs time $\Theta(NKD)$. This yields the claim. □

Special Cases. Recall our results from Section 5.3. For the classical fuzzy K-means problem, we know that the corresponding fuzzifier p_m is $(2m)$-increase-bounded. This implies that Algorithm 5 is a $(1+\epsilon)$-approximation algorithm for the classical K-means problem with runtime

$$2^{\mathcal{O}(K|X|\log(m/\epsilon))} \cdot D .$$

In contrast, for the r-fuzzy K-means problem with the exponential fuzzifier e_γ, this algorithm is *not* applicable since the fuzzifier e_γ is simply not increase-bounded (see Lemma 5.33).

7.4 $\left(\mathtt{const}\cdot \mathbf{c}_r(K)^{-1}\right)$-Approximation Algorithm

The following result is a straightforward application of Lemma 6.1.

Theorem 7.4 (K-means α-approximation)**.** *Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a $\mathbf{c}_r$-contribution-bounded fuzzifier function. Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$ and let $C \subseteq \mathbb{R}^D$ be a vector containing $K \geq 1$ means such that $\mathrm{km}_X(C) \leq \alpha \cdot \mathrm{km}^{OPT}_{(X,K)}$. Then, $\phi^{(r)}_X(C) \leq \alpha \cdot \mathbf{c}_r(K)^{-1} \cdot \phi^{OPT}_{(X,K,r)}$.*

Proof. Let $C^* \subseteq \mathbb{R}^D$, $|C^*| = K$, such that $\phi^{OPT}_{(X,K,r)} = \phi^{(r)}_X(C^*)$. With Lemma 6.1, we can conclude that $\phi^{(r)}_X(C) \leq \mathrm{km}_X(C) \leq \alpha \cdot \mathrm{km}^{OPT}_{(X,K)} \leq \alpha \cdot \mathrm{km}_X(C^*) \leq \alpha \cdot \mathbf{c}_r(K)^{-1} \cdot \phi^{(r)}_X(C^*) = \alpha \cdot \mathbf{c}_r(K)^{-1} \cdot \phi^{OPT}_{(X,K,r)}$. □

This means that we can simply use an K-means clustering algorithm for the r-fuzzy K-means problem if we accept that the approximation factor, which is guaranteed with regard to the K-means problem, might worsen by a factor $\mathbf{c}_r(K)^{-1}$.

Special Cases. This result is applicable for the e_γ-fuzzy K-means problem with the exponential fuzzifier e_γ. Recall our results from Section 5.3.4. We know that this function is $\mathbf{c}_{\mathrm{e}_\gamma}$-contribution-bounded by a *constant*: $\mathbf{c}_{\mathrm{e}_\gamma}(K) = \gamma/(e^\gamma - 1)$ for all $K \in \mathbb{N}$. This means that, for instance, the algorithm of Matoušek (2000) is a $((e^\gamma-1)/\gamma)$-approximation algorithm for the e_γ-fuzzy K-means problem with runtime $\mathcal{O}(|X|\log(|X|)^K \epsilon^{-2D})$ (for unweighted data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$).

Next, consider the classical fuzzy K-means problem. The fuzzifier function p_m is $\mathbf{c}^*_{\mathrm{p}_m}$-contribution-bounded with $\mathbf{c}^*_{\mathrm{p}_m}(K) = 1/K^{m-1}$ for all $K \in \mathbb{N}$ (see Section 5.3.2). Hence, an α-approximation algorithm for the K-means problem is an $(\alpha \cdot K^{m-1})$-approximation algorithm for the classical fuzzy K-means problem with fuzzifier value $m \in (1,\infty)$. So, for example, the famous K-means++ algorithm by Arthur and Vassilvitskii (2007) yields, in expectation, an $\mathcal{O}(\log(K)\cdot K^{m-1})$-approximation to the classical fuzzy K-means problem and needs only linear time $\mathcal{O}(|X|KD)$ (for unweighted data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$).

"By a small sample, we may judge of the whole piece."

Miguel de Cervantes from Don Quixote

Chapter 8

Superset Sampling for Fuzzy Clusters

The key observation behind the superset sampling technique is that certain properties of a data set can be determined, with constant probability and to some precision, by examining a small subset of the data that has been sampled *uniformly at random*.

Inaba et al. (1994) showed that the mean $\mathbf{m}(S)$ of a uniform sample S from an unweighted data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$ is close to the mean $\mathbf{m}(X)$, with high probability. More precisely, their squared Euclidean distance is at most a small multiple of the variance

$$\mathbf{var}(X) = \frac{\mathbf{d}(X)}{|X|} = \frac{1}{|X|} \sum_{x \in X} \|x - \mathbf{m}(X)\|_2^2 \ .$$

Assume that we want to find the mean of an unknown hard cluster $A \subseteq X$. If A has not too small a size, then a uniform sample S_X from X contains a certain fraction of the points in A. That is, $S_A \subseteq S_X \cap A$ does not have too small a size. Since S_X is a uniform sample from X, the subset S_A is a uniform sample of A. With the initial observation, one can conclude that the mean $\mathbf{m}(S_A)$ of the sample is close to the mean $\mathbf{m}(A)$ of the cluster, in the sense that their Euclidean distance is bounded by a small multiple of the variance $\mathbf{var}(A)$ of the cluster. To sum up, by a clever way of uniform sampling and an exhaustive enumeration, we can construct a set that contains a good approximation to the mean of an unknown hard cluster, with high probability. By repeating this process and exhaustively enumerating all possible combinations, we can construct a set of candidates that contains approximations to the means of K unknown hard clusters, with high probability.

This technique is obviously useful for the K-means problem and probably for similar hard clustering problems. However, the r-fuzzy K-means problem is no such problem. The key ingredient that, nevertheless, enables us to apply the superset sampling technique is our result from Chapter 3. There, we showed that for each soft clustering there exist hard clusters that exhibit characteristics similar to those of the soft clusters.

Overview. First, we give an overview of related work in Section 8.1 and briefly state our main contributions in Section 8.2. In Section 8.3, we describe the implications of our hard-to-soft technique from Chapter 3 with respect to the r-fuzzy K-means problem. In Section 8.4, we describe two different ways in which the superset sampling technique can be applied to approximate the means of K unknown hard clusters. Then, in Section 8.5, we focus on the details that need to be considered when combining these results. Finally, the results of this combination are presented and discussed in Section 8.6.

Publication. In this chapter, we generalize and discuss the results from (Blömer et al., 2016, Theorem 2+4), which deal with the classical fuzzy K-means problem.

8.1 Related Work

First of all, the superset sampling technique has been used to tackle the K-means problem: Inaba et al. (1994), who also present the first exact algorithm for the K-means problem that runs in time $\mathcal{O}(|X|^{KD+1})$, applies this technique to show that there is a $(1+\epsilon)$-approximation algorithm for the 2-means problem (i.e., $K=2$) that runs in time $\mathcal{O}(|X|\cdot(1/\epsilon)^D)$. Later, Kumar et al. (2004, 2010) use it to show that there is a randomized $(1+\epsilon)$-algorithm for the K-means problem (with arbitrary K) whose runtime is linear in the dimension D. More precisely, its runtime is $2^{(K/\epsilon)^{\mathcal{O}(1)}}\cdot D\,|X|$. Ackermann et al. (2010) extend these results to K-median problems with respect to dissimilarity measures where the respective 1-median problem can be approximated by taking a random sample and solving the 1-median problem for this sample exactly. This covers, for instance, the K-median problem with respect to the Kullback-Leibler divergence and other special Bregman divergences.

8.2 Contribution

First and foremost, we state the first polynomial-time approximation scheme (PTAS) for the r-fuzzy K-means problem with respect to unweighted data sets and for $[0,1]$-reducing fuzzifier functions r that are increase-bounded by a constant, under the assumption that the number of clusters K is constant. This result covers all fuzzifier functions presented in Section 5.3, except the exponential fuzzifiers e_γ.

Second, we state a randomized version of this algorithm which is substantially (asymptotically) faster and returns a solution whose cost is at most a factor $(1+\epsilon)$-worse than the best solution whose r-fuzzy clusters have a certain minimum weight (Section 8.6.2). Though there is some similarity to a constraint clustering approach, this result needs to be handled with caution. We discuss its flaws and present a (nonetheless) reasonable application of this algorithm in Section 8.6.2.

8.3 From Fuzzy Clusters to Hard Clusters

In the first part of this thesis we showed that for clusters with not too small a weight there always exist hard clusters whose statistics are similar to those of the soft clusters. More precisely, we showed that this property holds for clusters defined by probabilistic membership values. Therefore, it also applies to r-fuzzy clusters where the fuzzifier function r is $[0,1]$-reducing.

Theorem 8.1. *Let $\epsilon\in(0,1]$, $X\in\mathrm{Dom}\left(\mathbb{R}^D,\mathbb{R}_+\right)$, $P\in\Delta_{|X|,K-1}$, and let $f:\mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$ be a $[0,1]$-reducing function such that*

$$\forall k\in[K]\colon\ \mathbf{w}\left(A_k^{(X,r(P))}\right)\geq\frac{16}{\epsilon}\cdot K\cdot \mathtt{w}_{\max}^{(X)}\,.$$

Then, there exist pairwise disjoint hard clusters $A_1,\dots,A_K\subseteq X$ such that for all $k\in[K]$ we have

$$\mathbf{w}(A_k)\geq\frac{1}{2}\mathbf{w}\left(A_k^{(X,r(P))}\right)\,, \tag{8.1}$$

$$\left\|\mathbf{m}(A_k)-\mathbf{m}\left(A_k^{(X,r(P))}\right)\right\|_2^2\leq\frac{\epsilon}{2}\cdot\mathbf{var}\left(A_k^{(X,r(P))}\right)\,,\ \textit{and} \tag{8.2}$$

$$\mathbf{d}(A_k)\leq 4K\cdot\mathbf{d}\left(A_k^{(X,r(P))}\right)\,. \tag{8.3}$$

Proof. Since r is $[0,1]$-reducing, $r(P)$ is a probabilistic membership matrix. Thus, applying Theorem 3.21 to the membership values $r_{nk}:=r(p_{nk})$, $n\in[N]$ and $k\in[K]$, yields the claim. □

Unfortunately, to the best of our knowledge, the hard clusters A_k do not exhibit any concrete structure: Points that belong to the same cluster are not necessarily "close" to one another (i.e., there is no locality property). In particular, the convex hulls of the hard clusters will probably overlap. The hard clusters do not even necessarily cover X (i.e., $\dot{\cup}_{k\in[K]}A_k \neq X$). Due to these properties, it is not clear how techniques that do not solely rely on sampling can be applied. For instance, we presume that the sample and prune technique from Ackermann et al. (2010) and the K-means++ algorithm (see Arthur and Vassilvitskii, 2007) require a locality property, while the algorithm by Bhattacharya et al. (2016), which is based on the K-means++ algorithm, requires that the hard clusters form a hard clustering.

8.4 Applying Superset Sampling

In this section, we focus on the problem of approximating the means of unknown hard clusters $A_k \subseteq X$ of some data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$ with rational weights. If we know that the weight of each cluster A_k is a certain fraction of the weight of X, then we can tackle this problem via superset sampling.

From (Ackermann, 2009) we directly obtain the following lemma.

Lemma 8.2 (Weighted Superset Sampling)**.** *Let $\alpha \in (0,1]$, $\epsilon \in (0,1]$, and $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$. Consider an arbitrary but fixed (unknown) data set $A \subseteq X$ with*

$$\mathbf{w}(A) \geq \alpha \cdot \mathbf{w}(X) \ .$$

Let $S \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$ be the data set that contains at least $4/(\alpha\epsilon)$ random samples from the unweighted *data set $((x_n, 1))_{n\in[N]}$: Each sample is drawn independently and identically according to the distribution that assigns data point $(x_n, 1)$ probability $w_n/\mathbf{w}(X)$. Then, with a probability of at least $1/10$, there exists a data set $A' \subseteq S$ of size $|A'| = \lceil 2/\epsilon \rceil$ satisfying*

$$\left\|\mathbf{m}(A) - \mathbf{m}(A')\right\|_2^2 \leq \epsilon \cdot \mathbf{var}(A) \ .$$

Proof. A proof can be found in (Ackermann, 2009, p. 75). □

Now consider K unknown hard clusters $C_1, \ldots, C_K \subseteq X$. Assume that the weight $\mathbf{w}(A_k)$ of *each* cluster is at least a certain fraction of the total weight $\mathbf{w}(X)$. Then, by repeatedly sampling candidate means as in Lemma 8.2 and combining these candidate means in each way possible, we obtain a set of candidates $C = (\tilde{\mu}_k)_{k\in[K]}$ where for each $k \in [K]$ the mean vector $\tilde{\mu}_k$ is close to the mean $\mathbf{m}(A_k)$ of the hard cluster A_k.

Theorem 8.3. *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, and $\alpha \in (0,1]$. Let $(A_k)_{k\in[K]}$ be an arbitrary but fixed vector of (unknown) hard clusters $A_k \subseteq X$.*

Given X, K, ϵ, and α, Algorithm 6 constructs a set $T \subset (\mathbb{R}^D)^K$ such that, with constant probability, there exists a vector $\left(\mu_k\right)_{k\in[K]} \in T$ such that for all $k \in [K]$ with

$$\mathbf{w}(A_k) \geq \alpha \cdot \mathbf{w}(X)$$

we have

$$\left\|\mu_k - \mathbf{m}(A_k)\right\|_2^2 \leq \epsilon \cdot \mathbf{var}(A_k) \ .$$

The size of T is

$$|T| \in 2^{\mathcal{O}(K \log\log(K) \log(1/(\alpha\epsilon)))} \ .$$

The algorithms' runtime is $\mathcal{O}(|X| + |T| \cdot D)$.

Algorithm 6 Superset Sampling

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, and $\alpha \in (0,1]$

Start with an empty data set $M := \emptyset$
Pre-compute the distribution p over X with $p((x_n, w_n)) = \frac{w_n}{\mathbf{w}(X)}$ for all $n \in [N]$.
for all $r \in [\lceil 10\log(2K)\rceil]$ **do**
    Start with an empty data set $S_r := \emptyset$.
    for all $s \in [\lceil 4/(\alpha\epsilon)\rceil]$ **do**
        Sample (x, w) from X according to p.
        $S_r := S_r \dot\cup ((x, 1))$
    $M_r := \emptyset$
    for all $S' \subset S_r$ with $\left|S'\right| = \lceil 2/\epsilon\rceil$ **do**
        $M_r := M_r \dot\cup \left(\mathbf{m}(S')\right)$
    $M := M \cup M_r$
$T := M^K$
return T

Proof. Let $R = \lceil 10\log(2K)\rceil$. Observe that M contains all means $\mathbf{m}(S')$ of data sets S' with $S' \subset S_r$, $\left|S'\right| = \lceil 2/\epsilon\rceil$, and $r \in [R]$. Fix an arbitrary $k \in [K]$ with $\mathbf{w}(A_k) \geq \alpha \cdot \mathbf{w}(X)$. According to Lemma 8.2, with a probability of at least $p := 1 - (9/10)^R$, there is a $\mu_k \in M$ satisfying

$$\left\|\mathbf{m}(A_k) - \mu_k\right\|_2^2 \leq \epsilon \cdot \mathbf{var}(A_k) \ .$$

Since $R \geq 10\log(2K)$, we have that $(9/10)^R \leq (2K)^{\log(9/10)\cdot 10} \leq 1/(2K)$. Hence, $p \geq 1 - 1/(2K)$. By taking the union bound, we obtain that, with a probability of at least $1/2$, T contains a tuple $(\mu_k)_{k\in[K]}$ with the desired property.

Next, we analyse the size of T. Thee algorithm constructs $R = \lceil 10\log(2K)\rceil$ data sets S_r with $|S_r| = \lceil 4/(\alpha\epsilon)\rceil$. For each of these data sets S_r, the algorithm constructs at most

$$|M_r| = \mathcal{O}\left(\left(\frac{4}{\alpha\epsilon}\right)^{2/\epsilon}\right)$$

vectors. T contains all possible combinations of K vectors from $M = \dot\cup_{r\in[R]} M_r$. Therefore,

$$\begin{aligned}
|T| &\in \mathcal{O}\left(\left(R \cdot \left(\frac{4}{\alpha\epsilon}\right)^{2/\epsilon}\right)^K\right) \\
&\subseteq \mathcal{O}\left(16^K \cdot (\log(2K))^K \cdot 2^{K\log(4/(\alpha\epsilon))}\right) && (R \leq 16\log(2K)) \\
&\subseteq \mathcal{O}\left(2^{4K} \cdot 2^{K\cdot\log(\log(2K))} \cdot 2^{K(2+\log(1/(\alpha\epsilon)))}\right) \\
&\subseteq 2^{\mathcal{O}(K\log\log(K)\log(1/(\alpha\epsilon)))} \ .
\end{aligned}$$

Now consider the runtime. The construction of *all* data sets S_r with $r \in [R]$ needs time $\mathcal{O}\left(R \cdot \left(|X| + D \cdot \frac{1}{\alpha\epsilon}\right)\right)$ (Vose, 1991; Knuth, 1997). Enumerating all the data sets S' with size $\lceil 2/\epsilon\rceil$ and computing the mean of each S' needs time $\mathcal{O}(|M| \cdot \frac{1}{\epsilon} D)$. Enumerating all elements in T needs time $\mathcal{O}(|T|D) = \mathcal{O}(|M|^K D)$. Hence, the overall running time is

$$\mathcal{O}\left(|X| + R \cdot \left(\frac{D}{\alpha\epsilon}\right) + R \cdot |M| \cdot \frac{1}{\epsilon} D + |T| \cdot D\right) \subseteq \mathcal{O}\left(|X| + |T| \cdot D\right) \ .$$

□

Observe that each candidate mean constructed by Algorithm 6 is the mean of a data set S' of size $\lceil 2/\epsilon\rceil$. Instead constructing these sets S' via sampling, we can simply enumerate all

possible data sets S' of size $\lceil 2/\epsilon \rceil$. In other words, we can replace the random sampling by an exhaustive enumeration. Thereby, we obtain a deterministic algorithm that computes a set of candidate means. Most notably, this algorithm does not need to know the ratio between the weights of the clusters and data points.

Algorithm 7 De-Randomized Superset Sampling

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, and $\alpha \in (0,1]$

1: $M := \emptyset$
2: **for all** data sets $S' \subseteq \mathrm{Dom}(\{x_n \mid (x_n, w_n) \in X\}, \{1\})$ with $\left|S'\right| = \lceil 2/\epsilon \rceil$ **do**
3: $\quad M := M \cup \left\{\mathbf{m}(S')\right\}$
4: $M := T^K$
5: **return** T

Corollary 8.4. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{Q}_+\right)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, and $\alpha \in (0,1]$, Algorithm 7 constructs a set $T \subset (\mathbb{R}^D)^K$ with the following properties: For all vectors $(A_k)_{k\in[K]}$ of sets $A_k \subseteq X$, there exists a vector $\left(\mu_k\right)_{k\in[K]} \in T$ such that for all $k \in [K]$ with $A_k \neq \emptyset$ we have*

$$\left\|\mu_k - \mathbf{m}(A_k)\right\|_2^2 \leq \epsilon \cdot \mathbf{var}(A_k) \ .$$

The size of T is

$$|T| \in |X|^{\mathcal{O}(K/\epsilon)} \ .$$

The algorithms' runtime is

$$D \cdot |X|^{\mathcal{O}(K/\epsilon)} \ .$$

Proof. On the one hand, observe that Lemma 8.2 holds for all choices of $\alpha \in (0,1]$, especially $\alpha = \min\{\mathbf{w}(A_k)/\mathbf{w}(X) \mid k \in [K],\ A_k \neq \emptyset\}$. On the other hand, observe that the candidates constructed in the proof of Theorem 8.3 are always means of sets of size $\lceil 2/\epsilon \rceil$. Combining both observations yields the claim regarding the approximation factor.

Observe that there are at most $|X|^{\lceil 2/\epsilon \rceil}$ data sets with points from $\{x_n \mid (x_n, w_n) \in X\}$ and size $\lceil 2/\epsilon \rceil$. Hence, the size of T is $|X|^{\mathcal{O}(K/\epsilon)}$. The algorithms' runtime is $\mathcal{O}\left(|M| \cdot \frac{2}{\epsilon} D + |T| \cdot D\right) \subseteq \mathcal{O}\left(|T| D\right) \subseteq D \cdot |X|^{\mathcal{O}(K/\epsilon)}$. □

Again, we stress the fact that this approach does not require that the weights $\mathbf{w}(A_k)$ of the unknown hard clusters A_k make up a certain fraction of the weight $\mathbf{w}(X)$.

8.5 Combining the Results

In this section, we show how the results from the previous two sections can be combined.

8.5.1 Approximation Factor

Let us start with the easy part: Consider an arbitrary soft clustering P. Assume that there are hard clusters that imitate the r-fuzzy clusters given by P as described in Theorem 8.1 (with $\epsilon/2$ instead of ϵ). Moreover, assume that we have found means that are close to the means of these hard clusters, as described in Theorem 8.3 and Corollary 8.4 (with $\epsilon/(32K)$ instead of ϵ). Then we have found an approximation to the corresponding r-fuzzy K-means problem:

Lemma 8.5 (combination)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ be a data set, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, and $K \in \mathbb{N}$.*

Consider a hard K-clustering $(A_k)_{k\in[K]}$ of X, $P \in \Delta_{N,K-1}$, and $(\mu_k)_{k\in[K]} \subset \mathbb{R}^D$ satisfying

$$\left\|\mu_k - \mathbf{m}(A_k)\right\|_2^2 \leq \frac{\epsilon}{32K} \cdot \mathbf{var}(A_k) \;, \tag{8.4}$$

$$\mathbf{w}(A_k) \geq \frac{1}{2} \cdot \mathbf{w}\left(A_k^{(X,r(P))}\right) \;, \tag{8.5}$$

$$\left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,r(P))}\right)\right\|_2^2 \leq \frac{\epsilon}{4} \cdot \mathbf{var}\left(A_k^{(X,r(P))}\right) \;, \textit{ and} \tag{8.6}$$

$$\mathbf{d}(A_k) \leq 4K \cdot \mathbf{d}\left(A_k^{(X,r(P))}\right) \;. \tag{8.7}$$

Then,

$$\phi_X^{(r)}((\mu_k)_{k\in[K]}) \leq (1+\epsilon)\phi_X^{(r)}(P) \;.$$

Proof. We start by observing that

$$\begin{aligned}
\phi_X^{(r)}((\mu_k)_{k\in[K]}) \leq& \phi_X^{(r)}((\mu_k)_{k\in[K]}, P) \\
=& \phi_X^{(r)}(P) + \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \cdot \left\|\mu_k - \mathbf{m}\left(A_k^{(X,r(P))}\right)\right\|_2^2 && \text{(Lemma 2.20)} \\
\leq& \phi_X^{(r)}(P) + 2\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \left\|\mu_k - \mathbf{m}(A_k)\right\|_2^2 && \text{(Lemma A.3)} \\
&+ 2\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,r(P))}\right)\right\|_2^2 \;.
\end{aligned}$$

We can bound the second term of this upper bound by

$$\begin{aligned}
2\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \left\|\mu_k - \mathbf{m}(A_k)\right\|_2^2 \leq& \frac{\epsilon}{16K} \cdot \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \cdot \frac{\mathbf{d}(A_k)}{\mathbf{w}(A_k)} && \text{(Equation (8.4))} \\
\leq& \frac{\epsilon}{8K} \cdot \sum_{k=1}^{K} \mathbf{d}(A_k) && \text{(Equation (8.5))} \\
\leq& \frac{\epsilon}{2} \cdot \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,r(P))}\right) && \text{(Equation (8.7))} \\
=& \frac{\epsilon}{2} \cdot \phi_X^{(r)}(P) \;.
\end{aligned}$$

Furthermore, we can bound the third term by

$$\begin{aligned}
2\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,r(P))}\right) \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,r(P))}\right)\right\|_2^2 \leq& \frac{\epsilon}{2} \cdot \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,r(P))}\right) && \text{(Equation (8.6))} \\
=& \frac{\epsilon}{2} \cdot \phi_X^{(r)}(P) \;.
\end{aligned}$$

Putting these inequalities together yields the claim. □

However, we can only apply Theorem 8.1, Theorem 8.3, and Corollary 8.4 if certain conditions are satisfied.

8.5.2 Removing the Restriction to Rational Weights

Note that Theorem 8.3 and Corollary 8.4 hold true only for data sets with rational weights. This is not really a problem because there is no application that requires our algorithms to deal with real-valued weights. Nonetheless, one can easily get rid of this restriction.

Lemma 8.6 (rounding the weights)**.** *Let* $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ *and* $\epsilon \in [0,1]$*. Set*

$$B := \left\lceil \frac{1}{\epsilon \cdot \mathtt{w}_{\min}^{(X)}} \right\rceil$$

and let $X^{\mathbb{Q}} := \left((x_n, w_n^{\mathbb{Q}})\right)_{n\in[N]}$ *be the re-weighted data set where*

$$\forall n \in [N]\colon\ w_n^{\mathbb{Q}} := \frac{\lceil w_n \cdot B \rceil}{B} \in \mathbb{Q}_+ \ .$$

Then we have

$$\forall n \in [N]\colon\ w_n \le w_n^{\mathbb{Q}} \le (1+\epsilon) w_n \ .$$

For all means $C \subset \mathbb{R}^D$*, soft clusterings* $P \in \Delta_{N,K-1}$*, and* $r : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$*, we have*

$$\phi_X^{(r)}(C,P) \le \phi_{X^{\mathbb{Q}}}^{(r)}(C,P) \le (1+\epsilon)\phi_X^{(r)}(C,P) \ .$$

Proof. Let $n \in [N]$. On the one hand, $\lceil w_n \cdot B \rceil / B \ge (w_n \cdot B)/B = w_n$. On the other hand, $\lceil w_n \cdot B \rceil / B \le (w_n \cdot B + 1)/B = w_n + 1/B$ where $1/B = 1/\left\lceil 1/(\epsilon \cdot \mathtt{w}_{\min}^{(X)}) \right\rceil \le 1/(1/(\epsilon \cdot \mathtt{w}_{\min}^{(X)})) = \epsilon \cdot \mathtt{w}_{\min}^{(X)} \le \epsilon \cdot w_n$. This yields the first part of the claim.

Using the first part of the claim and recalling the definition of $\phi_X^{(r)}(C,P)$ yields the second part of the claim. □

8.5.3 Removing the Restriction to Clusters with A Minimum Weight

Theorem 8.1 requires that each r-fuzzy cluster has a certain absolute weight. However, the given r-fuzzy clusters might have an arbitrarily small weight. To circumvent this problem, we want to increase these weights.

We know from Section 2.3.4 that we can construct a data set X_c whose weight is c times larger than the weight $\mathbf{w}(X)$ of the given data set X by adding c copies of each data point in X to X_c. If we make corresponding copies of the soft assignments of a soft clustering P, then we obtain a soft clustering P_c of X_c whose r-fuzzy clusters are c times heavier than the r-fuzzy clusters of X given by P.

Obviously, we can only make use of this approach if the following two properties are satisfied: First, we need to be able to make use of mean vectors C that we computed with respect to X_c instead of X. Fortunately, the cost function changes in a predictable way:

Corollary 8.7 (adding copies)**.** *Let* $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$*, and* $c \in \mathbb{N}$*. Let* X_c *be the data set that contains* c *copies of each data point from* X*.*

For every $P \in \Delta_{|X|,K-1}$*, we have* $\phi_X^{(r)}(P) = \frac{1}{c}\phi_{X_c}^{(r)}(P^c)$*, where* $P^c \in \Delta_{c\cdot|X|,K-1}$ *contains* c *copies of each assignment from* P *(ordered in the same way as the copies in* X_c *are ordered).*

Moreover, for all means $C \subset \mathbb{R}^D$*,* $\phi_X^{(r)}(C) = \frac{1}{c}\phi_{X_c}^{(r)}(C)$*.*

Proof. From Corollary 2.26 it follows that the mean vectors induced by P with respect to X coincide with the mean vectors induced by P^c with respect to X_c. This yields the first claim. To see that the second claim holds true, recall Lemma 5.17. □

Second, we need to be able to compute an appropriate number of copies c. However, as said before, the weights of arbitrary r-fuzzy clusters might be arbitrarily small. Hence, the number of copies that we have to add might be arbitrarily large. Therefore, we need an additional trick.

We make use of our notion of non-negligible clusters. Recall from Section 6.2 that for each soft clustering, there exists a soft clustering that has similar cost and no negligible clusters. Each non-negligible cluster has a certain minimum support and, hence, the corresponding r-fuzzy clusters have a certain minimum weight. Observe that neither the required absolute bound nor the minimum weight of a non-negligible cluster depend on the number of data points. Moreover, we know both bounds. These observations lead us to the following result:

Corollary 8.8 (add copies to increase the weights)**.** *Let $K \in \mathbb{N}$, and let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an $\mathbf{i}_r$-increase-bounded fuzzifier function. Consider some data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ and a soft K-clustering $P \in \Delta_{|X|,K-1}$. Choose some constant $a \in \mathbb{R}_{>0}$.*

Set

$$c := \left\lceil a \cdot \left(r\left(\frac{\epsilon}{2\mathbf{i}_r K^2} \right) \cdot \mathtt{w}_{\min}^{(X)} \right)^{-1} \right\rceil .$$

Let X_c be the data set that contains c copies of each data point from X and let $P^c \in \Delta_{c\cdot|X|,K-1}$ be the soft K-clustering that contains c copies of each assignment from P (ordered in the same way as the copies of X in X_c).

Then, there exists a soft L-clustering P^c_{sup} of X_c with $L \leq K$ such that

$$\phi^{(r)}_{X_c}(P^c_{sup}) \leq (1+\epsilon) \cdot \phi^{(r)}_{X_c}(P^c)$$

and

$$\mathbf{w}\left(A_k^{(X_c, r(P^c_{sup}))} \right) \geq a \ .$$

Proof. Write $X = ((x_n, w_n))_{n\in[N]}$. From Theorem 6.3, we know that there exists a soft L-clustering $P_{sup} = (p_{nl})_{n,l}$ of X with $L \leq K$ such that

$$\phi^{(r)}_X(P_{sup}) \leq (1+\epsilon) \cdot \phi^{(r)}_X(P) \tag{8.8}$$

and such that $\forall l \in [L]\ \exists n \in [N] :\ p_{nl} \geq \frac{\epsilon}{2\mathbf{i}_r K^2}$. From the latter property and the properties of a fuzzifier function, it follows that

$$\forall l \in [L] :\ \mathbf{w}\left(A_l^{(X, r(P_{sup}))} \right) = \sum_{n=1}^{N} w_n r(p_{nk}) \geq r\left(\frac{\epsilon}{2\mathbf{i}_r K^2} \right) \cdot \mathtt{w}_{\min}^{(X)} \ . \tag{8.9}$$

Let $P^c_{sup} \in \Delta_{c\cdot|X|,K-1}$ be the soft K-clustering that contains c copies of each assignment from P_{sup} (ordered in the same way as the copies of X are ordered in X_c). From Corollary 8.7 and (8.8), we can conclude that $\phi^{(r)}_{X_c}(P^c_{sup}) \leq (1+\epsilon) \cdot \phi^{(r)}_{X_c}(P^c)$. From Corollary 2.26 and (8.9), we can conclude that

$$\mathbf{w}\left(A_l^{(X_c, r(P^c_{sup}))} \right) = c \cdot \mathbf{w}\left(A_l^{(X, r(P_{sup}))} \right) \geq c \cdot r\left(\frac{\epsilon}{2\mathbf{i}_r K^2} \right) \cdot \mathtt{w}_{\min}^{(X)} \geq a$$

for all $l \in [L]$. This yields the claim. □

We point out that the required absolute weight and the minimum weight of a non-negligible cluster depend on the minimum and maximum weight of a data point in the given data set, respectively. Therefore, we cannot remove the restriction by scaling the weights.

8.6 Algorithms

In this section, we state and discuss two algorithms: First, we present a deterministic $(1+\epsilon)$-approximation algorithm that applies the de-randomized version of superset sampling from Theorem 8.1. Second, we describe a randomized $(1+\epsilon)$-approximation algorithm that uses superset sampling directly (Theorem 8.3). For both of these algorithms, the proof of their correctness relies on our soft-to-hard clustering technique.

8.6.1 A Deterministic Approximation Algorithm (Algorithm 8)

Let us start with an application of the de-randomized version of superset sampling from Theorem 8.1. Before we consider a concrete fuzzifier function, take note of the result in its most general form:

Algorithm 8 Deterministic Approximation via Superset Sampling

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{i}_r \in [1,\infty)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$

1: Choose

$$\tilde{\epsilon} := \epsilon/6 \quad \text{and} \quad B := \left\lceil \left(\tilde{\epsilon} \cdot \mathtt{w}_{\min}^{(X)}\right)^{-1} \right\rceil .$$

2: Choose the number of copies

$$c := \left\lceil \left(\frac{32}{\tilde{\epsilon}} K(1+\tilde{\epsilon}) \mathtt{w}_{\max}^{(X)}\right) \cdot \left(r\left(\frac{\tilde{\epsilon}}{2\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X)}\right)^{-1} \right\rceil .$$

3: Construct a data set $X_c^{\mathbb{Q}}$ that, for each $n \in [N]$, contains c copies of the data point $(x_n, c \cdot w_n^{\mathbb{Q}})$, where $w_n^{\mathbb{Q}} := \lceil w_n \cdot B \rceil / B$.

4: Apply Algorithm 7 to $X_c^{\mathbb{Q}}$, K, $\tilde{\epsilon}/(32K)$ to compute a set of candidate solutions $T \subset \left(\mathbb{R}^D\right)^K$

5: Determine $C \in \arg\min \left\{ \phi_X^{(r)}(C') \,\middle|\, C' \in T \right\}$.

6: **return** C

Theorem 8.9. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, an $\mathbf{i}_r$-increase-bounded $[0,1]$-reducing fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the value $\mathbf{i}_r \in [1,\infty)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, Algorithm 8 computes means $C \subset \mathbb{R}^D$, $|C| \leq K$, such that*

$$\phi_X^{(r)}(C) \leq (1+\epsilon) \cdot \phi_{(X,K,m)}^{OPT} .$$

The algorithms' runtime is

$$D \cdot \mathbf{t}_r(K) \cdot \left(r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \cdot |X|\right)^{\mathcal{O}\left(K^2 \log(K)/\epsilon \log(1/\epsilon)\right)} .$$

Proof. First, we show that this algorithm returns a solution C with the desired approximation factor. Fix an arbitrary soft K-clustering P_{opt} of X and K means C_{opt} with

$$\phi_X^{(r)}(C_{opt}, P_{opt}) = \phi_{(X,K,r)}^{OPT} . \tag{8.10}$$

Let $X^{\mathbb{Q}} := \left((x_n, w_n^{\mathbb{Q}})\right)_{n\in[N]}$ where $w_n^{\mathbb{Q}} := \lceil w_n \cdot B \rceil / B$ for each $n \in [N]$. Then, from Lemma 8.6, we know

$$\mathtt{w}_{\min}^{(X^{\mathbb{Q}})} \geq \mathtt{w}_{\min}^{(X)} \quad \text{and} \tag{8.11}$$

$$\mathtt{w}_{\max}^{(X^{\mathbb{Q}})} \leq (1+\tilde{\epsilon}) \mathtt{w}_{\max}^{(X)} . \tag{8.12}$$

Let $X_c^{\mathbb{Q}}$ be defined as in the algorithm.

Let P_{opt}^c be the soft K-clustering of $X_c^{\mathbb{Q}}$ that contains c copies of each assignment from P_{opt} (ordered in the same way as the copies in $X_c^{\mathbb{Q}}$ are ordered). With (8.11), we can conclude that

$$c \geq \left\lceil \left(\frac{32}{\tilde{\epsilon}} K(1+\tilde{\epsilon}) \cdot \mathtt{w}_{\max}^{(X)}\right) \cdot \left(r\left(\frac{\tilde{\epsilon}}{2\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X^{\mathbb{Q}})}\right)^{-1} \right\rceil .$$

Hence, with Corollary 8.8, we can conclude that there exists a soft L-clustering P_{sup}^c of $X_c^{\mathbb{Q}}$ with $L \leq K$,

$$\phi_{X^{\mathbb{Q}}}^{(r)}(P_{sup}^c) \leq (1+\tilde{\epsilon}) \phi_{X^{\mathbb{Q}}}^{(r)}(P_{opt}^c) , \tag{8.13}$$

and

$$\begin{aligned}
\forall l \in [L]\colon \mathbf{w}\left(A_l^{(X_c^{\mathbb{Q}}, r(P_{sup}^c))}\right) &\geq \frac{32}{\tilde{\epsilon}} K(1+\tilde{\epsilon}) \mathtt{w}_{\max}^{(X)} \\
&\geq \frac{32}{\tilde{\epsilon}} K \mathtt{w}_{\max}^{(X^{\mathbb{Q}})} && \text{(Equation (8.12))} \\
&= \frac{32}{\tilde{\epsilon}} K \mathtt{w}_{\max}^{(X_c^{\mathbb{Q}})} && \text{(Corollary 2.26)} \\
&\geq \frac{32}{\tilde{\epsilon}} L \mathtt{w}_{\max}^{(X_c^{\mathbb{Q}})} . && (L \leq K)
\end{aligned}$$

Due to the latter bound and Theorem 8.1, there exist hard clusters $A_1, \ldots, A_L \subseteq X_c^{\mathbb{Q}}$ where, for each $l \in [L]$, we have

$$\begin{aligned}
\mathbf{w}(A_l) &\geq \frac{1}{2}\mathbf{w}\left(A_l^{(X_c^{\mathbb{Q}}, r(P_{sup}^c))}\right) , \\
\left\| \mathbf{m}(A_l) - \mathbf{m}\left(A_l^{(X_c^{\mathbb{Q}}, r(P_{sup}^c))}\right) \right\|_2^2 &\leq \frac{\tilde{\epsilon}}{4} \cdot \mathbf{var}\left(A_l^{(X_c^{\mathbb{Q}}, r(P_{sup}^c))}\right) \text{ , and} \\
\mathbf{d}(A_l) &\leq 4L \cdot \mathbf{d}\left(A_l^{(X_c^{\mathbb{Q}}, r(P_{sup}^c))}\right) .
\end{aligned}$$

From Corollary 8.4, we know that the set T contains a candidate $C = (\mu_k)_{k\in[K]} \in T$ with

$$\forall l \in [L]\colon \left\|\mu_l - \mathbf{m}(A_l)\right\|_2^2 \leq \frac{\tilde{\epsilon}}{32K}\mathbf{var}(A_l) \leq \frac{\tilde{\epsilon}}{32L}\mathbf{var}(A_l) .$$

Combining these results via Lemma 8.5 gives

$$\phi_{X_c^{\mathbb{Q}}}^{(r)}(C) \leq (1+\tilde{\epsilon})\phi_{X_c^{\mathbb{Q}}}^{(r)}(P_{sup}^c) . \tag{8.14}$$

Let P be the r-fuzzy L-clustering of $X^{\mathbb{Q}}$ induced by C. Then, observe that

$$\begin{aligned}
\phi_X^{(r)}(C) &\leq \phi_X^{(r)}(C,P) \\
&\leq \phi_{X^{\mathbb{Q}}}^{(r)}(C,P) && \text{(Lemma 8.6)} \\
&= \phi_{X^{\mathbb{Q}}}^{(r)}(C) && \text{(by definition of } P) \\
&= \frac{1}{c}\phi_{X_c^{\mathbb{Q}}}^{(r)}(C) && \text{(Corollary 8.7)} \\
&\leq (1+\tilde{\epsilon}) \cdot \frac{1}{c}\phi_{X_c^{\mathbb{Q}}}^{(r)}(P_{sup}^c) && \text{(Equation (8.14))} \\
&\leq (1+\tilde{\epsilon})^2 \cdot \frac{1}{c}\phi_{X_c^{\mathbb{Q}}}^{(r)}(P_{opt}^c) && \text{(Equation (8.13))} \\
&\leq (1+\tilde{\epsilon})^2 \cdot \phi_{X^{\mathbb{Q}}}^{(r)}(P_{opt}) && \text{(Corollary 8.7)} \\
&\leq (1+\tilde{\epsilon})^2 \cdot \phi_{X^{\mathbb{Q}}}^{(r)}(C_{opt}, P_{opt}) && \text{(cf. Equation (8.10))} \\
&\leq (1+\tilde{\epsilon})^3 \cdot \phi_X^{(r)}(C_{opt}, P_{opt}) && \text{(Lemma 8.6)} \\
&= (1+\tilde{\epsilon})^3 \cdot \phi_{(X,K,r)}^{OPT} && \text{(Equation (8.10))} \\
&\leq (1+6\tilde{\epsilon}) \cdot \phi_{(X,K,r)}^{OPT} && \text{(Lemma A.1)} \\
&= (1+\epsilon) \cdot \phi_{(X,K,r)}^{OPT} . && (\tilde{\epsilon} = \epsilon/6)
\end{aligned}$$

Since Algorithm 8 computes the cost of each $C' \in T$ and returns the C' with the smallest r-fuzzy K-means cost, this yields our claim.

Second, we analyse the algorithm's runtime. By definition, we have $\left|X_c^{\mathbb{Q}}\right| = c \cdot |X|$, where

$$c \leq (32 \cdot 2 \cdot 6) \cdot K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} .$$

$X_c^{\mathbb{Q}}$ can be constructed in $\mathcal{O}(c\,|X|D)$. We apply Algorithm 7 to $X_c^{\mathbb{Q}}$ to compute a set T, where we use $\tilde{\epsilon}/(32K)$ instead of ϵ. Hence, due to Corollary 8.4, the size of T is

$$|T| \in (c\cdot|X|)^{\mathcal{O}(K/(\tilde{\epsilon}/32K))} \subseteq (c\cdot|X|)^{\mathcal{O}(K^2/\epsilon)} \ .$$

and the time needed to construct T is $D\cdot(c\cdot|X|)^{\mathcal{O}(K^2/\epsilon)}$. Observe that

$$\begin{aligned}(c\cdot|X|)^{\mathcal{O}(K^2/\epsilon)} &\subseteq \left(K\cdot\frac{1}{\epsilon}\cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1}\cdot\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\cdot|X|\right)^{\mathcal{O}(K^2/\epsilon)}\\ &\subseteq \left(r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1}\cdot\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\cdot|X|\right)^{\mathcal{O}(K^2\log(K)/\epsilon\log(1/\epsilon))} \ .\end{aligned}$$

Finally, note that we have to evaluate $\phi_X^{(r)}(C)$ for each candidate $C\in T$. This needs time

$$\mathcal{O}\left(|T|\cdot(|X|\cdot D\cdot\mathbf{t}_r(K)+|X|KD)\right)\subseteq\mathcal{O}\left(|T|\cdot|X|\cdot D\cdot\mathbf{t}_r(K)\right) \ ,$$

since we assume that $\mathbf{t}_r(K)\in\Omega(K)$ (Assumption 5.19). This yields the claim. □

Observe that Algorithm 9 is a polynomial-time approximation scheme for the r-fuzzy K-means problem if $K,\mathbf{i}_r\in\mathcal{O}(1)$ are constants and if the given data sets X satisfy $\mathtt{w}_{\max}^{(X)}/\mathtt{w}_{\min}^{(X)}\in|X|^{\mathcal{O}(1)}$. Note that this covers all unweighted data sets $X\in\mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$ and all the fuzzifier functions presented in Section 5.3, except for the exponential fuzzifier function $\mathtt{e}_\gamma$, which is not increase-bounded.

Special Case. Consider the classical fuzzy K-means problem with the polynomial fuzzifier function $\mathtt{p}_m(x)=x^m$ with $m\in(1,\infty)$. Recall from Section 5.3.2, that we can set $\mathbf{t}_{\mathtt{p}_m}(K)=\mathcal{O}(K)$ and $\mathbf{i}_{\mathtt{p}_m}=4m$. Hence,

$$\mathtt{p}_m\left(\frac{\epsilon}{12\cdot\mathbf{i}_{\mathtt{p}_m}\cdot K^2}\right)^{-1}=\left(\frac{48\cdot m\cdot K^2}{\epsilon}\right)^m \ .$$

This means that for the classical fuzzy K-means problem, Theorem 8.9 describes a $(1+\epsilon)$-approximation algorithm with runtime

$$D\cdot K\cdot\left(\left(\frac{48\cdot m\cdot K^2}{\epsilon}\right)^m\cdot\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}|X|\right)^{\mathcal{O}(K^2\log(K)/\epsilon\log(1/\epsilon))}\subseteq D\cdot\left(\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}|X|\right)^{\mathcal{O}(K^3\epsilon^{-2}m^2)} \ .$$

To sum up, if $K\in\mathbb{N}$ and $m\in(1,\infty)$ are constants and if we are only given data sets where $\mathtt{w}_{\max}^{(X)}/\mathtt{w}_{\min}^{(X)}\in|X|^{\mathcal{O}(1)}$, then we have a polynomial-time approximation scheme for the classical fuzzy K-means problem.

8.6.2 A Randomized Algorithm (Algorithm 9)

Algorithm 7 applies a de-randomized version of the superset-sampling technique. As already explained in Section 8.4, this eliminates the need to fix a ratio α between the weights of the r-fuzzy clusters, whose means we want to approximate, and the weight of the given data set. If we knew the ratio α_{opt} between the weights of optimal r-fuzzy clusters and the weight of the given data set in advance, then we could apply the randomized version of sampling-sampling in a way that guarantees that we approximate the means of optimal r-fuzzy clusters. However, we do not know α_{opt}.

In the following, we first consider the application of the superset sampling technique with some arbitrary but fixed ratio $\alpha\in(0,1]$. We explain why, with respect to an arbitrary but fixed constant ratio α, the use of this algorithm is questionable. However, we also show that, for each data set, we can choose a specific α that works nearly as good as the optimal ratio α_{opt}, which we do not know. To this end, we make use of our notion of negligible clusters from Section 6.2.

A Questionable Algorithm. The following algorithm does not approximate an optimal solution. Its goal is to find a solution whose cost is not much worse than the minimum cost of a solution where the ratio between the weight of each r-fuzzy cluster and the data set is at least α.

Algorithm 9 Randomized Approximation via Superset Sampling

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{i}_r \in [1,\infty)$, $K \in \mathbb{N}$, and $\epsilon, \alpha \in (0,1]$

1: Choose
$$\tilde{\epsilon} := \epsilon/6 \quad \text{and} \quad B := \left\lceil \left(\tilde{\epsilon} \cdot \mathtt{w}_{\min}^{(X)}\right)^{-1} \right\rceil .$$
2: Choose the number of copies
$$c := \left\lceil \left(32K\tilde{\epsilon}^{-1}(1+\tilde{\epsilon}) \cdot \mathtt{w}_{\max}^{(X)}\right) \cdot \left(r\left(\frac{\tilde{\epsilon}}{2\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X)}\right)^{-1} \right\rceil .$$
3: Construct a data set $X_c^{\mathbb{Q}}$ that, for each $n \in [N]$, contains c copies of the data point $(x_n, c \cdot w_n^{\mathbb{Q}})$, where $w_n^{\mathbb{Q}} := \lceil w_n \cdot B \rceil / B$.
4: Apply Algorithm 6 to $X_c^{\mathbb{Q}}$, K, $\tilde{\epsilon}/(32K)$ to compute a set of candidate solutions $T \subset \left(\mathbb{R}^D\right)^K$.
5: Determine $C \in \arg\min\left\{\phi_X^{(r)}(C') \,\middle|\, C' \in T\right\}$.
6: **return** C

Theorem 8.10. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, an $\mathbf{i}_r$-increase-bounded $[0,1]$-reducing fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the value $\mathbf{i}_r \in [1,\infty)$, $\epsilon \in (0,1]$, and $\alpha \in [0,1]$, Algorithm 9 computes means $C \subset \mathbb{R}^D$, $|C| \leq K$, such that with constant probability*

$$\phi_X^{(r)}(C) \leq (1+\epsilon)\phi_X^{(r)}(P_\alpha) ,$$

where

$$P_\alpha \in \arg\min\left\{\phi_X^{(r)}(P) \,\middle|\, \begin{array}{c} \text{soft } L\text{-clustering } P \text{ of } X \text{ with } L \leq K \text{ and} \\ \text{where } \forall l \in [L] : \mathbf{w}\left(A_l^{(X,r(P))}\right) \geq \alpha \cdot \mathbf{w}(X) \end{array}\right\} ,$$

if this minimum exists.

The algorithms' runtime is

$$|X| \cdot \left(\mathcal{O}\left(K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right) + D \cdot \mathbf{t}_r(K) \cdot 2^{\mathcal{O}\left(K \log(K)^2 \log(1/(\alpha\epsilon))\right)}\right)$$

Proof. Observe that Algorithm 9 differs from Algorithm 8 only in the way that superset sampling is applied: Algorithm 8 uses Algorithm 7 to compute a set of candidate solutions T. Algorithm 6, as used by Algorithm 8, computes a set of candidate solutions T with the same property regarding the hard clusters $A_1, \ldots, A_K$, with a constant success probability (cf. Corollary 8.4 and Theorem 8.3). Hence, Algorithm 9 computes a $(1+\epsilon)$-approximation, with constant probability.

As for the runtime, recall that by definition $\left|X_c^{\mathbb{Q}}\right| = c \cdot |X|$. Hence, $X_c^{\mathbb{Q}}$ can be constructed in $\mathcal{O}(c\,|X|\,D)$. We apply Algorithm 6 to $X_c^{\mathbb{Q}}$ with $\tilde{\epsilon}/(32K)$ instead of ϵ. According to Theorem 8.3, this needs time $(c\,|X|) + |T| \cdot D)$ where

$$|T| \in 2^{\mathcal{O}(K \log\log(K) \log(1/(\alpha\tilde{\epsilon})))} \subseteq 2^{\mathcal{O}\left(K \log(K)^2 \log(1/(\alpha\epsilon))\right)}$$

and

$$c \in \mathcal{O}\left(K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right) .$$

Evaluating $\phi_X^{(r)}(C)$ for each candidate $C \in T$ needs time $\mathcal{O}(|T| \cdot |X| \cdot D \cdot \mathbf{t}_r(K))$ (cf. Assumption 5.19). By putting all these bounds together, we can bound the overall runtime by

$$\begin{aligned}
&\mathcal{O}(c\,|X| + |T| \cdot |X| \cdot D \cdot \mathbf{t}_r(K)) \\
=&\mathcal{O}(|X| \cdot (c + |T| \cdot D \cdot \mathbf{t}_r(K))) \\
=&|X| \cdot \left(\mathcal{O}\left(K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2} \right)^{-1} \cdot \frac{\mathbf{w}_{\max}^{(X)}}{\mathbf{w}_{\min}^{(X)}} \right) + D \cdot \mathbf{t}_r(K) \cdot 2^{\mathcal{O}(K \log(K)^2 \log(1/(\alpha\epsilon)))} \right)
\end{aligned}$$

This yields the claim. □

Two Flaws. Theorem 8.10 seems to be a constraint clustering approach (Basu et al., 2008). However, there are two flaws: First, the constraint only applies to the solution that the outcome of the algorithm is compared to. The outcome of the algorithm is *not* guaranteed to satisfy the same constraint as well. Second, the constraint is difficult to interpret. The sum of all the weights of all r-fuzzy clusters in a fuzzy K-means clustering, in general, does not sum up to the weight of the data set. Therefore, the constraint

$$\forall l \in [L] : \mathbf{w}\left(A_l^{(X, r(P))} \right) \geq \alpha \cdot \mathbf{w}(X)$$

is potentially much stronger than the constraint that the clusters are *balanced* in the sense that

$$\forall l \in [L] : \mathbf{w}\left(A_l^{(X, r(P))} \right) \geq \alpha \cdot \sum_{l=1}^{L} \mathbf{w}\left(A_l^{(X, r(P))} \right) .$$

This problem becomes very obvious in the following example:

Example 8.11 (an artificial and extreme case)**.** *For the classical fuzzy K-means problem with fuzzifier $m \in (1, \infty)$ the gap between these restrictions might be up to a factor $\mathbf{c}^*_{\mathrm{p}_m} = K^{m-1}$: Consider the "uniform" soft K-clustering P_u with $p_{nk} = 1/K$ for all $n \in [N]$ and $k \in [K]$. The p_m-fuzzy clusters described by P_u are perfectly balanced in the sense that*

$$\forall k \in [K] : \mathbf{w}\left(A_k^{(X, \mathrm{p}_m(P_u))} \right) = \sum_{n=1}^{N} \left(\frac{1}{K} \right)^m w_n = \frac{1}{K^m} \cdot \mathbf{w}(X) .$$

However, when we apply Theorem 8.10 with $\alpha = 1/K^2$ and $m = 3$, then we know that the outcome of the algorithm is compared to a soft clustering P_α that is chosen from a set that does not *contain the perfectly balanced uniform soft K-clustering P_u.*

That is, the parameter α does not describe how balanced the (r-fuzzy) clusters of the sought solution are. To sum up, we are still missing a concise interpretation of the constraint from Theorem 8.10.

A Reasonable Application of a Questionable Algorithm. From Theorem 6.3, we know that there exists a $(1+\epsilon)$-approximation to the r-fuzzy K-means problem where each r-fuzzy cluster has a certain weight. This fact directly leads us to the following result.

Corollary 8.12. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, an $\mathbf{i}_r$-increase-bounded $[0,1]$-reducing fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the value $\mathbf{i}_r \in [1, \infty)$, $\epsilon/4 \in (0, 1/4]$, and $\alpha := r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right)^{\frac{\mathbf{w}_{\min}^{(X)}}{\mathbf{w}_{\max}^{(X)} \cdot |X|}}$, Algorithm 9 computes means $C \subset \mathbb{R}^D$, $|C| \leq K$, such that with constant probability*

$$\phi_X^{(r)}(C) \leq (1+\epsilon)\phi_{(X,K,r)}^{OPT} .$$

The algorithms' runtime is

$$D \cdot \mathbf{t}_r(K) \cdot \left(r\left(\frac{\epsilon}{8\mathbf{i}_r K^2} \right)^{-1} \frac{\mathbf{w}_{\max}^{(X)}}{\mathbf{w}_{\min}^{(X)}} \cdot |X| \right)^{\mathcal{O}(K \log(K)^2 \cdot \log(1/\epsilon))} .$$

Proof. From Theorem 6.3, we know that there exists a $C \subseteq \mathbb{R}^D$ with $|C| =: L \leq K$ and $\phi_X^{(r)}(C) \leq (1+\epsilon/4) \cdot \phi_{(X,K,r)}^{OPT}$ that induces some soft L-clustering $P = (p_{nk})_{n,k}$ of X where, for all $l \in [L]$, there exists some $n \in [N]$ with $p_{nl} \geq \frac{\epsilon}{8\mathbf{i}_r K^2}$. Observe that $\phi_X^{(r)}(P) \leq \phi_X^{(r)}(C,P)$.

Moreover, we can conclude that, for all $l \in [L]$, we have $\mathbf{w}\left(A_k^{(X,P)}\right) \geq r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X)}$. Let $\alpha := r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right) \frac{\mathtt{w}_{\min}^{(X)}}{\mathtt{w}_{\max}^{(X)} \cdot |X|}$. Then,

$$\alpha \cdot \mathbf{w}(X) = r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right) \frac{\mathtt{w}_{\min}^{(X)}}{\mathtt{w}_{\max}^{(X)} \cdot |X|} \cdot \mathbf{w}(X) \leq r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right) \frac{\mathtt{w}_{\min}^{(X)}}{\mathbf{w}(X)} \cdot \mathbf{w}(X) = r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right) \mathtt{w}_{\min}^{(X)} .$$

That is, $\alpha \cdot \mathbf{w}(X)$ is a lower bound on the weights of the r-fuzzy clusters of X defined by P.

Hence, by applying Algorithm 8 as described in the claim, we compute (with constant probability) a solution whose cost are at most a factor $(1+\epsilon/4)$ larger than the cost of the solution induced by P, which is at most a factor $(1+\epsilon/4)$ larger than an optimal cost. So, overall, we compute a solution whose cost is at most a factor $(1+\epsilon/4)^2 \leq (1+\epsilon)$ worse than an optimal solution, with constant probability.

Observe that our execution of the algorithm needs time

$$\begin{aligned}
&|X| \cdot \left(\mathcal{O}\left(K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \right) + D \cdot \mathbf{t}_r(K) \cdot 2^{\mathcal{O}\left(K \log(K)^2 \cdot \log\left(r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right)^{-1} \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \cdot |X|\right) \cdot \log(1/\epsilon)\right)} \right) \\
\subseteq &|X| \cdot \left(\mathcal{O}\left(K \cdot \frac{1}{\epsilon} \cdot r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \right) + D \cdot \mathbf{t}_r(K) \cdot \left(r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right)^{-1} \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \cdot |X| \right)^{\mathcal{O}(K \log(K)^2 \cdot \log(1/\epsilon))} \right) \\
\subseteq &D \cdot \mathbf{t}_r(K) \cdot \left(r\left(\frac{\epsilon}{8\mathbf{i}_r K^2}\right)^{-1} \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \cdot |X| \right)^{\mathcal{O}(K \log(K)^2 \cdot \log(1/\epsilon))}
\end{aligned}$$

□

Compared to the runtime of the deterministic algorithm from Algorithm 8, which is bounded by

$$D \cdot \mathbf{t}_r(K) \cdot \left(r\left(\frac{\epsilon}{12\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} \cdot |X| \right)^{\mathcal{O}(K^2 \log(K)/\epsilon \log(1/\epsilon))} ,$$

the runtime of the randomized algorithm from Corollary 8.12 has only a slightly better dependence on K.

Special Case. Again, consider the classical fuzzy K-means problem with the polynomial fuzzifier function $\mathrm{p}_m(x) = x^m$ with $m \in (1,\infty)$. Analogously to the previous section,

$$\mathrm{p}_m\left(\frac{\epsilon}{8 \cdot \mathbf{i}_{\mathrm{p}_m} \cdot K^2}\right)^{-1} = \left(\frac{32 \cdot m \cdot K^2}{\epsilon}\right)^m .$$

Hence, for the classical fuzzy K-means problem, Corollary 8.12 describes a *randomized* $(1+\epsilon)$-approximation algorithm with runtime

$$D \cdot K \cdot \left(\left(\frac{32 \cdot m \cdot K^2}{\epsilon}\right)^m \cdot \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} |X| \right)^{\mathcal{O}(K \log(K)^2/\epsilon \log(1/\epsilon))} \subseteq D \cdot \left(\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} |X| \right)^{\mathcal{O}(K^2 \epsilon^{-2} m^2)} .$$

"One Ring to rule them all, one Ring to find them."

J. R. R. Tolkien

Chapter 9

A Discretization

Chen (2009) described a construction for the K-means problem that can be thought of as a well-informed grid search. In Chapter 7, we considered a grid search for the soft assignments of a good solution. Now we consider a grid search for good mean vectors. A naïve approach to such a grid search would be to construct a D-dimensional grid where the grid cells have some fixed side length and search through all grid points that lie within the convex hull of the point set. With some additional information and the use of exponential grids, Chen (2009) improved this naïve approach substantially. His approach uses a coarse K-means solution as additional information. That is, one pre-computes a constant-factor approximation of the K-means problem. This solution gives a rough idea of where the data points lie. In particular, one knows that the data points are rather concentrated around the means of the coarse K-means solution. This information is then exploited by constructing a special kind of exponential grid around the means. Chen (2009) used this construction to develop a coreset and a $(1+\epsilon)$-approximation algorithm for the K-means problem.

In Section 6.1 we showed that there is a coarse relation between the K-means and the r-fuzzy K-means cost function, given that the fuzzifier function r is contribution-bounded. This coarse relation is the key ingredient that enables us to interpret this construction with respect to the r-fuzzy K-means problem.

In this chapter, we focus on analysing the basic properties of the construction. In the subsequent chapters we will then apply the results in two different ways: In Chapter 10, we use it to construct a $(1+\epsilon)$-approximation algorithm for the r-fuzzy K-means problem. In Chapter 12, we show that it helps to construct a coreset for the r-fuzzy K-means problem.

Overview. In Section 9.1 we summarize our contribution. After a brief introduction of some notation in Section 9.2, we describe the construction formally in Section 9.3. In particular, we define a search space $\mathfrak{U} \subseteq \mathbb{R}^D$ and a representative $\mathfrak{g}(x)$ for each point $x \in \mathfrak{U}$ in the search space. In Section 9.4, we focus on the properties of the search space $\mathfrak{U}$ and distances between points $x \in \mathfrak{U}$ and their representatives $\mathfrak{g}(x)$. In Section 9.5, we have a closer look at the size of the set $\mathfrak{g}(\mathfrak{U})$, which contains all representatives, and the time needed to compute this set.

Publication. In this chapter, we generalize results that have been published in Blömer et al. (2017).

9.1 Contribution

We give a detailed analysis of the construction described by Chen (2009): On the one hand, we identify the single properties of this construction that have already been exploited for the K-means problem. On the other hand, we analyse these properties with respect to the

r-fuzzy K-means problem. The benefit is twofold: First, this allows for a discussion of the usefulness of this approach. It becomes clear that this construction is probably only useful (and our analysis is only useful) if the coarse K-means solution and the fuzzifier function meet certain requirements. Second, apart from deriving a coreset construction similar to Chen (2009) (see Chapter 12), from our analysis it becomes clear that the construction can be used to derive a $(1+\epsilon)$-approximation algorithm for the r-fuzzy K-means problem, as we show in Chapter 10.

9.2 Preliminaries

From now on, we use the following abbreviations:

Definition 9.1 (distance)**.** *For each $x \in \mathbb{R}^D$ and $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$, we let*

$$\mathtt{dist}(x, C) := \min\left\{ \left\| x - \mu_k \right\|_2 \;\middle|\; k \in [K] \right\} .$$

Definition 9.2 (ball)**.** *For each $\mu \in \mathbb{R}^D$ and $r \in \mathbb{R}_+$, we let*

$$\mathrm{B}(\mu, r) := \left\{ x \in \mathbb{R}^D \;\middle|\; \left\| x - \mu \right\|_2 \leq r \right\}$$

be the closed ball around μ with radius r.

Definition 9.3. *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, and $\alpha, \beta \in [1, \infty)$. We call $\mathfrak{M} \subset \mathbb{R}^D$ an (α, β)-approximation to the K-means problem with respect to X if*

$$\mathrm{km}_X(\mathfrak{M}) \leq \alpha \cdot \mathrm{km}_{(X,K)}^{OPT} \quad \text{and} \quad |\mathfrak{M}| = \lfloor \beta \cdot K \rfloor .$$

Moreover, the following simple lemma will be useful.

Lemma 9.4. *For all $a, b, c \in R^D$ we have*

$$\|a - c\|_2^2 \leq \|a - b\|_2^2 + \|b - c\|_2^2 + 2\|a - b\|_2 \|c - b\|_2 .$$

Proof. For all $a, b, c \in R^D$, we have

$$\begin{aligned}
\|a - c\|_2^2 &\leq (\|a - b\|_2 + \|b - c\|_2)^2 && \text{(Lemma A.3)} \\
&= \|a - b\|_2^2 + 2\langle a - b,\ b - c\rangle + \|b - c\|_2^2 \\
&\leq \|a - b\|_2^2 + 2|\langle a - b,\ b - c\rangle| + \|b - c\|_2^2 \\
&\leq \|a - b\|_2^2 + 2\|a - b\|_2 \|b - c\|_2 + \|b - c\|_2^2 .
\end{aligned}$$

where the last inequality is due to the Cauchy-Schwarz inequality. □

9.3 Basic Construction

The following construction is due to (Chen, 2009, pp. 935): Consider a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$. Assume we are given L mean vectors $\mathfrak{M} = (\mathfrak{m}_l)_{l\in[L]} \subseteq \mathbb{R}^D$ that give us a rough idea of where the data points from X lie. For now, we do not assume any specific property of $\mathfrak{M}$. We will deal with the question of how to choose $\mathfrak{M}$ later on in Section 9.4.5.

In the first step of the construction, we restrict the search space to the union of closed balls around the single means.

Definition 9.5 (search space)**.** *Let $\mathfrak{Descr} := \mathbb{N}_0 \times \mathbb{R}_+ \times \left\{ (\mathfrak{m}_l)_{l\in[L]} \subseteq \mathbb{R}^D \;\middle|\; L, D \in \mathbb{N} \right\}$.*

For each $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$ with $\mathfrak{M} \subseteq \mathbb{R}^D$, we call the set

$$\mathfrak{U}(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) := \bigcup_{\mathfrak{m}\in\mathfrak{M}} \mathrm{B}\left(\mathfrak{m}, 2^{\mathfrak{E}} \cdot \mathfrak{R}\right) \subset \mathbb{R}^D . \tag{9.1}$$

the search space described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$.

We partition each ball into exponentially increasing rings.

Definition 9.6 (ring)**.** *Let* $(\mathfrak{E},\mathfrak{R},(\mathfrak{m}_l)_{l\in[L]}) \in \mathfrak{Descr}$*. Let* $l \in [L]$ *and* $j \in \{0,\dots,\mathfrak{E}\}$*. The* (l,j)*-th ring defined by* $(\mathfrak{E},\mathfrak{R},(\mathfrak{m}_l)_{l\in[L]})$ *is the set*

$$\mathfrak{U}_{l,j} \coloneqq \begin{cases} \mathrm{B}(\mathfrak{m}_l,\mathfrak{R}) \,, & \text{if } j = 0 \\ \mathrm{B}\big(\mathfrak{m}_l, 2^j\mathfrak{R}\big) \setminus \mathrm{B}\big(\mathfrak{m}_l, 2^{j-1}\mathfrak{R}\big) & \text{if } j \geq 1 \end{cases} . \tag{9.2}$$

The union of the rings $\mathfrak{U}_{l,j}$ defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ coincides with the search space $\mathfrak{U}$ defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$:

$$\bigcup_{k\in[L]} \bigcup_{j\in\{0,\dots,\mathfrak{E}\}} \mathfrak{U}_{l,j} = \mathfrak{U} \,.$$

For each $l \in [L]$, the rings $\mathfrak{U}_{l,j}$ with $j \in \{0,1,\dots,\mathfrak{E}\}$ are pairwise disjoint. However, this is not necessarily true for a fixed $j \in \{0,1,\dots,\mathfrak{E}\}$ and the rings $\mathfrak{U}_{l,j}$ with $l \in [L]$. Hence, the rings do *not* necessarily form a partition of $\mathfrak{U}(\mathfrak{E},\mathfrak{R},\mathfrak{M})$.

Finally, we put a grid over each ring. We let the sizes of the grid cells of these rings grow exponentially as well. We use these grid cells to define representative points as follows.

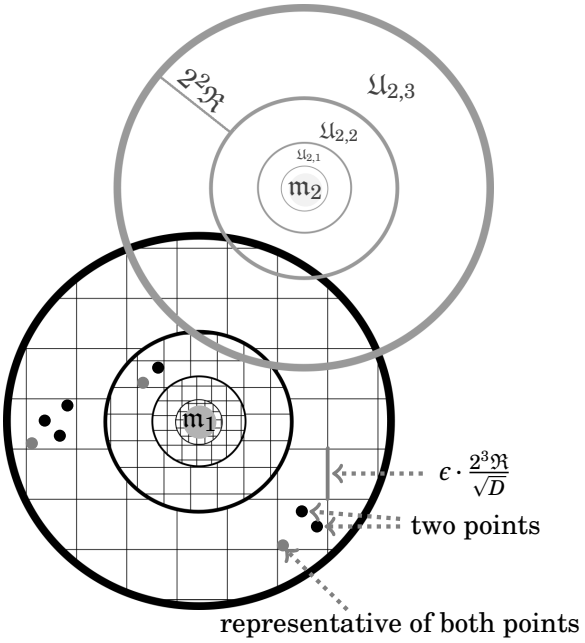


Figure 9.1: Sketch of the construction.

Definition 9.7 (representatives)**.** *Let* $\epsilon \in (0,1]$*,* $(\mathfrak{E},\mathfrak{R},\mathfrak{M}) \in \mathfrak{Descr}$*, and let* $\mathfrak{U}$ *be the search space described by* $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$*. A function* $\mathfrak{g} : \mathfrak{U} \to \mathfrak{U}$ *is a representative function if it satisfies the following property:*

For each $l \in [|\mathfrak{M}|]$ *and* $j \in \{0,\dots,\mathfrak{E}\}$*, let* $\mathfrak{U}_{l,j}$ *be the* (l,j)*-th ring defined by* $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$*. Assume that each ring* $\mathfrak{U}_{l,j} \subseteq \mathbb{R}^D$ *is partitioned into an axis-parallel grid with cells (i.e., hypercubes) of side length*

$$\epsilon \cdot \frac{2^j\mathfrak{R}}{\sqrt{D}} \,.$$

There is a function $\mathfrak{C}$ *that assigns each* $x \in \mathfrak{U}$ *to a grid cell* $\mathfrak{C}(x)$ *that contains* x *such that for all* $x,y \in \mathfrak{U}$ *with* $\mathfrak{C}(x) = \mathfrak{C}(y)$ *we have* $\mathfrak{g}(x) = \mathfrak{g}(y)$*.*

In simple words, a representative function $\mathfrak{g}$ maps all points in a grid cell to the same representative, which is also contained in the grid cell. However, it does not matter where the representative lies inside this cell.

Example 9.8 (representatives)**.** *First, assign each* $x \in \mathfrak{U}$ *to a grid cell* $\mathfrak{C}(x)$ *that contains* x *(i.e., choose a tie breaker for border points and overlapping cells). Let* $\mathbf{C} = \{\mathfrak{C}(x) \mid x \in \mathfrak{U}\}$*. For each* $\mathfrak{C} \in \mathbf{C}$ *choose an arbitrary point* $\mathfrak{c}(\mathfrak{C})$ *from* $\mathfrak{C}$ *(e.g. a vertex of the respective hypercube). Then, let* $\mathfrak{g}$ *be the function that maps each* $x \in \mathfrak{U}$ *to* $\mathfrak{c}(\mathfrak{C}(x))$*.*

We call the set that contains all possible representatives a discrete search space:

Definition 9.9 (discrete search space)**.** *Let* $\epsilon \in (0,1]$ *and* $(\mathfrak{E},\mathfrak{R},\mathfrak{M}) \in \mathfrak{Descr}$*. Let* $\mathfrak{U} \subset \mathbb{R}^D$ *be the search space described by* $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ *and let* $\mathfrak{g}$ *be a representative function defined by* $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ *and* ϵ*. The set*

$$\mathfrak{G} \coloneqq \mathfrak{g}(\mathfrak{U}) = \{\mathfrak{g}(x) \mid x \in \mathfrak{U}\} \subseteq \mathfrak{U}$$

is a discrete search space defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ *and* ϵ*.*

9.4 Distances and Costs

In this section, we focus on the properties of a search space $\mathfrak{U}$, its rings $\mathfrak{U}_{l,j}$, and representative points $\mathfrak{g}(x)$ of points $x \in \mathfrak{U}$ in the search space $\mathfrak{U}$.

9.4.1 Outside the Search Space

Consider some data set $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$. If we choose the radius $2^{\mathfrak{E}} \cdot \mathfrak{R}$ of each ball around each mean $\mathfrak{m} \in \mathfrak{M}$ large enough, then all the points x_n and some space around each point x_n are contained in the resulting search space $\mathfrak{U} = \mathfrak{U}(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$. It suffices to choose a radius $2^{\mathfrak{E}} \cdot \mathfrak{R} \propto \sqrt{\mathrm{km}_X(\mathfrak{M})/\mathtt{w}_{\min}^{(X)}}$.

Lemma 9.10. *Let $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $a, b \in \mathbb{R}$, $a, b \in [1, \infty)$, and $\epsilon \in (0, 1]$. Let $\mathfrak{U} \subset \mathbb{R}^D$ be the search space described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$, where*

$$\mathfrak{E} = \left\lfloor \frac{1}{2} \log\left(9ab \frac{\mathbf{w}(X)}{\mathtt{w}_{\min}^{(X)}}\right) \right\rfloor \quad \text{and} \quad \mathfrak{R} = \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{a\mathbf{w}(X)}} \,.$$

Then,

$$\bigcup_{x\in X} \mathtt{B}(x, \mathfrak{r}) \subseteq \mathfrak{U}$$

where

$$\mathfrak{r} = 2\sqrt{b \frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}}} \,.$$

Proof. Towards a contradiction, assume that there exists an $(x, w) \in X$ with $\mathtt{B}(x, r) \not\subseteq \mathfrak{U}$. Due to Definition 9.5, this implies that, for all $\mathfrak{m} \in \mathfrak{M}$, we have $\mathtt{B}(x, r) \not\subseteq \mathtt{B}\left(\mathfrak{m}, 2^{\mathfrak{E}}\mathfrak{R}\right)$. Hence,

$$\mathtt{dist}(x, \mathfrak{M}) > 2^{\mathfrak{E}}\mathfrak{R} - \mathfrak{r} \,.$$

Observe that

$$2^{\mathfrak{E}}\mathfrak{R} \leq \sqrt{9ab \frac{\mathbf{w}(X)}{\mathtt{w}_{\min}^{(X)}}} \cdot \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{a\mathbf{w}(X)}} = 3 \cdot \sqrt{b \frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}}} \,.$$

Hence,

$$\mathtt{dist}(x, \mathfrak{M}) > 2^{\mathfrak{E}}\mathfrak{R} - \mathfrak{r} = \sqrt{b \frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}}} \geq \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}}} \,. \qquad (b \geq 1)$$

To sum up, we have

$$\exists (x, w) \in X : \ \mathtt{w}_{\min}^{(X)} \mathtt{dist}(x, \mathfrak{M})^2 > \mathrm{km}_X(\mathfrak{M}) \ ,$$

which contradicts the fact that

$$\forall (x, w) \in X : \ \mathrm{km}_X(\mathfrak{M}) = \sum_{(x', w')\in X} w' \mathtt{dist}\left(x', \mathfrak{M}\right)^2 \geq \mathtt{w}_{\min}^{(X)} \mathtt{dist}(x, \mathfrak{M})^2 \,.$$

This yields the claim. □

9.4.2 Rings

The following simple yet useful observation corresponds to (a part of the proof of) Claim 5.3 from (Chen, 2009, p. 934). It exploits the fact that the radii of the rings are exponentially increasing.

Lemma 9.11 (diameter of a ring)**.** *Let* $(\mathfrak{E}, \mathfrak{R}, (\mathfrak{m}_l)_{l\in[L]}) \in \mathfrak{Descr}$*,* $\epsilon \in (0,1]$*,* $l \in [L]$*, and* $j \in \{0,1,\dots,\mathfrak{E}\}$*. Let* $\mathfrak{U}_{l,j}$ *be the* (l,j)*-th ring described by* $(\mathfrak{E}, \mathfrak{R}, (\mathfrak{m}_l)_{l\in[L]})$*. For all* $x \in \mathfrak{U}_{l,j}$*, we have*

$$\mathtt{diam}(\mathfrak{U}_{l,j}) \leq 2\max\left\{2\left\|x-\mathfrak{m}_l\right\|_2, \mathfrak{R}\right\} .$$

Proof. First, we show that $2^j\mathfrak{R} \leq \max\left\{2\left\|x-\mathfrak{m}_l\right\|_2, \mathfrak{R}\right\}$ for all $x \in \mathfrak{U}_{l,j}$. If $j=0$, then $2^0\mathfrak{R} = \mathfrak{R}$. If $j \geq 1$, then by definition we have $\left\|x-\mathfrak{m}_l\right\|_2 \geq 2^{j-1}\mathfrak{R}$ for all $x \in \mathfrak{U}_{l,j}$. Hence, $2\left\|x-\mathfrak{m}_l\right\|_2 \geq 2\cdot 2^{j-1}\mathfrak{R} = 2^j\mathfrak{R}$. This yields our initial claim.

Now consider some arbitrary but fixed $x, y \in \mathfrak{U}_{l,j}$. Due to the triangle inequality, we have $\left\|x-y\right\|_2 \leq \left\|x-\mathfrak{m}_l\right\|_2 + \left\|\mathfrak{m}_l-y\right\|_2 \leq 2\cdot 2^j\mathfrak{R}$. With our first claim, we can conclude that $\left\|x-y\right\|_2 \leq 2\max\left\{2\left\|x-\mathfrak{m}_l\right\|_2, \mathfrak{R}\right\}$. □

9.4.3 A Point and Its Representative

Recall that we put exponentially growing grids on the exponentially growing rings. The representative of a point in the search space is some fixed point inside the grid cell that contains the point. Hence, a point and its representative are always contained in the same grid cell and, needless to say, in the same ring.

Lemma 9.12 (distance between point and representative)**.** *Let* $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$ *and* $\epsilon \in (0,1]$*. Let* $\mathfrak{U} \subseteq \mathbb{R}^D$ *be the search space described by* $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ *and let* $\mathfrak{g}$ *be a representative function defined by* $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ *and* ϵ*.*

For all $x \in \mathfrak{U}$ *and all* $y \in \mathbb{R}^D$*, we have*

$$\begin{aligned}\left\|x-\mathfrak{g}(x)\right\|_2 &\leq 2\epsilon\cdot(\min\{\mathtt{dist}(x,\mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x),\mathfrak{M})\} + \mathfrak{R}) \\ &\leq 2\epsilon\cdot\left(\min\{\left\|y-x\right\|_2, \left\|y-\mathfrak{g}(x)\right\|_2\} + \mathtt{dist}(y,\mathfrak{M}) + \mathfrak{R}\right)\end{aligned}$$

and

$$\left\|x-\mathfrak{g}(x)\right\|_2^2 \leq 12\epsilon^2\left(\min\{\left\|y-x\right\|_2, \left\|y-\mathfrak{g}(x)\right\|_2\}^2 + \mathtt{dist}(y,\mathfrak{M})^2 + \mathfrak{R}^2\right) .$$

Proof. Consider an arbitrary but fixed $x \in \mathfrak{U}$. By Definition 9.7, there is a grid cell in some ring $\mathfrak{U}_{l,j}$ that contains x and its representative $\mathfrak{g}(x)$.

If $j=0$, then we have $\left\|x-\mathfrak{g}(x)\right\|_2 \leq \epsilon\mathfrak{R}$, due to Definition 9.7. If $j \geq 1$, then from Definition 9.7 and Definition 9.6 we know that

$$\left\|x-\mathfrak{g}(x)\right\|_2 \leq \epsilon\cdot 2^j\mathfrak{R} = 2\epsilon\cdot(2^{j-1}\mathfrak{R})$$

and

$$2^{j-1}\mathfrak{R} \leq \min\{\mathtt{dist}(x,\mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x),\mathfrak{M})\} .$$

Hence, if $j \geq 1$, we have

$$\left\|x-\mathfrak{g}(x)\right\|_2 \leq 2\epsilon\cdot\min\{\mathtt{dist}(x,\mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x),\mathfrak{M})\} .$$

By taking the sum of the bounds for the case $j=0$ and $j \geq 1$, we obtain

$$\begin{aligned}\left\|x-\mathfrak{g}(x)\right\|_2 &\leq 2\epsilon\cdot\min\{\mathtt{dist}(x,\mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x),\mathfrak{M})\} + \epsilon\mathfrak{R} \\ &\leq 2\epsilon\cdot(\min\{\mathtt{dist}(x,\mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x),\mathfrak{M})\} + \mathfrak{R}) .\end{aligned}$$

This yields the first inequality in the claim.

Observe that for all $z, z' \in \mathbb{R}^D$ and $C \subset \mathbb{R}^D$ we have $\mathtt{dist}(z, C) \leq \|z - z'\|_2 + \mathtt{dist}(z', C)$ due to the triangle inequality. Hence, for all $y \in \mathbb{R}^D$, we have

$$\begin{aligned}\min\{\mathtt{dist}(x, \mathfrak{M}), \mathtt{dist}(\mathfrak{g}(x), \mathfrak{M})\} &\leq \min\left\{\|x - y\|_2 + \mathtt{dist}(y, \mathfrak{M}),\ \|\mathfrak{g}(x) - y\|_2 + \mathtt{dist}(y, \mathfrak{M})\right\} \\ &= \min\left\{\|x - y\|_2,\ \|\mathfrak{g}(x) - y\|_2\right\} + \mathtt{dist}(y, \mathfrak{M}) \ .\end{aligned}$$

This yields the second inequality in the claim.

Finally, applying Lemma A.2 yields the last inequality in the claim. □

9.4.4 Replace Means by Their Representatives (K-Means)

Now consider some means $C \subseteq \mathfrak{U}$ contained in the search space and some point $x \in \mathbb{R}^D$. Replace the means by their representatives $\mathfrak{g}(C)$. With the help of the previous result, we can compare the minimum distances $\mathtt{dist}(x, C)$ and $\mathtt{dist}(x, \mathfrak{g}(C))$ with one another. This result corresponds to (part of) the proof of Lemma 5.11 from (Chen, 2009, p. 937).

Lemma 9.13 (replacing means by their representatives)**.** *Let $\epsilon \in (0, 1]$ and $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$. Let $\mathfrak{U} \subset \mathbb{R}^D$ be the search space described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and let $\mathfrak{g}$ be a representative function defined by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and ϵ.*

Let $C = (\mu_k)_{k \in [K]} \subseteq \mathfrak{U}$, $\mathfrak{g}(C) \coloneqq \left(\mathfrak{g}(\mu_l)\right)_{l \in [L]}$, and $x \in \mathbb{R}^D$. Then,

$$\mathtt{dist}(x, \mathfrak{g}(C))^2 \leq \mathtt{dist}(x, C)^2 + 18\epsilon\left(2\,\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right)$$

and

$$\mathtt{dist}(x, C)^2 \leq \mathtt{dist}(x, \mathfrak{g}(C))^2 + 18\epsilon\left(2\,\mathtt{dist}(x, \mathfrak{g}(C))^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \ .$$

Proof. Consider an index $l \in [K]$ where $\mathtt{dist}(x, C) = \|x - \mu_l\|_2$. Observe that

$$\begin{aligned}\mathtt{dist}(x, \mathfrak{g}(C))^2 - \mathtt{dist}(x, C)^2 &= \min_{k \in [K]} \|x - \mathfrak{g}(\mu_k)\|_2^2 - \|x - \mu_l\|_2^2 \\ &\leq \|x - \mathfrak{g}(\mu_l)\|_2^2 - \|x - \mu_l\|_2^2 \\ &\leq \|\mu_l - \mathfrak{g}(\mu_l)\|_2^2 + 2\,\|\mu_l - \mathfrak{g}(\mu_l)\|_2 \cdot \|x - \mu_l\|_2 \ . && \text{(Lemma 9.4)}\end{aligned}$$

The first summand can be bounded by

$$\begin{aligned}\|\mu_l - \mathfrak{g}(\mu_l)\|_2^2 &\leq 12\epsilon^2\left(\|\mu_l - x\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) && \text{(Lemma 9.12)} \\ &= 12\epsilon^2\left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \ . && (\mathtt{dist}(x, C) = \|x - \mu_l\|_2)\end{aligned}$$

The second summand can be bounded by

$$\begin{aligned}&2\,\|\mu_l - \mathfrak{g}(\mu_l)\|_2\,\|x - \mu_l\|_2 \\ &\leq 4\epsilon\left(\|x - \mu_l\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R}\right) \cdot \|x - \mu_l\|_2 && \text{(Lemma 9.12)} \\ &\leq 2\epsilon\left(\left(\|x - \mu_l\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R}\right)^2 + \|x - \mu_l\|_2^2\right) && \text{(Lemma A.2)} \\ &\leq 2\epsilon\left(3 \cdot \left(\|x - \mu_l\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) + \|x - \mu_l\|_2^2\right) && \text{(Lemma A.2)} \\ &\leq 6\epsilon\left(2\,\|x - \mu_l\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \\ &= 6\epsilon\left(2\,\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \ . && (\mathtt{dist}(x, C) = \|x - \mu_l\|_2)\end{aligned}$$

By combining these bounds, we obtain

$$\begin{aligned}&\mathtt{dist}(x, \mathfrak{g}(C))^2 - \mathtt{dist}(x, C)^2 \\ &\leq 12\epsilon^2\left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) + 6\epsilon\left(2\,\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \\ &\leq 18\epsilon\left(2\,\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) \ .\end{aligned}$$

This yields the first part of the claim.

Since the second part of the claim can be proven analogously, we only point out the main arguments. Consider an index $l \in [L]$ where $\mathtt{dist}(x, \mathfrak{g}(C)) = \left\|x - \mathfrak{g}(\mu_l)\right\|_2$. Then, due to Lemma 9.4, we have

$$\mathtt{dist}(x, \mathfrak{g}(C))^2 - \mathtt{dist}(x, C)^2 \leq \left\|\mu_l - \mathfrak{g}(\mu_l)\right\|_2^2 + 2\left\|\mu_l - \mathfrak{g}(\mu_l)\right\|_2 \cdot \left\|x - \mathfrak{g}(\mu_l)\right\|_2^2 .$$

With the help of Lemma 9.12 and $\mathtt{dist}(x, \mathfrak{g}(C)) = \left\|x - \mathfrak{g}(\mu_l)\right\|_2$, one can bound the first summand by

$$\left\|\mu_l - \mathfrak{g}(\mu_l)\right\|_2^2 \leq 12\epsilon^2 \left(\mathtt{dist}(x, \mathfrak{g}(C))^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) .$$

Using Lemma 9.12, Lemma A.2, and $\mathtt{dist}(x, \mathfrak{g}(C)) = \left\|x - \mathfrak{g}(\mu_l)\right\|_2$, we can bound

$$2\left\|\mu_l - \mathfrak{g}(\mu_l)\right\|_2 \cdot \left\|x - \mathfrak{g}(\mu_l)\right\|_2^2 \leq 6\epsilon \left(2\,\mathtt{dist}(x, \mathfrak{g}(C))^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) .$$

By combining these bounds we obtain the second claim. □

Consequently, the difference between the K-means cost $\mathrm{km}_X(C)$ and the K-means cost $\mathrm{km}_X(\mathfrak{g}(C))$ can be bounded as follows. This result corresponds to Lemma 5.11 from (Chen, 2009, p. 937).

Corollary 9.14 (replacing means by their representatives)**.** *Let $\epsilon \in (0, 1]$ and $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$. Let $\mathfrak{U} \subset \mathbb{R}^D$ be the search space described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and let $\mathfrak{g}$ be a representative function defined by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and ϵ.*

Let $C = (\mu_k)_{k\in[K]} \subseteq \mathfrak{U}$, $\mathfrak{g}(C) := \left(\mathfrak{g}(\mu_k)\right)_{k\in[K]}$, and $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_{\geq 0}\right)$. Then,

$$\mathrm{km}_X(\mathfrak{g}(C)) \leq \mathrm{km}_X(C) + 18\epsilon \left(2\,\mathrm{km}_X(C) + \mathrm{km}_X(\mathfrak{M}) + \mathbf{w}(X)\mathfrak{R}^2\right) ,$$

and

$$\mathrm{km}_X(C) \leq \mathrm{km}_X(\mathfrak{g}(C)) + 18\epsilon \left(2\,\mathrm{km}_X(\mathfrak{g}(C)) + \mathrm{km}_X(\mathfrak{M}) + \mathbf{w}(X)\mathfrak{R}^2\right) .$$

9.4.5 Replace Means by Their Representatives (r-Fuzzy K-Means)

The K-means cost of a point (x, w) in X is simply the minimum squared Euclidean distance $\mathtt{dist}(x, C)^2$ multiplied by the weight w (see Problem 4.3). In contrast, the r-fuzzy K-means cost of (x, w) with respect to C is given by the sum over all squared Euclidean distances $\left\|x - \mu_k\right\|_2^2$, with $k \in [K]$, that are multiplied with the respective fuzzified probabilities $r(p_{nk})$ and the weight w. Hence, transfering Lemma 9.13 to the r-fuzzy K-means cost is more involved.

Lemma 9.15 (replacing means by their representatives)**.** *Let $\epsilon \in (0, 1]$ and $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M}) \in \mathfrak{Descr}$. Let $\mathfrak{U} \subset \mathbb{R}^D$ be the search space described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and let $\mathfrak{g}$ be a representative function defined by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and $\tilde{\epsilon} = \epsilon/36$.*

Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a [0, 1]-reducing fuzzifier function, $(x, w) \in \mathbb{R}^D \times \mathbb{R}_+$, $C = (\mu_k)_{k\in[K]} \subseteq \mathfrak{U}$, and $\mathfrak{g}(C) := \left(\mathfrak{g}(\mu_k)\right)_{k\in[K]}$. Then,

$$\frac{1}{w}\left|\phi^{(r)}_{((x,w))}(C) - \phi^{(r)}_{((x,w))}(\mathfrak{g}(C))\right| \leq \epsilon \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2\right) .$$

Proof. Consider an arbitrary but fixed $(x, w) \in X$. Let $(p_k)_{k\in[K]}$ and $(\tilde{p}_k)_{k\in[K]}$ be the r-fuzzy clusterings of (x, w) induced by C and $\mathfrak{g}(C)$, respectively. Since r is [0, 1]-reducing, we know

$$\sum_{k=1}^{K} r(p_k) \leq 1 \quad \text{and} \quad \sum_{k=1}^{K} r(\tilde{p}_k) \leq 1 . \tag{9.3}$$

Let

$$\mathcal{E} := \frac{1}{w}\left|\phi^{(r)}_{((x,w))}(C) - \phi^{(r)}_{((x,w))}(\mathfrak{g}(C))\right| = \left|\sum_{k=1}^{K} r(p_k)\left\|x-\mu_k\right\|_2^2 - r(\tilde{p}_k)\left\|x-\mathfrak{g}(\mu_k)\right\|_2^2\right| .$$

First of all note that if the first term in $\mathcal{E}$ is larger than the second, then

$$\begin{aligned}
\mathcal{E} &= \sum_{k=1}^{K} r(p_k)\left\|x-\mu_k\right\|_2^2 - r(\tilde{p}_k)\left\|x-\mathfrak{g}(\mu_k)\right\|_2^2 \\
&\leq \sum_{k=1}^{K} r(\tilde{p}_k)\left(\left\|x-\mu_k\right\|_2^2 - \left\|x-\mathfrak{g}(\mu_k)\right\|_2^2\right) && \text{(Lemma 5.17)} \\
&\leq \sum_{k=1}^{K} r(\tilde{p}_k)\left(\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2^2 + 2\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2\left\|x-\mathfrak{g}(\mu_k)\right\|_2\right) . && \text{(Lemma 9.4)}
\end{aligned}$$

Analogously, if the second term in $\mathcal{E}$ is larger than the first, then

$$\mathcal{E} \leq \sum_{k=1}^{K} r(p_k)\left(\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2^2 + 2\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2\left\|x-\mu_k\right\|_2\right) .$$

Hence, we have

$$\begin{aligned}
\mathcal{E} \leq \max\Bigg\{ &\sum_{k=1}^{K} r(\tilde{p}_k)\left(\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2^2 + 2\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2\left\|x-\mathfrak{g}(\mu_k)\right\|_2\right) , && (9.4) \\
&\sum_{k=1}^{K} r(p_k)\left(\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2^2 + 2\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2\left\|x-\mu_k\right\|_2\right)\Bigg\} . && (9.5)
\end{aligned}$$

Before we bound this value, observe that

$$\begin{aligned}
&\sum_{k=1}^{K} r(\tilde{p}_k)\left\|x-\mathfrak{g}(\mu_k)\right\|_2^2 && (9.6) \\
&= \frac{1}{w}\phi^{(r)}_{((x,w))}(\mathfrak{g}(C)) && \text{(Lemma 5.17)} \\
&\leq \mathtt{dist}(x,\mathfrak{g}(C))^2 && \text{(Lemma 6.1)} \\
&\leq \mathtt{dist}(x,C)^2 + 18\tilde{\epsilon}\left(2\,\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) && \text{(Lemma 9.13)} \\
&= (1+36\epsilon)\,\mathtt{dist}(x,C)^2 + 18\tilde{\epsilon}\,\mathtt{dist}(x,\mathfrak{M})^2 + 18\tilde{\epsilon}\mathfrak{R}^2 && (\text{where } \tilde{\epsilon} = \epsilon/36) \\
&\leq 2\,\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2 . && (9.7)
\end{aligned}$$

Next, we bound the single terms in (9.4). Observe that

$$\begin{aligned}
&\sum_{k=1}^{K} r(\tilde{p}_k)\left\|\mu_k-\mathfrak{g}(\mu_k)\right\|_2^2 \\
&\leq 12\tilde{\epsilon}^2 \sum_{k=1}^{K} r(\tilde{p}_k)\left(\left\|x-\mathfrak{g}(\mu_k)\right\|_2^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) && \text{(Lemma 9.12)} \\
&\leq 12\tilde{\epsilon}^2 \left(\sum_{k=1}^{K} r(\tilde{p}_k)\left\|x-\mathfrak{g}(\mu_k)\right\|_2^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) && \text{(Equation (9.3))} \\
&\leq 12\tilde{\epsilon}^2 \left(2\,\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) && \text{(Equation (9.7))} \\
&\leq 24\tilde{\epsilon}^2 \left(\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) \\
&\leq \tilde{\epsilon}\left(\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2\right) . && (\tilde{\epsilon} = \epsilon/36)
\end{aligned}$$

Moreover, we obtain

$$
\begin{aligned}
&\sum_{k=1}^{K} r(\tilde{p}_k) \left\| \mu_k - \mathfrak{g}(\mu_k) \right\|_2 \left\| x - \mathfrak{g}(\mu_k) \right\|_2 \\
&\leq 2\tilde{\epsilon} \sum_{k=1}^{K} r(\tilde{p}_k) \left(\left\| \mathfrak{g}(\mu_k) - x \right\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R} \right) \left\| x - \mathfrak{g}(\mu_k) \right\|_2 && \text{(Lemma 9.12)} \\
&\leq \tilde{\epsilon} \sum_{k=1}^{K} r(\tilde{p}_k) \left(\left(\left\| \mathfrak{g}(\mu_k) - x \right\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R} \right)^2 + \left\| x - \mathfrak{g}(\mu_k) \right\|_2^2 \right) && \text{(Lemma A.2)} \\
&\leq \tilde{\epsilon} \sum_{k=1}^{K} r(\tilde{p}_k) \left(3\,\mathtt{dist}(x, \mathfrak{M})^2 + 3\mathfrak{R}^2 + 4 \left\| x - \mathfrak{g}(\mu_k) \right\|_2^2 \right) && \text{(Lemma A.2)} \\
&\leq 4\tilde{\epsilon} \sum_{k=1}^{K} r(\tilde{p}_k) \left(\mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 + \left\| x - \mathfrak{g}(\mu_k) \right\|_2^2 \right) \\
&\leq 4\tilde{\epsilon} \left(\mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 + \sum_{k=1}^{K} r(\tilde{p}_k) \left\| x - \mathfrak{g}(\mu_k) \right\|_2^2 \right) && \text{(Equation (9.3))} \\
&\leq 8\tilde{\epsilon} \left(\mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 + 2\,\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) && \text{(Equation (9.7))} \\
&\leq 8\tilde{\epsilon} \left(\mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 + \mathtt{dist}(x, C)^2 \right) .
\end{aligned}
$$

Next, we bound the single terms in (9.5). We have

$$
\begin{aligned}
&\sum_{k=1}^{K} r(p_k) \left\| \mu_k - \mathfrak{g}(\mu_k) \right\|_2^2 \\
&\leq 12\tilde{\epsilon}^2 \sum_{k=1}^{K} r(p_k) \left(\left\| x - \mu_k \right\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) && \text{(Lemma 9.12)} \\
&\leq 12\tilde{\epsilon}^2 \left(\sum_{k=1}^{K} r(p_k) \left\| x - \mu_k \right\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) && \text{(Equation (9.3))} \\
&\leq 12\tilde{\epsilon}^2 \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) && \text{(Lemma 6.1 + Lemma 5.17)} \\
&\leq \tilde{\epsilon} \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) . && (\tilde{\epsilon} = \epsilon/36)
\end{aligned}
$$

Furthermore, we can bound

$$
\begin{aligned}
&\sum_{k=1}^{K} r(p_k) \left\| \mu_k - \mathfrak{g}(\mu_k) \right\|_2 \left\| x - \mu_k \right\|_2 \\
&\leq 2\tilde{\epsilon} \sum_{k=1}^{K} r(p_k) \left(\left\| \mu_k - x \right\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R} \right) \left\| x - \mu_k \right\|_2 && \text{(Lemma 9.12)} \\
&\leq \tilde{\epsilon} \sum_{k=1}^{K} r(p_k) \left(\left(\left\| \mu_k - x \right\|_2 + \mathtt{dist}(x, \mathfrak{M}) + \mathfrak{R} \right)^2 + \left\| x - \mu_k \right\|_2^2 \right) && \text{(Lemma 9.4)} \\
&\leq \tilde{\epsilon} \sum_{k=1}^{K} r(p_k) \left(4 \left\| \mu_k - x \right\|_2^2 + 3\,\mathtt{dist}(x, \mathfrak{M})^2 + 3\mathfrak{R}^2 \right) && \text{(Lemma 9.4)} \\
&\leq 4\tilde{\epsilon} \sum_{k=1}^{K} r(p_k) \left(\left\| \mu_k - x \right\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) \\
&\leq 4\tilde{\epsilon} \left(\sum_{k=1}^{K} r(p_k) \left\| \mu_k - x \right\|_2^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) && \text{(Equation (9.3))} \\
&\leq 4\tilde{\epsilon} \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) . && \text{(Lemma 6.1 + Lemma 5.17)}
\end{aligned}
$$

Combining the bounds on the terms from (9.4) and (9.5), we obtain that

$$
\mathcal{E} \leq \max \left\{ 9\tilde{\epsilon} \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right),\ 5\tilde{\epsilon} \left(\mathtt{dist}(x, C)^2 + \mathtt{dist}(x, \mathfrak{M})^2 + \mathfrak{R}^2 \right) \right\}
$$

$$\leq \epsilon \left(\mathtt{dist}(x,C)^2 + \mathtt{dist}(x,\mathfrak{M})^2 + \mathfrak{R}^2 \right) . \qquad (\tilde{\epsilon} = \epsilon/36)$$

This yields the claim. □

Given this result, it is easy to see that the difference between the r-fuzzy K-means costs $\phi_X^{(r)}(C)$ and $\phi_X^{(r)}(\mathfrak{g}(C))$ can be bounded as follows.

Corollary 9.16 (replacing means by their representatives)**.** *Let $\epsilon \in (0,1]$ and $(\mathfrak{E},\mathfrak{R},\mathfrak{M}) \in \mathfrak{Descr}$. Let $\mathfrak{U} \subset \mathbb{R}^D$ be the search space described by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ and let $\mathfrak{g}$ be a representative function defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ and $\tilde{\epsilon} = \epsilon/36$.*

Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a [0,1]-reducing fuzzifier function, $K \in \mathbb{N}$, $C = (\mu_k)_{k\in[K]} \subseteq \mathfrak{U}$, $\mathfrak{g}(C) := \left(\mathfrak{g}(\mu_k)\right)_{k\in[K]}$, and $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$. Then,

$$\left| \phi_X^{(r)}(C) - \phi_X^{(r)}(\mathfrak{g}(C)) \right| \leq \epsilon \left(\mathrm{km}_X(C) + \mathrm{km}_X(\mathfrak{M}) + \mathbf{w}(X)\mathfrak{R}^2 \right) .$$

Proof. Note that $\left| \phi_X^{(r)}(C) - \phi_X^{(r)}(\mathfrak{g}(C)) \right| \leq \sum_{n=1}^{N} \left| \phi_{((x_n,w_n))}^{(r)}(C) - \phi_{((x_n,w_n))}^{(r)}(\mathfrak{g}(C)) \right|$. Bound each summand via Lemma 9.15. This yields the claim. □

Discussion. This bound and all the bounds that we presented in the previous section heavily depend on the definition of the K-means cost. Consequently, our application of this construction also heavily depends on the relation between the r-fuzzy K-means and the K-means cost function. Our goal is that the discrete search space $\mathfrak{g}(\mathfrak{U})$ contains a good representative for each solution contained in the continuous search space $\mathfrak{U}$. That is, we want that the upper bound from Corollary 9.16 computes to some small multiple of $\phi_X^{(r)}(C)$. To this end, we need the following paramter setting: To be able to bound the first summand $\epsilon \cdot \mathrm{km}_X(C)$ by $\phi_X^{(r)}(C)$ via Lemma 6.1, we require that

$$\epsilon \propto \mathbf{c}_r(K)^{-1} .$$

Given this setting of ϵ, it suffices if $\mathrm{km}_X(\mathfrak{M}) \in \mathcal{O}(\mathrm{km}_X(C))$. We can only guarantee that this is true for all possible $C \subseteq \mathbb{R}^D$ with $|C| \leq K$ if

$$\mathrm{km}_X(\mathfrak{M}) \leq \alpha \, \mathrm{km}_{(X,K)}^{OPT}$$

for some constant $\alpha \in \mathcal{O}(1)$. Given this bound on $\mathrm{km}_X(\mathfrak{M})$, it remains to choose the parameter $\mathfrak{R}$ such that

$$\mathbf{w}(X)\mathfrak{R}^2 \in \mathcal{O}(\mathrm{km}_X(\mathfrak{M})) .$$

To sum up, there is a strong dependence on notions from K-means clustering.

9.5 A Discrete Search Space

In this section, we focus on the discrete search space, which contains all possible representatives of points from the search space. Clearly, the size of a discrete search space $\mathfrak{G}$, which is described by $(\mathfrak{E},\mathfrak{R},\mathfrak{M})$ and ϵ, linearly depends on the number $|\mathfrak{M}|$ of the given means and the exponent $\mathfrak{E}$ of the radius (see Definition 9.9). The following result and its proof correspond to Claim 4.3 and Claim 5.7 from (Chen, 2009, p. 930,936).

Lemma 9.17 (Size of the Discrete Search Space)**.** *The size of a discrete search space $\mathfrak{G} \subset \mathbb{R}^D$ defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M}) \in \mathfrak{Descr}$ and $\epsilon \in (0,1]$ is bounded by*

$$|\mathfrak{G}| \leq |\mathfrak{M}| \cdot (\mathfrak{E}+1) \cdot \left(\frac{16\sqrt{\pi \cdot e}}{\epsilon} \right)^D .$$

Proof. Let $\mathfrak{M} := (\mathfrak{m}_l)_{l\in[L]}$. Recall Definition 9.7 and Definition 9.9. Consider the grid which partitions the rings $\mathfrak{U}_{l,j}$. For each $l \in [|\mathfrak{M}|]$ and $j \in \{0,\dots,\mathfrak{E}\}$, the ring $\mathfrak{U}_{l,j}$ is partitioned into an axis-parallel grid with side length $s_j := \epsilon \cdot \frac{2^j\mathfrak{R}}{\sqrt{D}}$. Hence, the volume of each grid cell in which $\mathfrak{U}_{l,j}$ is split up is given by

$$v_j := s_j^D = \left(\epsilon \cdot \frac{2^j\mathfrak{R}}{\sqrt{D}}\right)^D .$$

By definition, the ring $\mathfrak{U}_{l,j}$ is completely contained in a (closed) ball around $\mathfrak{m}_l$ with radius $2^j\mathfrak{R}$. Observe that the (smallest) cube containing a grid cell may not be completely covered by this ball, but certainly by the ball around $\mathfrak{m}_l$ with radius $2^j\mathfrak{R} + s_j \le 2^{j+1}\mathfrak{R}$. The volume of the latter ball is given by

$$V_j := \frac{\pi^{D/2}(2^{j+1}\mathfrak{R})^D}{\Gamma(D/2+1)} ,$$

where Γ denotes Euler's gamma function (Hopcroft and Kannan, 2017, pp. 16). Hence, the number of grid cells in which $\mathfrak{U}_{l,j}$ is split up is at most

$$\frac{V_j}{v_j} = \frac{\pi^{D/2}(2^{j+1}\mathfrak{R})^D}{\Gamma(D/2+1)} \cdot \left(\frac{\sqrt{D}}{\epsilon\cdot 2^j\mathfrak{R}}\right)^D = \frac{\pi^{D/2}2^D\sqrt{D}^D}{\Gamma(D/2+1)\cdot\epsilon^D} .$$

Note that $\Gamma(n) = (n-1)!$ for all $n \in \mathbb{N}$ (Hopcroft and Kannan, 2017, p. 17). Hence, $\Gamma(D/2+1) \ge \Gamma(\lfloor D/2+1\rfloor) \ge (\lfloor D/2+1\rfloor - 1)! = \lfloor D/2\rfloor!$ for all $D \ge 4$. From Stirling's approximation, we know that $n! \ge (n/e)^n$ for all $n \in \mathbb{N}$ (Cormen et al., 2001, p. 55). Hence,

$$\Gamma(D/2+1) \ge (\lfloor D/2\rfloor/e)^{\lfloor D/2\rfloor} \ge (D/(4e))^{D/2-1} .$$

Therefore,

$$\frac{V_j}{v_j} \le \frac{\pi^{D/2}2^D D^{D/2}(4e)^{D/2-1}}{D^{D/2-1}\cdot\epsilon^D} = \frac{\pi^{D/2}2^D D(4e)^{D/2}}{\epsilon^D} \le \frac{\pi^{D/2}4^D(4e)^{D/2}}{\epsilon^D} \le \left(\frac{16\sqrt{\pi\cdot e}}{\epsilon}\right)^D .$$

Overall, there are $L\cdot(\mathfrak{E}+1)$ different rings. Hence, the total number of grid cells is at most $L\cdot(\mathfrak{E}+1)\cdot\left(\frac{16\sqrt{\pi\cdot e}}{\epsilon}\right)^D$. This yields the claim. □

Lemma 9.18. *A discrete search space $\mathfrak{G} \subset \mathbb{R}^D$ defined by $(\mathfrak{E},\mathfrak{R},\mathfrak{M}) \in \mathfrak{Descr}$ and $\epsilon \in (0,1]$ can be computed in time*

$$\mathcal{O}\left(|\mathfrak{M}|^2\cdot\mathfrak{E}\cdot(2/\epsilon)^{3D}\right) .$$

Proof. Recall our analysis of the size of $\mathfrak{G}$ from the proof of Lemma 9.17. Due to this analysis, we know that we have to iterates over $\mathcal{O}\left(|\mathfrak{M}|\cdot\mathfrak{E}\cdot(1/\epsilon)^D\cdot 2^D\right)$ grid points. Testing whether a point lies inside a certain ring needs time $\mathcal{O}(D)$. Determining which of the means in $\mathfrak{M}$ is closest to a point needs time $\mathcal{O}(|\mathfrak{M}|D)$. Combining these bounds yields the claim. □

Chapter 10

An ϵ-Approximate Mean Set

In the previous chapter, we constructed two search spaces with respect to a data set X: a continuous search space $\mathfrak{U} \subseteq \mathbb{R}^D$ and a discrete search space $\mathfrak{G} \subset \mathbb{R}^D$ of finite size $|\mathfrak{G}| < \infty$. These search spaces have the nice property that we can map mean vectors $C \subseteq \mathfrak{U}$ from the continuous search space $\mathfrak{U}$ to representatives $\mathfrak{g}(C) \subseteq \mathfrak{G}$ in the discrete search space $\mathfrak{G}$ such that the r-fuzzy K-means cost $\phi_X^{(r)}(\mathfrak{g}(C))$ of the representatives is similar to the r-fuzzy K-means cost $\phi_X^{(r)}(C)$ of the original means C. The idea pursued in this chapter is the following: If we knew that good mean vectors are contained in the continuous search space $\mathfrak{U}$, then the aforementioned properties ensure that the discrete search space $\mathfrak{G}$ also contains a similarly good mean vectors. In particular, if near-optimal mean vectors are contained in $\mathfrak{U}$, then there are also near-optimal mean vectors in the discrete search space $\mathfrak{G}$. As $\mathfrak{G}$ is finite, we could find these near-optimal mean vectors via an exhaustive search.

In this chapter, we show that this idea works: We can indeed construct a discrete search space that contains means that induce a $(1+\epsilon)$-approximation to the r-fuzzy K-means problem for certain fuzzifier functions r. We call such a set of means an ϵ-approximate mean set.

Overview. In Section 10.1 and Section 10.2, we give a brief overview of the related work and our contribution. In Section 10.3, we state our formal definition of approximate means sets for the r-fuzzy K-means problem, a construction of such sets, and an approximation algorithm. In Section 10.5, we state the complete proofs.

Publication. In this chapter, we generalize and correct the corresponding result from Blömer et al. (2016), which deals with the classical fuzzy K-means problem.

10.1 Related Work

In this chapter, we reuse the construction and results that we presented in Chapter 9. These results are based on the work of Chen (2009). Besides that, we make use of the notion of approximate mean sets. Usually, these sets are called approximate *centroid* sets (Matoušek, 2000). We hope to avoid confusion (and not cause any) by sticking with the term 'means' instead of introducing the synonym 'centroid'. There are several algorithms for the K-means problem that make use of this notion: Matoušek (2000) presents a $(1+\epsilon)$-approximation algorithm for the K-means problem with runtime $\mathcal{O}(|X|\epsilon^{-2K^2D}\log(|X|)^K)$. Effros and Schulman (2004) construct a $(1+\epsilon)$-algorithm for the K-means problem which runs in time $\mathrm{poly}(K)\cdot(D/\epsilon)^{\mathcal{O}(D)}|X|\log\log(|X|)+(D/\epsilon)^{\mathcal{O}(KD)}$. The work of Feldman et al. (2007) is based on a construction that yields a weak coreset for the K-means problem and a $(1+\epsilon)$-approximation algorithm with running time $\mathcal{O}(|X|KD)+D\cdot\mathrm{poly}(K/\epsilon)+2^{\tilde{\mathcal{O}}(K/\epsilon)}$.

10.2 Contribution

First, we show how a finite ϵ-approximate mean set can be computed for the r-fuzzy K-means problem, given that the fuzzifier function is increase-bounded, contribution-bounded, and $[0,1]$-reducing. More precisely, we show that a slightly refined version of the construction of Chen (2009) can be used to compute such a set. Second, we use this result to show that there is a $(1+\epsilon)$-approximation algorithm for the r-fuzzy K-means problem (for the aforementioned class of fuzzifier functions). This algorithm simply performs an exhaustive search through all the solutions in the aforementioned ϵ-approximate mean set. Later, in Chapter 11, we will use a dimension reduction technique to improve the performance of this algorithm. For an overview and comparison of all our approximation algorithms, we refer to Chapter 13.

10.3 Main Result

Let us start by formalizing our notion of an ϵ-approximate mean set. For the sake of simplicity, we denote a vector of means as a *solution* and use the following notation:

Notation 10.1 (short notation)**.** *For $M \subseteq \mathbb{R}^D$ and $K \in \mathbb{N}$, the set of solutions with exactly K mean vectors from M is $M^K := \{(\mu_1,\dots,\mu_K) \mid \forall k \in [K]\colon\ \mu_k \in M\}$. Analogously, we let $M^{\leq K} := \bigcup_{k\in[K]} \{(\mu_1,\dots,\mu_k) \mid \forall l \in [k]\colon\ \mu_l \in M\}$.*

An ϵ-approximate mean set is a set of solutions that contains a $(1+\epsilon)$-approximation to the r-fuzzy K-means problem.

Definition 10.2 (ϵ-approximate mean set)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $\epsilon \in [0,1]$, and $K \in \mathbb{N}$.*

A set $\Theta \subseteq (\mathbb{R}^D)^{\leq K}$ is an ϵ-approximate mean set of X for the r-fuzzy K-means problem if

$$\exists C^* \in \Theta\colon\ \phi_X^{(r)}(C^*) \leq (1+\epsilon)\cdot \phi_{(X,K,r)}^{OPT}\ .$$

Usually, one calls these sets (K,ϵ)-approximate mean sets. Here, K is either clear from context or stated explicitly, as part of the term "r-fuzzy K-means problem". Besides that, we point out that allowing for solutions with less than K mean vectors is not important in this chapter, but it will be handy in Chapter 12.

The following result shows that one can construct a discrete search space $\mathfrak{G}$ that is an ϵ-approximate mean set.

Theorem 10.3. *Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, a fuzzifier $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is $\mathbf{i}_r$-increase-bounded, $\mathbf{c}_r$-contribution-bounded, and $[0,1]$-reducing, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$, Algorithm 10 computes a discrete search space $\mathfrak{G} \subseteq \mathbb{R}^D$ such that $\mathfrak{G}^K$ is an ϵ-approximate mean set of X for the r-fuzzy K-means problem.*

The size of $\mathfrak{G}$ is

$$|\mathfrak{G}| = \mathcal{O}\left(\log\left(|X|\,\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\cdot K\cdot \epsilon^{-D}\cdot H_{(r,K,\epsilon)}\right)\ ,$$

where

$$H_{(r,K,\epsilon)} = \max\left\{\log\left(r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1}\right),\ \log\left(\mathbf{c}_r(K)^{-1}\right),\ 1\right\}\ .$$

Before we provide a proof in Section 10.5, let us consider an application of this constructive result.

Algorithm 10 Exhaustive Search Through a Discrete Search Space

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$

1: Construct an unweighted data set $\hat{X}$ that, for each $n \in [N]$, contains $\lceil w_n / \mathrm{w}_{\min}^{(X)} \rceil$ copies of the data point $(x_n, 1)$.

2: Apply the algorithm from (Matoušek, 2000) to compute an α'-approximation $\mathfrak{M}$ to the K-means problem with respect to $\hat{X}$, where α' is some constant.

3: Set

$$\alpha := 2 \cdot \alpha' \quad \text{and} \quad b := \max\left\{ r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1}, \mathbf{c}_r(K)^{-1} \right\} .$$

4: Let

$$\mathfrak{E} := \left\lfloor \frac{1}{2} \log\left(9 \cdot \alpha \cdot b \cdot \frac{\mathbf{w}(X)}{\mathrm{w}_{\min}^{(X)}} \right) \right\rfloor \quad \text{and} \quad \mathfrak{R} := \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{\alpha \mathbf{w}(X)}} .$$

5: Compute a discrete search space $\mathfrak{G}$ described by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and

$$\tilde{\epsilon} := \epsilon \cdot \frac{\mathbf{c}_r(K)}{72 \cdot (2+\alpha)} .$$

6: Determine $C := \arg\min\left\{ \phi_X^{(r)}(C) \;\middle|\; C \subseteq \mathfrak{G} \text{ with } |C| = K \right\}$.

10.4 Application (Corollary 10.4)

By an exhaustive search through an ϵ-approximate mean set we can find a good solution.

Corollary 10.4. *Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, a fuzzifier $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is $\mathbf{i}_r$-increase-bounded, $\mathbf{c}_r$-contribution-bounded, and $[0,1]$-reducing, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$, Algorithm 10 computes means $C \subseteq \mathbb{R}^D$, $|C| = K$, such that*

$$\phi_X^{(r)}(C) \leq (1+\epsilon)\phi_{(X,K,r)}^{OPT} .$$

The algorithms' runtime is

$$\left(|X| \frac{\mathrm{w}_{\max}^{(X)}}{\mathrm{w}_{\min}^{(X)}} \right) \cdot \log\left(|X| \frac{\mathrm{w}_{\max}^{(X)}}{\mathrm{w}_{\min}^{(X)}} \right)^K \cdot 2^{\mathcal{O}(K^2 D \log(1/\epsilon))} \cdot H_{(r,K,\epsilon)}^K \cdot \mathbf{t}_r(K)$$

where

$$H_{(r,K,\epsilon)} = \max\left\{ \log\left(r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1} \right), \log\left(\mathbf{c}_r(K)^{-1}\right), 1 \right\} .$$

We prove this result in the next Section 10.5. Observe that Algorithm 10 is a polynomial-time approximation scheme (PTAS) for the r-fuzzy K-means problem if $K, \mathbf{c}_r(K), \mathbf{i}_r, D \in \mathcal{O}(1)$ are constants and if the given data sets X satisfy $\mathrm{w}_{\max}^{(X)} / \mathrm{w}_{\min}^{(X)} \in |X|^{\mathcal{O}(1)}$. Note that this observation covers all unweighted data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$ and all the fuzzifier functions presented in Section 5.3, except the exponential fuzzifier function e_γ because it is not increase-bounded.

Alternative. We can accelerate the algorithm a little bit by using a randomized K-means approximation algorithm instead of an deterministic K-means approximation algorithm in the second step of the algorithm.

Corollary 10.5 (randomized alternative)**.** *Consider a randomized variant of Algorithm 10 that uses the algorithm of Aggarwal et al. (2009) instead of the algorithm of Matoušek (2000) to compute a constant-factor approximation to the K-means problem with respect to $\hat{X}$ in the second step of the algorithm. Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, a fuzzifier $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is $\mathbf{i}_r$-increase-bounded, $\mathbf{c}_r$-contribution-bounded, and $[0,1]$-reducing, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$, this algorithm computes means $C \subseteq \mathbb{R}^D$, $|C| = K$, such that $\phi_X^{(r)}(C) \leq (1+\epsilon)\phi^{OPT}_{(X,K,r)}$, with constant probability. The algorithms' runtime is*

$$|X|\frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}KD + |X|\log\left(|X|\frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}\right)^{\mathcal{O}(K)} \cdot 2^{\mathcal{O}(K\log(K)D\log(1/\epsilon))} \cdot H^K_{(r,K,\epsilon)} \cdot \mathbf{t}_r(K)\,.$$

where $H_{(r,K,\epsilon)}$ is defined as in Corollary 10.4.

We prove this result in the next section.

Special Case. Consider the classical fuzzy K-means problem with the polynomial fuzzifier function $\mathrm{p}_m(x) = x^m$ with $m \in (1,\infty)$. Recall from Section 5.3.2 that we can set $\mathbf{c}_{\mathrm{p}_m}(K) = K^{m-1}$, $\mathbf{i}_{\mathrm{p}_m} = 4m$, and $\mathbf{t}_{\mathrm{p}_m}(K) = \Theta(K)$. Observe that $H_{(\mathrm{p}_m,K,\epsilon)} = \max\left\{m\log\left(\frac{16mK^2}{\epsilon}\right),\ (m-1)\log(K),\ 1\right\} \subseteq \mathcal{O}(m^2K\epsilon^{-1})$. Hence, $H^K_{(\mathrm{p}_m,K,\epsilon)} \in 2^{\mathcal{O}(\log(m)K\log(K)\log(1/\epsilon))}$. So, for unweighted data sets, the deterministic algorithm from Corollary 10.4 has runtime

$$|X|\log(|X|)^K \cdot 2^{\mathcal{O}(K^2\cdot D\cdot\log(m)\cdot\log(1/\epsilon))}\,,$$

while the randomized variant from Corollary 10.5 needs time

$$|X|\log(|X|)^{\mathcal{O}(K)} \cdot 2^{\mathcal{O}(K\log(K)D\log(m)\log(1/\epsilon))}\,.$$

So, for unweighted data sets, it does not really matter which algorithm we choose. The dependence on the number of clusters K is only slightly different.

10.5 Analysis

In the following, we analyse a single run of Algorithm 10 given $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$, and a $[0,1]$-reducing, $\mathbf{c}_r$-contribution-bounded and $\mathbf{i}_r$-increase-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Additionally, we let

$$\mathfrak{U} := \mathfrak{U}(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$$

where the elements from the tuple $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ are defined as in Algorithm 10. Consider the discrete space $\mathfrak{G}$ defined by the algorithm. Let $\mathfrak{g}$ be the corresponding representative function, i.e., the representative function that is defined by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ and $\tilde{\epsilon}$ satisfying

$$\mathfrak{G} = \{\mathfrak{g}(x) \mid x \in \mathfrak{U}\}\,.$$

Outline

The following analysis consists of five steps: First, we show that $\mathfrak{M}$ is a $(2\cdot\alpha')$-approximation to the K-means problem with respect to X. Second, we show that there exists a good approximation in the search space $\mathfrak{U}$. Third, we show that representatives $\mathfrak{g}(C) \subseteq \mathfrak{G}$ incur an r-fuzzy K-means cost similar to the means $C \subseteq \mathfrak{U}$. Fourth, we conclude from these results that it suffices to search through all solutions consisting of representatives from $\mathfrak{G}$ to find a good approximation. Finally, we analyse the runtime of the algorithm.

Approximation to the K-Means Problem

We need a constant-factor approximation to the K-means problem with respect to the weighted data set X. Therefore, we apply an algorithm that computes an α'-approximation for the K-means problem with respect to unweighted data sets to the data set $\hat{X}$, which contains copies of unweighted versions of the data points from X.

Observation 10.6. *The size of $\hat{X}$ is $\left|\hat{X}\right| = \sum_{n=1}^{N} \lceil w_n / \mathtt{w}_{\min}^{(X)} \rceil \leq 2 \sum_{n=1}^{N} w_n / \mathtt{w}_{\min}^{(X)} \leq 2|X| \cdot \mathtt{w}_{\max}^{(X)} / \mathtt{w}_{\min}^{(X)}$.*

Claim 10.7. *$\mathfrak{M}$ is a $(2\alpha')$-approximation to the K-means problem with respect to the weighted data set X. That is,*

$$\mathrm{km}_X(\mathfrak{M}) \leq (2\alpha') \cdot \mathrm{km}_{(X,K)}^{OPT} .$$

Proof. For all $n \in [N]$, we have $w_n / \mathtt{w}_{\min}^{(X)} \leq \lceil w_n / \mathtt{w}_{\min}^{(X)} \rceil \leq 2 w_n / \mathtt{w}_{\min}^{(X)}$ since $w_n / \mathtt{w}_{\min}^{(X)} \geq 1$. Hence, by definition of the K-means cost (see Problem 4.3), we have

$$\forall C \subseteq \mathbb{R}^D : \quad \frac{1}{\mathtt{w}_{\min}^{(X)}} \mathrm{km}_X(C) \leq \mathrm{km}_{\hat{X}}(C) \leq \frac{2}{\mathtt{w}_{\min}^{(X)}} \mathrm{km}_X(C) . \tag{10.1}$$

We know that $\mathfrak{M}$ is an α'-approximation for the K-means problem with respect to $\hat{X}$. Hence, $\mathrm{km}_{\hat{X}}(\mathfrak{M}) \leq \alpha' \mathrm{km}_{(\hat{X},K)}^{OPT}$. Let $C_X^{opt} \subseteq \mathbb{R}^D$, $\left|C^{opt}\right| = K$, be a solution with $\mathrm{km}_{(X,K)}^{OPT} = \mathrm{km}_X\left(C_X^{opt}\right)$. With (10.1), we can conclude that

$$\mathrm{km}_X(\mathfrak{M}) \leq \mathtt{w}_{\min}^{(X)} \mathrm{km}_{\hat{X}}(\mathfrak{M}) \leq \mathtt{w}_{\min}^{(X)} \alpha' \mathrm{km}_{(\hat{X},K)}^{OPT} \leq \mathtt{w}_{\min}^{(X)} \alpha' \mathrm{km}_{\hat{X}}\left(C_X^{opt}\right) \leq 2\alpha' \mathrm{km}_{(X,K)}^{OPT} .$$

This yields the claim. □

Existence of a Good Solution in $\mathfrak{U}$

Let us start by showing that there exists a good approximation in the search space $\mathfrak{U}$. We point out that this result requires that the coarse K-means solution $\mathfrak{M}$, which Algorithm 10 uses to construct the search space $\mathfrak{U}$, contains (at most) K mean vectors.

Claim 10.8 (existence of an approximation in $\mathfrak{U}$)**.** *There exist means $C \subset \mathfrak{U}$ with $|C| \leq K$ and*

$$\phi_X^{(r)}(C) \leq \left(1 + \frac{\epsilon}{2}\right) \phi_{(X,K,r)}^{OPT} .$$

Proof. Consider optimal means $O \subset \mathbb{R}^D$ with $|O| = K$ where $\phi_X^{(r)}(O) = \phi_{(X,K,r)}^{OPT}$. From Theorem 6.3, we know that there exists $O' = (\mu_l)_{l \in [L]} \subseteq O$, $L := |C| \leq K$, such that

$$\phi_X^{(r)}(O') \leq \left(1 + \frac{\epsilon}{2}\right) \phi_{(X,K,r)}^{OPT} \tag{10.2}$$

and such that O' induces an r-fuzzy clustering $P = (p_{nl})_{n \in [N], l \in [L]}$ of X where

$$\forall l \in [L] : \ \exists n \in [N] : \ p_{nl} \geq \frac{\epsilon/2}{2\mathbf{i}_r K^2} = \frac{\epsilon}{4\mathbf{i}_r K^2} . \tag{10.3}$$

Towards a contradiction, assume that for all solutions $C \subset \mathfrak{U}$ with $|C| \leq K$ and $\phi_X^{(r)}(C) \leq \left(1 + \frac{\epsilon}{2}\right) \phi_{(X,K,r)}^{OPT}$ we have $C \setminus \mathfrak{U} \neq \emptyset$.

Then, due to (10.2), there must exist a mean vector $\mu_k \in O' \setminus \mathfrak{U}$. More precisely, we have $\mu_k \notin \mathfrak{U} = \mathfrak{U}(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ where $\mathfrak{R} = \sqrt{\mathrm{km}_X(\mathfrak{M}) / (\alpha \mathbf{w}(X))}$ and

$$\mathfrak{E} \geq \left\lfloor \frac{1}{2} \log\left(9 \cdot \alpha \cdot r \left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathbf{w}(X)}{\mathtt{w}_{\min}^{(X)}}\right) \right\rfloor .$$

With Lemma 9.10, we can conclude that

$$\forall n \in [N]\colon \left\|x_n - \mu_k\right\|_2^2 \geq 2 \cdot r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1} \cdot \frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}} \,. \tag{10.4}$$

Hence,

$$\begin{aligned}
\left(1+\frac{\epsilon}{2}\right)\phi_{(X,K,r)}^{OPT} &\geq \phi_X^{(r)}(O') && \text{(Equation (10.2))}\\
&= \phi_X^{(r)}(O', P) && (P \text{ induced by } O')\\
&\geq \sum_{n=1}^{N} w_n r(p_{nk}) \left\|x_n - \mu_k\right\|_2^2 \\
&\geq \mathbf{w}\left(A_k^{(X,r(P))}\right) \cdot 2 \cdot r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1} \frac{\mathrm{km}_X(\mathfrak{M})}{\mathtt{w}_{\min}^{(X)}} \,. && \text{(Equation (10.4))}
\end{aligned}$$

Consequently,

$$\mathbf{w}\left(A_k^{(X,r(P))}\right) \leq \frac{\left(1+\frac{\epsilon}{2}\right)\phi_{(X,K,r)}^{OPT} \cdot r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X)}}{2\,\mathrm{km}_X(\mathfrak{M})} \,.$$

Now we exploit the fact that $|\mathfrak{M}| = K$. With Lemma 6.1, we can conclude that

$$\phi_{(X,K,r)}^{OPT} \leq \mathrm{km}_X(\mathfrak{M}) \,.$$

Besides that, $\left(1+\frac{\epsilon}{2}\right)/2 = \frac{1}{2} + \frac{\epsilon}{4} < 1$. Hence,

$$\mathbf{w}\left(A_k^{(X,R)}\right) = \sum_{n=1}^{N} r(p_{nk}) w_n < r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right) \cdot \mathtt{w}_{\min}^{(X)} \,.$$

With the properties of a fuzzifier function, it follows that $p_{nk} < \frac{\epsilon}{4\mathbf{i}_r K^2}$ for all $n \in [N]$, which contradicts (10.3). □

Replacing a Solution in $\mathfrak{U}$ by a Solution in $\mathfrak{G}$

Next, we show that representatives $\mathfrak{g}(C)$ perform similar to the corresponding means $C \subseteq \mathfrak{U}$ from the search space. The following result is an analogon of Lemma 5.8 from (Chen, 2009, p. 936).

Claim 10.9 (representatives suffice)**.** *For all $C = (\mu_k)_{k\in[K]} \subseteq \mathfrak{U}$ and their representatives $\mathfrak{g}(C) := \left(\mathfrak{g}(\mu_k)\right)_{k\in[K]}$, we have*

$$\left|\phi_X^{(r)}(C) - \phi_X^{(r)}(\mathfrak{g}(C))\right| \leq \frac{36\tilde{\epsilon}(2+\alpha)}{\mathbf{c}_r(K)} \cdot \phi_X^{(r)}(C) \,.$$

Proof. As r is [0,1]-reducing, Corollary 9.16 gives

$$\left|\phi_X^{(r)}(C) - \phi_X^{(r)}(\mathfrak{g}(C))\right| \leq 36\tilde{\epsilon}\left(\mathrm{km}_X(C) + \mathrm{km}_X(\mathfrak{M}) + \mathbf{w}(X)\mathfrak{R}^2\right) \,.$$

Recall from Lemma 6.1 that

$$\mathrm{km}_X(C) \leq \frac{1}{\mathbf{c}_r(K)} \phi_X^{(r)}(C) \,.$$

Recall that, by definition, $\mathbf{c}_r(K) \leq 1$. Moreover, recall from Claim 10.7 that $\mathfrak{M}$ is an α-approximation to the K-means problem. Hence,

$$\mathrm{km}_X(\mathfrak{M}) \leq \alpha\,\mathrm{km}_{(X,K)}^{OPT} \leq \alpha\,\mathrm{km}_X(C) \leq \alpha \frac{1}{\mathbf{c}_r(K)} \phi_X^{(r)}(C)$$

Besides that, with Lemma 6.1, we can conclude that

$$\mathbf{w}(X)\mathfrak{R}^2 = \frac{\mathrm{km}_X(\mathfrak{M})}{\alpha} \leq \frac{1}{\mathbf{c}_r(K)} \frac{\mathrm{km}_X(\mathfrak{M})}{\alpha} \leq \frac{1}{\mathbf{c}_r(K)} \phi_X^{(r)}(C) \,. \qquad (\alpha \geq 1)$$

A combination of these inequalities yields the claim. □

Existence of a Good Solution in $\mathfrak{G}$

Now we can conclude that there are good representatives in $\mathfrak{G}$. In other words, we show that, in order to find a good approximation, it suffices to evaluate all possible solutions $(\tilde{\mu}_k)_{k\in[K]} \subseteq \mathfrak{G}$.

Claim 10.10 (existence of an approximation in $\mathfrak{G}$)**.** *There exist means $C \subseteq \mathfrak{G}$ with size $|C| \leq K$ such that*

$$\phi_X^{(r)}(C) \leq (1+\epsilon)\phi_{(X,K,r)}^{OPT} .$$

Proof. From Claim 10.8, we know that there exist means $C = \{\mu_l\}_{l\in[L]} \subset \mathfrak{U}$ with $L \leq K$ and

$$\phi_X^{(r)}(C) \leq \left(1+\frac{\epsilon}{2}\right)\phi_{(X,K,r)}^{OPT} .$$

From Claim 10.9 we know that

$$\phi_X^{(r)}(\mathfrak{g}(C)) \leq \left(1+\frac{36\tilde{\epsilon}(2+\alpha)}{\mathbf{c}_r(K)}\right)\cdot\phi_X^{(r)}(C) .$$

By combining these inequalities, we obtain

$$\begin{aligned}
\phi_X^{(r)}(\mathfrak{g}(C)) &\leq \left(1+\frac{36\tilde{\epsilon}(2+\alpha)}{\mathbf{c}_r(K)}\right)\left(1+\frac{\epsilon}{2}\right)\phi_{(X,K,r)}^{OPT} \\
&\leq \left(1+\frac{\epsilon}{2}+\frac{72\tilde{\epsilon}(2+\alpha)}{\mathbf{c}_r(K)}\right)\phi_{(X,K,r)}^{OPT} && (\epsilon \leq 1) \\
&\leq (1+\epsilon)\phi_{(X,K,r)}^{OPT} , && (\tilde{\epsilon} = \epsilon\cdot\tfrac{\mathbf{c}_r(K)}{72(2+\alpha)})
\end{aligned}$$

which yields the claim. □

This last claim shows that Algorithm 10 returns a solution satisfying the desired approximation guarantee.

Size of $\mathfrak{G}$ and Runtime

It remains to analyse the size of $\mathfrak{G}$ and the algorithms' runtime. As in Corollary 10.4, let

$$H_{(r,K,\epsilon)} := \max\left\{\log\left(r\left(\frac{\epsilon}{4\mathbf{i}_r K^2}\right)^{-1}\right),\ \log\left(\mathbf{c}_r(K)^{-1}\right),\ 1\right\}.$$

Claim 10.11 (size)**.** *The discrete search space $\mathfrak{G}$ has the size*

$$|\mathfrak{G}| \in \mathcal{O}\left(\log\left(|X|\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\cdot K\cdot\left(\frac{2}{\epsilon}\right)^{D}\cdot H_{(r,K,\epsilon)}\right) .$$

Proof. Recall Lemma 9.17, plug in the parameters defined in Algorithm 10, and recall that $\alpha \in \mathcal{O}(1)$ and $|\mathfrak{M}| = K$. Note that $\mathbf{w}(X)/\mathtt{w}_{\min}^{(X)} \leq |X|\cdot\mathtt{w}_{\max}^{(X)}/\mathtt{w}_{\min}^{(X)}$. □

Claim 10.12 (runtime of Algorithm 10)**.** *The overall runtime of Algorithm 10 is*

$$\left(|X|\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\cdot\log\left(|X|\frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)^{K}\cdot 2^{\mathcal{O}(\log(K)KD\log(1/\epsilon))}\cdot H_{(r,K,\epsilon)}^{K}\cdot\mathbf{t}_r(K) .$$

Proof. First, consider our execution of the algorithm of Matoušek (2000). We apply it to the data set $\hat{X}$, which has the size $\left|\hat{X}\right| \leq 2|X|\mathtt{w}_{\max}^{(X)}/\mathtt{w}_{\min}^{(X)}$ (see Observation 10.6). We apply it with a constant precision $\epsilon \in (0,1)$. Hence, we need time

$$\mathcal{O}\left(\left|\hat{X}\right| \cdot \log\left(\left|\hat{X}\right|\right)^K 2^{K^2 D}\right) \subseteq \mathcal{O}\left(\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right) \cdot \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)^K 2^{K^2 D}\right) .$$

Next, consider the computation of the discrete search space. With Lemma 9.18 we can conclude that the time needed to compute $\mathfrak{G}$ is

$$\begin{aligned}\mathcal{O}\left(|\mathfrak{M}| \cdot \mathfrak{E} \cdot (2/\epsilon)^{3D}\right) &\subseteq \mathcal{O}\left(K^2 \cdot \left(\log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\right) \cdot H_{(r,K,\epsilon)} \cdot (2/\epsilon)^{3D}\right) \\ &\subseteq \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right) \cdot H_{(r,K,\epsilon)} \cdot 2^{\mathcal{O}(D \log(1/\epsilon) \log(K))} .\end{aligned}$$

Finally, consider the exhaustive search through the discrete search space $\mathfrak{G}$. There are $|\mathfrak{G}|^K$ possible combinations of K elements from $\mathfrak{G}$. Evaluating the cost $\phi_X^{(r)}(C)$ of a set $C \subseteq \mathbb{R}^D$ with $|C| = K$ needs time $\mathcal{O}(|X| D \mathbf{t}_r(K))$ (Assumption 5.19). With the help of the previous claim (and the fact that $\mathbf{w}(X) \leq |X| \cdot \mathtt{w}_{\max}^{(X)}$), we can conclude that the K representatives that incur the smallest cost can be determined in time

$$\begin{aligned}\mathcal{O}\left(|X| D \mathbf{t}_r(K) |\mathfrak{G}|^K\right) &\subseteq \mathcal{O}\left(|X| D \mathbf{t}_r(K) \left(\log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right) H_{(r,K,\epsilon)} \cdot K \left(\frac{2}{\epsilon}\right)^{-D}\right)^K\right) \\ &= \mathcal{O}\left(|X| D \left(\log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\right)^K \cdot H_{(r,K,\epsilon)}^K \cdot K^K \mathbf{t}_r(K) \left(\frac{2}{\epsilon}\right)^{-KD}\right) \\ &\subseteq |X| \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)^K \cdot 2^{\mathcal{O}(\log(K) KD \log(1/\epsilon))} \cdot H_{(r,K,\epsilon)}^K \cdot \mathbf{t}_r(K) .\end{aligned}$$

A combination of these bounds yields the claim. □

Claim 10.13 (runtime of algorithm from Corollary 10.5)**.** *The overall runtime of the algorithm from Corollary 10.5 is*

$$|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} KD + |X| \cdot \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)^{\mathcal{O}(K)} \cdot 2^{\mathcal{O}(\log(K) KD \log(1/\epsilon))} \cdot H_{(r,K,\epsilon)}^K \cdot \mathbf{t}_r(K) .$$

Proof. This proof is similar to the proof of Claim 10.12. Instead of the algorithm by Matoušek (2000), we apply the algorithm by Aggarwal et al. (2009) to the data set $\hat{X}$, which has size $\left|\hat{X}\right| \leq 2|X|\mathtt{w}_{\max}^{(X)}/\mathtt{w}_{\min}^{(X)}$. Executing this algorithm needs time

$$\mathcal{O}\left(\left|\hat{X}\right| KD + \operatorname{poly}\left(K, \log\left(\left|\hat{X}\right|\right)\right)\right) \subseteq \mathcal{O}\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}} KD + \operatorname{poly}\left(K, \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)\right)\right) .$$

The remaining steps of the algorithm from Corollary 10.5 coincide with Algorithm 10. From the proof of Claim 10.12, we already know that these steps need time

$$|X| \log\left(|X| \frac{\mathtt{w}_{\max}^{(X)}}{\mathtt{w}_{\min}^{(X)}}\right)^K \cdot 2^{\mathcal{O}(\log(K) KD \log(1/\epsilon))} \cdot H_{(r,K,\epsilon)}^K \cdot \mathbf{t}_r(K) .$$

A combination of these bounds yields the claim. □

Chapter 11

Dimension Reduction

A dimension reduction technique can be used to speed up an algorithm whose runtime crucially depends on the dimension of the given data set. The main idea is to map the given high-dimensional data set X_D to a low-dimensional representation X_d, feed this low-dimensional representation X_d to the algorithm, and then translate its output back to a solution with respect to X_D. More precisely, consider a high-dimensional data set $X_D = ((x_n, w_n))_{n\in[N]}$ with points $x_n \in \mathbb{R}^D$. We apply a dimension reduction technique to map each point $x_n \in \mathbb{R}^D$ from X_D to a point $\pi(x_n) \in \mathbb{R}^d$ where $d \ll D$. Then we solve our clustering problem on the resulting lower-dimensional data set $X_d = ((\pi(x_n), w_n))_{n\in[N]}$. Finally, we need to translate the solution back to a solution for the original data set X_D. Since the number of points is the same in both data sets, we can simply take the (soft) clustering P from the solution and compute the representatives induced by P with respect to X_D (instead of X_d). However, it depends on the mapping π whether the quality of the solution is preserved.

There are two kinds of provably accurate dimensionality reduction techniques known for K-means clustering: First, there are methods based on random projections, which are called Johnson-Lindenstrauss lemmata. They not only ensure that the quality of a K-means solution is preserved by the reduction; they also ensure that the pairwise Euclidean distances between the points are preserved up to a small factor. Clearly, this is useful not only in the context of the K-means problem, but also for the r-fuzzy K-means problem. Second, there is a spectral method known as principal component analysis (PCA). It is well known that this method is closely related to the K-means problem.

In this chapter, we show that the Johnson-Lindenstrauss lemma is useful in the context of r-fuzzy K-means. Moreover, we discuss the use of a PCA for the r-fuzzy K-means problem.

Overview. In Section 11.1, we describe the Johnson-Lindenstrauss lemma, give an overview of related work, and utilize the lemma for the r-fuzzy K-means problem. In Section 11.2, we briefly discuss spectral methods.

Publication. The results that we present Section 11.1 are generalizations of the results from (Blömer et al., 2016, Corollary 1, Lemma 3).

11.1 The Johnson Lindenstrauss Lemma

Johnson and Lindenstrauss (1984) showed that N points in a D-dimensional Euclidean space can be mapped down to a $\tilde{D} = \mathcal{O}(\log(N)/\epsilon^2)$ dimensional space while preserving the pairwise distances between the points by a factor $1 \pm \epsilon$. Our formulation of their result follows Dasgupta and Gupta (2003):

Lemma 11.1. *Let $X \subset \mathbb{R}^D$ with $|X| = N$ and $\epsilon \in [0,1]$. There is a linear map $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ with $\bar{D} = \mathcal{O}(\log(N)/\epsilon^2)$ such that for all $x, y \in X$ we have*

$$(1-\epsilon)\|x-y\|_2^2 \leq \|\pi(x)-\pi(y)\|_2^2 \leq (1+\epsilon)\|x-y\|_2^2 .$$

Moreover, there is a randomized algorithm that, given $X \subset \mathbb{R}^D$ and $\epsilon \in [0,1]$, finds such a map, with constant probability, and needs time $\mathcal{O}(D\log(N)/\epsilon^2)$.

In particular, the dimension $\bar{D}$ is *independent* from the dimension D of the original points. This means that the linear map preserves the pairwise distances between the N points to $\bar{D} = \mathcal{O}(\log(N)/\epsilon^2)$ dimensions, *regardless* of how high the given dimension D is.

11.1.1 Related Work

Recently, Larsen and Nelson (2016) showed that the bound on the dimension $\bar{D}$ given in the Johnson-Lindenstrauss lemma is tight. That is, every *linear* mapping $\pi : \mathbb{R}^D \to \bar{D}$ that satisfies the properties given in the lemma also satisfies $\bar{D} = \Omega(\min\{D, \log(N)/\epsilon^2\})$. However, this does not prove that we cannot map a data set to an even smaller dimension that preserves the r-fuzzy K-means cost (of a transferred solution).

The algorithm referred to in Lemma 11.1 is not very practical. Consequently, there has been a lot of work on developing more efficient algorithms: For instance, Ailon and Chazelle (2006) introduced the fast Johnson-Lindenstrauss transform which makes use of a sparse projection matrix with a randomized Fourier transform. This method has been repeatedly improved since then. For more information, see Ailon and Liberty (2013) and Kane and Nelson (2014), for instance.

11.1.2 Main Result

The key ingredient that enables us to apply the Johnson-Lindenstrauss lemma to the r-fuzzy K-means problem is the same as for the K-means problem: It is the observation that the cost can be expressed as a weighted sum of pairwise distances between points (Corollary 2.23).

Lemma 11.2. *Given $X \in \mathrm{Dom}(\mathbb{R}^D, \mathbb{R}_+)$ and $\epsilon \in [0,1]$, the algorithm from Lemma 11.1 computes a linear map $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ with $\bar{D} = \mathcal{O}(\log(|X|)/\epsilon^2)$ such that, with constant probability, the following property is satisfied:*

For all fuzzifier functions $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and soft clusterings P of X, we have

$$\phi^{(r)}_{\pi(X)}(P) \in [1 \pm \epsilon]\phi^{(r)}_X(P) .$$

Proof. Fix arbitrary $P = (p_{nk})_{n\in[N],k\in[K]} \in \Delta_{N,K-1}$ and $X = ((x_n, w_n))_{n\in[N]}$. First, observe that we map each point x_n to some $\pi(x_n)$, but we do not change its weight w_n. Thus, for all clusters with $\mathbf{w}\left(A_k^{(X,r(P))}\right) = 0$, we have $\mathbf{d}\left(A_k^{(X,r(P))}\right) = \mathbf{d}\left(A_k^{(\pi(X),r(P))}\right) = 0$. So, without loss of generality, we can assume that the cluster weights are strictly larger than 0. Then,

$$\begin{aligned}
\phi^{(r)}_{\pi(X)}(P) &= \sum_{k=1}^{K} \frac{\sum_{n=1}^{N}\sum_{m<n} r(p_{nk})r(p_{mk})w_n w_m \|\pi(x_n)-\pi(x_m)\|_2^2}{\sum_{n=1}^{N} r(p_{nk})w_n} && \text{(Corollary 2.23)} \\
&\in [1\pm\epsilon]\cdot \sum_{k=1}^{K} \frac{\sum_{n=1}^{N}\sum_{m<n} r(p_{nk})r(p_{mk})w_n w_m \|x_n-x_m\|_2^2}{\sum_{n=1}^{N} r(p_{nk})w_n} && \text{(Lemma 11.1)} \\
&= [1\pm\epsilon]\cdot \phi^{(r)}_X(P) .
\end{aligned}$$

□

That is, this dimension reduction technique preserves the r-fuzzy K-means cost of *each* soft clustering up to a small factor. Therefore, the quality of a solution is preserved:

Corollary 11.3. *Given $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ and $\frac{\epsilon}{3} \in (0, 1/3]$, the algorithm from Lemma 11.1 computes a linear map $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ with $\bar{D} = \mathcal{O}(\log(|X|)/\epsilon^2)$ such that, with constant probability, the following property is satisfied:*

For all fuzzifier functions $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and soft clusterings P of X with

$$\phi^{(r)}_{\pi(X)}(P) \leq \alpha \phi^{OPT}_{(\pi(X),K,r)}$$

it holds that

$$\phi^{(r)}_{X}(P) \leq \alpha(1+\epsilon)\phi^{OPT}_{(X,K,m)} \,.$$

Proof. Fix an arbitrary fuzzifier function r. Consider a soft K-clustering P of X with $\phi^{(r)}_{\pi(X)}(P) \leq \alpha \phi^{OPT}_{(\pi(X),K,r)}$. With Lemma 11.2, we can conclude

$$\phi^{(r)}_{X}(P) \leq \frac{1}{1-\epsilon/3}\phi^{(r)}_{\pi(X)}(P) \leq \alpha \frac{1}{1-\epsilon/3}\phi^{OPT}_{(\pi(X),K,r)} \,.$$

Let P^{OPT}_X be a soft K-clustering of X with $\phi^{(r)}_X(P^{OPT}_X) = \phi^{OPT}_{(X,K,r)}$. With Lemma 11.2, we can conclude

$$\phi^{OPT}_{(\pi(X),K,r)} \leq \phi^{(r)}_{\pi(X)}(P^{OPT}_X) \leq (1+\epsilon/3)\phi^{(r)}_X(P^{OPT}_X) = (1+\epsilon/3)\phi^{OPT}_{(X,K,r)} \,.$$

A combination of these inequalities gives

$$\phi^{(r)}_{X}(P) \leq \alpha \frac{1+\epsilon/3}{1-\epsilon/3}\phi^{OPT}_{(X,K,m)} \leq \alpha(1+\epsilon)\phi^{OPT}_{(X,K,m)} \,,$$

where we use that $\frac{1+\epsilon/3}{1-\epsilon/3} = 1 + 2\frac{\epsilon/3}{1-\epsilon/3} = 1 + 2\frac{\epsilon}{3-\epsilon} \leq 1+\epsilon$. This yields the claim. □

11.1.3 Application (Algorithm 11)

With the help of the Johnson-Lindenstrauss dimension reduction technique, we can speed up the algorithm from Corollary 10.5. More precisely, we get rid of the runtimes' exponential dependence on the dimension of the data points. A drawback is that the dependence on the number of clusters K becomes stronger.

In the following, we first state the general result and then simplify it with respect to special fuzzifier functions.

Algorithm 11 Combination with Dimension Reduction

Require: $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, and $\epsilon \in (0,1]$

1: Apply the algorithm from Corollary 11.3 to X and $\epsilon/4$ (instead of ϵ) to compute a map $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ with $\bar{D} = \mathcal{O}(\log(|X|)/\epsilon^2)$.
2: $Y := ((\pi(x_n), w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^{\bar{D}}, \mathbb{R}_+\right)$.
3: Apply our randomized variant of Algorithm 10 from Corollary 10.4 to Y, K, r, $\mathbf{c}_r$, $\mathbf{i}_r(K)$, and $\epsilon/4$ to compute a solution $C \subseteq \mathbb{R}^D$
4: **return** C

Corollary 11.4. *Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, $K \in \mathbb{N}$, a fuzzifier $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is $\mathbf{i}_r$-increase-bounded, $\mathbf{c}_r$-contribution-bounded, and $[0,1]$-reducing, the values $\mathbf{c}_r(K) \in (0,1]$ and $\mathbf{i}_r \in [1,\infty)$, and some $\epsilon \in (0,1]$, Algorithm 11 computes means $C \subseteq \mathbb{R}^D$, $|C| = K$, such that with constant probability*

$$\phi^{(r)}_X(C) \leq (1+\epsilon)\phi^{OPT}_{(X,K,r)} \,.$$

The algorithms' runtime is bounded by

$$D \cdot \frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}} \cdot |X|^{\mathcal{O}(K\log(K)\log(1/\epsilon)/\epsilon^2)} \cdot \log\left(\frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}\right)^{\mathcal{O}(K)} \cdot H^K_{(r,K,\epsilon)} \cdot \mathbf{t}_r(K) \,,$$

where

$$H_{(r,K,\epsilon)} = \max\left\{\log\left(r\left(\frac{\epsilon}{16\mathbf{i}_r K^2}\right)^{-1}\right),\ \log\left(\mathbf{c}_r(K)^{-1}\right),\ 1\right\} .$$

Proof. From Corollary 10.4, we know that C is a $(1+\epsilon/4)$-approximation to the r-fuzzy K-means problem with respect to Y. With Corollary 11.3 we can conclude that, with constant probability, C is a $(1+\epsilon/4)^2$-approximation to the r-fuzzy K-means problem with respect to X, where $(1+\epsilon/4)^2 \leq (1+\epsilon)$ (Lemma A.1).

According to Corollary 11.3, the computation of the mapping $\pi : \mathbb{R}^D \to \mathbb{R}^{\bar{D}}$ needs time $\mathcal{O}(D \cdot \log(|X|/\epsilon^2)$ and

$$\bar{D} = \mathcal{O}(\log(|X|)/\epsilon^2) . \tag{11.1}$$

So applying the linear mapping to each of the given points needs time

$$\mathcal{O}(|X| \cdot D \cdot \bar{D}) \subseteq \mathcal{O}\left(|X| \log(|X|) \cdot D/\epsilon^2\right) .$$

Observe that $|Y| = |X|$, $\mathtt{w}^{(Y)}_{\max} = \mathtt{w}^{(X)}_{\max}$, and $\mathtt{w}^{(Y)}_{\min} = \mathtt{w}^{(X)}_{\min}$. Hence, our application of the algorithm from Corollary 10.4 needs time

$$|X| \frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}} K\bar{D} + |X| \log\left(|X| \frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}\right)^{\mathcal{O}(K)} \cdot 2^{\mathcal{O}\left(\log(K) K \bar{D} \log(1/\epsilon)\right)} \cdot H^K_{(r,K,\epsilon)} \cdot \mathbf{t}_r(K) .$$

where $H_{(r,K,\epsilon)}$ is defined as in the claim (note that, in comparison to the definition of $H_{(r,K,\epsilon)}$ in Corollary 10.4, we replaced ϵ by $\epsilon/4$) and

$$2^{\mathcal{O}(K \log(K) \bar{D} \log(1/\epsilon))} \subseteq 2^{\mathcal{O}(K \log(K) \log(|X|) \log(1/\epsilon)/\epsilon^2)} \subseteq |X|^{\mathcal{O}(K \log(K) \log(1/\epsilon)/\epsilon^2)}$$

due to (11.1). Combining these bounds yields the claim. □

As already pointed out, in contrast to the runtime from Corollary 10.4, the runtime from Corollary 11.4 does not have an exponential dependence on the dimension of the data points. The dependence on the number of clusters K becomes stronger, though.

Special Case. Consider the classical fuzzy K-means problem with fuzzifier value $m \in (1,\infty)$. Recall from Section 5.3.2 that we can set $\mathbf{c}_{\mathrm{p}_m}(K) = K^{m-1}$, $\mathbf{i}_{\mathrm{p}_m} = 4m$, and $\mathbf{t}_{\mathrm{p}_m}(K) = \Theta(K)$. Hence, $H_{(\mathrm{p}_m,K,\epsilon)} = \max\left\{m \log\left(\frac{64mK^2}{\epsilon}\right),\ (m-1)\log(K),\ 1\right\} \in \mathcal{O}(m^2\epsilon^{-1}K)$. Consequently, $H^K_{(\mathrm{p}_m,K,\epsilon)} \in 2^{\mathcal{O}(\log(m)\log(1/\epsilon)\log(K))}$. We can simplify the runtime bound to

$$\begin{aligned} &D \cdot \frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}} \cdot |X|^{\mathcal{O}(K\log(K)\log(1/\epsilon)\log(m)/\epsilon^2)} \log\left(\frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}^K\right) \\ \subseteq &D \cdot \frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}} \cdot |X|^{\mathcal{O}(K^2\epsilon^{-3}m)} \log\left(\frac{\mathtt{w}^{(X)}_{\max}}{\mathtt{w}^{(X)}_{\min}}^K\right) . \end{aligned}$$

In particular, for unweighted data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, the runtime is simply

$$D \cdot |X|^{\mathcal{O}(K^2\epsilon^{-3}m)} .$$

For an overview of all our approximation algorithms, we refer to Chapter 13.

11.2 Principal Component Analysis

Simply speaking, a principal component analysis (PCA) is a dimension reduction technique that is based on a singular value decomposition of a matrix containing (data) points.

In a Nutshell. First, let us briefly review the PCA approach. For more information, we refer to (Bishop, 2006, pp.561) and (Kannan and Vempala, 2009). Assume we are given a centered version of an unweighted data set $X = ((x_n, 1))_{n\in[N]}$. That is, we assume that the mean of the whole data set is the zero vector ($\mathbf{m}(X) = 0_D$) (if it is not, just shift all points). Consider the matrix $M_X = (x_1 \dots x_N) \in \mathbb{R}^{D\times N}$ that contains all points as column vectors. Via singular value decomposition, one can find the appropriate D vectors that span the point matrix M_X. That is, each point x_n can be expressed as a linear combination of these points. The goal of a PCA is to identify the most meaningful d vectors that describe the point matrix M_X best. Given these meaningful vectors, one reduces the dimension by performing a change of the basis. That is, the d vectors are chosen such that this change of the basis results in an error as small as possible, where by error we refer to the sum of the squared Euclidean distances between the original data points and their representations in the d-dimensional subspace (Bishop, 2006, pp. 563). This corresponds to choosing the vectors that point into the direction where the variance of the data set is largest. In other words, one chooses the d vectors whose corresponding singular values are largest.

Connection to K-Means. Clearly, variances are essential in the PCA approach (Shlens, 2003): Those directions in which the variance of the data set is large are considered the most meaningful by this approach. This statistic is important in K-means clustering as well since the cost of each hard cluster coincides with the (normalized) variance of this hard cluster. The connection is even much closer than this observation: Ding and He (2004) show that K-means and PCA maximize the same objective function but with respect to different constraints. Drineas et al. (2004) show that one can utilize PCA for an approximation algorithm for the K-means problem (which they refer to as "the discrete clustering problem"). Moreover, another way to describe the relation between K means and PCA is to consider them both as variations of a matrix factorization problem, as noted by Watt et al. (2016) for instance. Besides that, note that there are also results showing that a PCA is related to (spherical) Gaussian mixture modelling (Vempala and Wang, 2004). To sum up, there is an inherently close relation between the K-means problem and PCA.

Connection to Fuzzy K-Means? Now the question arises whether we can (provably) utilize a PCA for the r-fuzzy K-means problem. On the one hand, in Section 4.2, we already discussed the differences between the K-means problem and the fuzzy K-means problem. The r-fuzzy K-means cost is a sum of (un-normalized) variances of re-weighted versions of the given point set with weights that probably exhibit no useful property (regarding the variances of the original point set). On the other hand, in Chapter 6, we showed that there is a coarse relation between the K-means and fuzzy K-means objective function and between the respective notions of negligible clusters. However, the relation between the objective functions is so coarse that when we apply results from K-means directly, we eventually incur a very large approximation factor (cf. Section 7.1). It is still an open question how dimension reduction techniques that rely on singular value decomposition can be used or adapted (provably) properly.

"You will need to know the difference between Friday and a fried egg. It's quite a simple difference, but an important one."

Douglas Adams[1]

Chapter 12

Coresets

Assume that we are given a data set that contains far more data points than we want to process. A way to deal with such a data set is to summarize its data points in some way and then to process the summary instead of the data set. Given that a suitable summary can be computed fast enough, this approach can effectively speed up an algorithm whose runtime heavily depends on the number of data points. Whether a summary is suitable obviously depends on what we want to do with the data set. One can roughly distinguish two types of summaries: coresets and sketches. Simply speaking, a coreset can be used as a surrogate for the data set, whereas a sketch may take a completely different form than the original data set. For instance, a coreset is often just a (weighted) subset of the original data set, while a Gaussian mixture model that has been fitted to the data set might be a useful sketch. As the title says, in this chapter we focus on coresets.

To this end, we return to the results of Chen (2009) once again, who gave a construction of coresets for the K-means problem. We already analysed and refined some of his results in Chapter 9. In this chapter, we show that a slightly refined version of his construction actually yields a useful coreset for the r-fuzzy K-means problem.

Overview. In Section 12.1, we give a brief overview of work regarding coreset constructions. In Section 12.2, we sum up our main contributions of this chapter. In Section 12.3, we formally state our main result and describe the algorithm that constructs a coreset for the r-fuzzy K-means problem. Section 12.5 contains our analysis of this algorithm. Finally, in Section 12.4, we show that our algorithm can be used to enhance Algorithm 11, which we presented in Section 11.1.

Publication. In this chapter, we generalize and correct results and proofs from Blömer et al. (2017).

12.1 Related Work

An overview of the early work of coresets can be found in Bādoiu et al. (2002) and Agarwal et al. (2005). Har-Peled and Mazumdar (2004) present a deterministic construction of a coreset with a size linear in K, exponential in D and logarithmic in $|X|$, namely $\mathcal{O}(K\epsilon^{-D}\log(|X|))$. Later, Har-Peled and Kushal (2005) improved this result to a coreset with a size polynomial in K and exponential in D, but independent of $|X|$, namely $\mathcal{O}(K^3\epsilon^{-D-1})$. Leveraging this idea, Chen (2009) presented a randomized coreset construction. His construction does *not* have an exponential dependence on D, but again introduces a factor that is poly-logarithmic in $|X|$. More precisely, the size of this coreset is

[1] Source: Douglas Adams, The Salmon of Doubt

$\mathcal{O}(K\log(|X|)\epsilon^{-2}(DK\log(D/\epsilon)+K\log(K)+K\log(\log(|X|))))$. Feldman et al. (2013) presented a construction based on low rank approximation, which results in a coreset size completely independent of D, but logarithmic in $|X|$. Recently, Lucic et al. (2016) proposed a coreset construction for a large class of hard and soft clustering problems that are based on Bregman divergences.

12.2 Contribution

We show that a coreset for the r-fuzzy K-means problem can be constructed via a slightly refined version of an algorithm by Chen (2009). Furthermore, we show that this coreset construction is good enough to improve Algorithm 14, which we presented in Section 11.1. That is, the size of this coreset is small enough and it can be computed fast enough, so that we can use it to speed up the application of this algorithm. For an overview of all of our approximation algorithms we refer to Section 13.1.

12.3 Main Result

Let us start by formalizing our notion of coresets for the r-fuzzy K-means problem. As in Chapter 10, we denote vectors of means as solutions and use the following notation:

Notation 10.1 (short notation)**.** *For $M \subseteq \mathbb{R}^D$ and $K \in \mathbb{N}$, the set of solutions with exactly K mean vectors from M is $M^K := \{(\mu_1,\dots,\mu_K) \mid \forall k \in [K]\colon\ \mu_k \in M\}$. Analogously, we let $M^{\leq K} := \bigcup_{k\in[K]} \{(\mu_1,\dots,\mu_k) \mid \forall l \in [k]\colon\ \mu_l \in M\}$.*

Simply speaking, a coreset S of a data set X behaves like the data set in the sense that the quality of each possible solution is similar.

Definition 12.1 ((strong) coreset)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $K \in \mathbb{N}$, and $\epsilon \in [0,1]$. The data set $S \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$ is a (strong) ϵ-coreset of X for the r-fuzzy K-means problem if*

$$\forall C \in (\mathbb{R}^D)^{\leq K}\colon\ \phi_S^{(r)}(C) \in [1 \pm \epsilon]\phi_X^{(r)}(C)\ . \tag{12.1}$$

Usually, these sets are called (K,ϵ)-coresets. In the following, K is either clear from context or stated explicitly, as part of the term "r-fuzzy K-means problem". In the next sections, we use the optional prefix "strong" to distinguish this kind of coreset from "weaker" coresets.

As said before, with a refined version of the algorithm from (Chen, 2009), we can actually compute such coresets. In fact, the only difference between our Algorithm 12 and the algorithm from (Chen, 2009) is that we tuned the sample size Q the precision ϵ.

Theorem 12.2 (coreset)**.** *Given an unweighted data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, a fuzzifier function r that is $[0,1]$-reducing, $\mathbf{i}_r$-increase-bounded, and $\mathbf{c}_r$-contribution-bounded, the values $\mathbf{c}_r(K) \in (0,1]$ and $\mathbf{i}_r \in [1,\infty)$, $\epsilon/4 \in (0,1/4)$, and $\delta \in (0,1)$, Algorithm 12 computes a data set $S \in \mathrm{Dom}(\{x \mid (x,w) \in X\}, \mathbb{N})$ such that, with a probability of at least $1-\delta$, the set S is an ϵ-coreset of X for the r-fuzzy K-means problem.*

The data set S has the size

$$|S| \in \mathcal{O}\left(\log(|X|)\log\log(|X|)^2 \cdot K^3 \cdot D \cdot \epsilon^{-3} \cdot \log\log\left(r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)^{-1}\right) \cdot \mathbf{c}_r(K)^{-3} \cdot \log(\delta^{-1})\right)$$

and its weights satisfy

$$\frac{\mathrm{w}_{\max}^{(S)}}{\mathrm{w}_{\min}^{(S)}} \leq |X|\ .$$

Algorithm 12 Strong Coreset

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \geq 1$, $\epsilon \in (0,1)$, and $\delta \in (0,1)$

1: Apply the randomized algorithm from Aggarwal et al. (2009) which computes, with probability at least $1 - \delta/3$, an (α, β)-bicriteria approximation $\mathfrak{M} \subseteq \mathbb{R}^D$ for the K-means problem with respect to the unweighted data set X where $\alpha, \beta = \mathcal{O}(1)$.

2: Set

$$\tilde{\epsilon} := \frac{\epsilon \cdot \mathbf{c}_r(K)}{504\alpha},$$
$$\mathfrak{E}' := \frac{1}{2} \log\left(9\alpha |X| \cdot \frac{20}{\tilde{\epsilon}^2 r\left(\frac{\epsilon}{2 \cdot \mathbf{i}_r K^2}\right)}\right), \text{ and}$$
$$\gamma := \beta K \cdot (\mathfrak{E}' + 1) \cdot \left(\frac{1692}{\tilde{\epsilon}}\right)^D.$$

3: Let S be the output of Algorithm 13, given X, K, r, $\mathbf{c}_r(K)$, $\tilde{\epsilon}$, $\delta/3$, γ, α, β, and $\mathfrak{M}$.

4: **Return** S

Algorithm 13 Sampling from K-Means Clusters Divided into Rings (cf. Chen (2009))

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\epsilon \in (0,1)$, $\delta \in (0,1)$, $\gamma \in \mathbb{N}$, $\alpha, \beta \in [1, \infty)$, $\mathfrak{M} = (\mathfrak{m}_l)_{l \in [L]} \subseteq \mathbb{R}^D$ with $L = \lfloor \beta K \rfloor$

1: Let $A_1, \ldots, A_L \subset \mathbb{R}^D$ be an L-means partition of $\{x \mid (x,1) \in X\}$ induced by $\mathfrak{M}$.

2: Let

$$\mathfrak{E} := \left\lceil \frac{1}{2} \log(\alpha |X|) \right\rceil \quad \text{and} \quad \mathfrak{R} := \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{\alpha |X|}}. \tag{12.2}$$

3: For a sufficiently large constant Q_{const}, let

$$Q := Q_{const} \cdot \left(\frac{\alpha}{\epsilon \cdot \mathbf{c}_r(K)}\right)^2 \ln\left(\frac{4L\mathfrak{E}\gamma^K}{\delta}\right).$$

4: **for all** $l \in [L]$ and $j \in \{0, \ldots, \mathfrak{E}\}$ **do**

5: Let $\mathfrak{U}_{l,j}$ be the (l,j)-th ring defined by $(\mathfrak{E}, \mathfrak{R}, \mathfrak{M})$ (Definition 9.6).

6: $X_{l,j} := \mathfrak{U}_{l,j} \cap A_l \subseteq \mathbb{R}^D$

7: **if** $X_{l,j} \neq \emptyset$ **then**

8: **if** $\frac{|X_{l,j}|}{Q} \in \mathbb{N}$ **then**

9: $S_{l,j} := (s_i)_{i \in [Q]}$ where each s_i has been uniformly sampled from $X_{l,j}$.

10: $S^{\omega}_{l,j} = \left(\left(s_i, \frac{|X_{l,j}|}{Q}\right)\right)_{i \in [Q]}$.

11: **else**

12: $\hat{Q} := \left|X_{l,j}\right| - Q \cdot \left\lfloor \frac{|X_{l,j}|}{Q} \right\rfloor$

13: $T_{lj} := (t_i)_{i \in [\hat{Q}]} \subseteq X_{l,j}$ be an arbitrary vector containing $\hat{Q}$ points from $X_{l,j}$.

14: $T^{\omega}_{lj} := ((t_i, 1))_{i \in [Q]}$

15: $R_{lj} := (r_i)_{i \in [Q]}$ where each r_i has been uniformly sampled from $X_{l,j} \setminus T_{lj}$

16: $R^{\omega}_{lj} = \left(\left(r_i, \frac{|X_{l,j} \setminus T_{lj}|}{Q}\right)\right)_{i \in [Q]}$

17: $S_{l,j} := S_{l,j} \cup R_{lj}$

18: $S^{\omega}_{l,j} := T^{\omega}_{lj} \dot{\cup} R^{\omega}_{lj}$

19: **return** $S := \bigcup_{l \in [L],\ j \in \{0, \ldots, \mathfrak{E}\}} S^{\omega}_{l,j}$.

The algorithms' runtime is

$$\mathcal{O}\left(|X|DK\log\left(\delta^{-1}\right)+|S|\right) .$$

For the sake of clarity, in this theorem and in the remainder of this chapter, we ignore the case that an evaluation of loglog might return a value less than one.

Special Case. With respect to the classical fuzzy K-means problem, our result simplifies as follows. Recall from Section 5.3.2 that for the polynomial fuzzifier function p_m with $m \in (1,\infty)$, we can set $\mathbf{i}_{\mathrm{p}_m} = 4m$ and $\mathbf{c}_{\mathrm{p}_m}(K) = K^{m-1}$. Let the probability of success $1-\delta$ be some constant. Then we can bound the size of S by

$$\begin{aligned}|S| &\in \mathcal{O}\left(\log(|X|)\log\log(|X|)^2 \cdot K^3 \cdot D \cdot \epsilon^{-3} \cdot \log\left(m\log\left(\frac{8mK^2}{\epsilon}\right)\right)\cdot K^{3(m-1)}\right)\\ &\subseteq \mathcal{O}\left(\log(|X|)\log\log(|X|)^2 \cdot K^{3m+2} \cdot D \cdot \epsilon^{-4}\right) .\end{aligned}$$

Generalization to Weighted Data Sets ($\mathbb{N}$). We can also compute coresets for data sets with weights in $\mathbb{N}$: Given the weighted data set $X = ((x_n, w_n))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{N}\right)$, construct a data set X_w that contains w_n copies of $(x_n, 1)$, for each $n \in [N]$. Then apply Algorithm 12 to X_w instead of X. With Corollary 2.26 and Theorem 12.2, it is easy to see that the algorithm computes a coreset S of the weighted set X. However, as we apply the algorithm to a data set that contains $|X_w| = \mathbf{w}(X)$ points instead of $|X|$ points, the size of this coreset and the runtime of the algorithm increase accordingly.

12.4 Application (Algorithm 14)

In this section, we show that Theorem 12.2 constructs a sufficiently small coreset fast enough so that it pays off to compute the coreset first and then to apply Algorithm 11 to the coreset.

The straightforward combination of the algorithms, as described in Algorithm 14, yields the following performance:

Algorithm 14 Combination of Techniques

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\mathbf{i}_r \in [1,\infty)$, $\epsilon \in [0,1]$

1: Let $\tilde{\epsilon} := \epsilon/6$
2: Apply Algorithm 12 to $X, K, r, \mathbf{c}_r(K), \mathbf{i}_r, \tilde{\epsilon}/4$ and sufficiently small δ to compute a data set $S \in \mathrm{Dom}\left(\{x \mid (x,w) \in X\}, \mathbb{N}\right)$.
3: Apply Algorithm 11 to $X, K, r, \mathbf{c}_r(K), \mathbf{i}_r$ and $\tilde{\epsilon}$ to compute a vector of means C.
4: **Return** C

Corollary 12.3. *Given a data set $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, a fuzzifier $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is $\mathbf{i}_r$-increase-bounded, $\mathbf{c}_r$-contribution-bounded, and $[0,1]$-reducing, the values $\mathbf{c}_r(K) \in (0,1]$ and $\mathbf{i}_r \in [1,\infty)$, and some $\epsilon \in (0,1]$, Algorithm 11 computes means $C \subseteq \mathbb{R}^D$, $|C| = K$, such that with constant probability*

$$\phi_X^{(r)}(C) \leq (1+\epsilon)\phi_{(X,K,r)}^{OPT} .$$

The algorithms' runtime is bounded by

$$D \cdot |X| \cdot \left(\log(|X|) \cdot D \cdot \max\left\{\log\left(r\left(\frac{\epsilon}{96 \cdot \mathbf{i}_r K^2}\right)^{-1}\right), 1\right\} \cdot \mathbf{c}_r(K)^{-1}\right)^{\mathcal{O}(K\log(K)^2\log(1/\epsilon)^2/\epsilon^2)} \cdot \mathbf{t}_r(K) .$$

Proof. As in the algorithm, let $\tilde{\epsilon} := \epsilon/6$. First, we analyse the approximation guarantee. Assume that S is an $\tilde{\epsilon}$-coreset of X. From Theorem 12.2, we know that this holds true with constant probability. That is,

$$\forall C \in (\mathbb{R}^D)^{\leq K} : \ \phi_S^{(r)}(C) \in [1 \pm \tilde{\epsilon}] \cdot \phi_X^{(r)}(C) \ . \tag{12.3}$$

From Corollary 11.4, we know that, with constant probability,

$$\phi_S^{(r)}(C) \leq (1+\tilde{\epsilon}) \cdot \phi_{(S,K,r)}^{OPT} \ . \tag{12.4}$$

Fix some $C_X^{OPT} \in (\mathbb{R}^D)^{\leq K}$ where $\phi_X^{(m)}(C_X^{OPT}) = \phi_{(X,K,r)}^{OPT}$. A combination of these inequalities gives

$$\begin{aligned}
\phi_X^{(r)}(C) &\leq \frac{1}{1-\tilde{\epsilon}} \cdot \phi_S^{(r)}(C) && \text{(Equation (12.3))} \\
&\leq \frac{1+\tilde{\epsilon}}{1-\tilde{\epsilon}} \cdot \phi_{(S,K,r)}^{OPT} && \text{(Equation (12.4))} \\
&\leq \frac{1+\tilde{\epsilon}}{1-\tilde{\epsilon}} \cdot \phi_S^{(r)}\left(C_X^{OPT}\right) \\
&\leq \frac{(1+\tilde{\epsilon})^2}{1-\tilde{\epsilon}} \cdot \phi_X^{(r)}\left(C_X^{OPT}\right) && \text{(Equation (12.3))} \\
&= \frac{(1+\tilde{\epsilon})^2}{1-\tilde{\epsilon}} \cdot \phi_{(X,K,r)}^{OPT} \\
&\leq (1+\epsilon) \cdot \phi_{(X,K,r)}^{OPT} \ ,
\end{aligned}$$

where in the last inequality we use the fact that

$$\frac{(1+\tilde{\epsilon})^2}{1-\tilde{\epsilon}} \leq \frac{1+3\tilde{\epsilon}}{1-\tilde{\epsilon}} = 1 + \frac{4\tilde{\epsilon}}{1-\tilde{\epsilon}} = 1 + \frac{4\epsilon/6}{1-\tilde{\epsilon}/6} = 1 + \frac{4\epsilon}{6-\epsilon} \leq 1 + \frac{4}{5}\epsilon \leq 1+\epsilon \ .$$

Next, consider the runtime. From Corollary 11.4, we can know that the computation of C needs time

$$T_C := D \cdot \frac{\mathtt{w}_{\max}^{(S)}}{\mathtt{w}_{\min}^{(S)}} \cdot |S|^{\mathcal{O}(K \log(K) \log(1/\tilde{\epsilon})/\tilde{\epsilon}^2)} \log\left(\frac{\mathtt{w}_{\max}^{(S)}}{\mathtt{w}_{\min}^{(S)}}\right)^{\mathcal{O}(K)} \cdot H_{(r,K,\tilde{\epsilon})}^K \cdot \mathbf{t}_r(K) \ ,$$

where $\tilde{\epsilon} = \epsilon/6$ and

$$\begin{aligned}
H_{(r,K,\tilde{\epsilon})}^K &= \max\left\{\log\left(r\left(\frac{\tilde{\epsilon}}{16\mathbf{i}_r K^2}\right)^{-1}\right), \ \log\left(\mathbf{c}_r(K)^{-1}\right), \ 1\right\}^K \\
&= \max\left\{\log\left(r\left(\frac{\epsilon}{96\mathbf{i}_r K^2}\right)^{-1}\right), \ \log\left(\mathbf{c}_r(K)^{-1}\right), \ 1\right\}^K \ . && (\tilde{\epsilon} = \epsilon/6)
\end{aligned}$$

From Theorem 12.2 we know that

$$\frac{\mathtt{w}_{\max}^{(S)}}{\mathtt{w}_{\min}^{(S)}} \leq |X|$$

and (since δ is some constant)

$$\begin{aligned}
|S| &\in \mathcal{O}\left(\log(|X|)\log(\log(|X|))^2 \cdot K^3 \cdot D \cdot \epsilon^{-3} \cdot \log\log\left(r\left(\frac{\epsilon}{2 \cdot \mathbf{i}_r K^2}\right)^{-1}\right) \cdot \mathbf{c}_r(K)^{-3}\right) \\
&\subseteq \mathcal{O}\left(\left(\log(|X|) \cdot D \cdot \log\log\left(r\left(\frac{\epsilon}{2 \cdot \mathbf{i}_r K^2}\right)^{-1}\right) \cdot \mathbf{c}_r(K)^{-1}\right)^{\log(K)\log(1/\epsilon)}\right) \ .
\end{aligned}$$

Besides that, note that $r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)^{-1} \leq r\left(\frac{\epsilon}{96\cdot \mathbf{i}_r K^2}\right)^{-1}$ since r is strictly increasing. Hence,

$$\begin{aligned}&|S|^{\mathcal{O}(K\log(K)\log(1/\tilde{\epsilon})/\tilde{\epsilon}^2)}\\ \subseteq &\mathcal{O}\left(\log(|X|)\cdot D\cdot \log\log\left(r\left(\frac{\epsilon}{96\cdot \mathbf{i}_r K^2}\right)^{-1}\right)\cdot \mathbf{c}_r(K)^{-1}\right)^{\mathcal{O}\left(K\log(K)^2\log(1/\epsilon)^2/\epsilon^2\right)}.\end{aligned}$$

By combining all these bounds, we obtain

$$T_C \in D\,|X|\cdot\left(\log(|X|)\cdot D\cdot \max\left\{\log\left(r\left(\frac{\epsilon}{96\cdot \mathbf{i}_r K^2}\right)^{-1}\right),\,1\right\}\cdot \mathbf{c}_r(K)^{-1}\right)^{\mathcal{O}(K\log(K)^2\log(1/\epsilon)^2/\epsilon^2)} \mathbf{t}_r(K)\,.$$

From Theorem 12.2, we know that the time needed to compute S is $\mathcal{O}(|X|DK+|S|)$. This yields the claim.

□

Special Case. With respect to the classical fuzzy K-means problem, the result can be simplified as follows. Recall from Section 5.3.2 that for the fuzzifier function p_m with $m\in(1,\infty)$, we can set $\mathbf{i}_{\mathrm{p}_m}=4m$, $\mathbf{c}_{\mathrm{p}_m}(K)=K^{m-1}$, and $\mathbf{t}_{\mathrm{p}_m}(K)=\Theta(K)$. This means that

$$\log\left(\mathrm{p}_m\left(\frac{\epsilon}{96\mathbf{i}_r K^2}\right)^{-1}\right)\cdot \mathbf{c}_{\mathrm{p}_m}(K)^{-1} = m\cdot\log\left(\frac{96\cdot 4mK^2}{\epsilon}\right)\cdot K^{m-1} \in \mathcal{O}\left(m^2\cdot\epsilon^{-1}\cdot K^m\right)$$

Hence, the runtime bound becomes

$$\begin{aligned}&|X|\left(\log(|X|)\cdot D\cdot m^2\cdot\epsilon^{-1}\cdot K^m\right)^{\mathcal{O}(K\log(K)^2\log(1/\epsilon)^2/\epsilon^2)}\\ \subseteq\,&|X|\left(\log(|X|)\cdot D\right)^{\mathcal{O}(K\log(K)^3\cdot\log(1/\epsilon)^3/\epsilon^2\cdot m)}\\ \subseteq\,&|X|\left(\log(|X|)\cdot D\right)^{\mathcal{O}(K^2\epsilon^{-3}m)}.\end{aligned} \tag{12.5}$$

Comparison. For the sake of simplicity, let us just compare Algorithm 11 and Algorithm 14 with respect to the classical fuzzy K-means problem. From Section 11.1.3, we know that the runtime of Algorithm 11, given an unweighted data set $X\in\mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$, is bounded by

$$D\cdot|X|^{\mathcal{O}(K^2\log(1/\epsilon)\log(m)/\epsilon^2)} \subseteq D\cdot|X|^{\mathcal{O}(K^2\epsilon^{-3}m)}.$$

In comparison to this bound, the runtime bound from (12.5) is clearly preferable as its dependence on the number of points $|X|$ is weaker. A downside of our bound from (12.5) is that its dependence on the dimension D is worse. For an overview of all of our algorithms we refer to Section 13.1.

Remark. Note that we cannot improve the approximation algorithm from Theorem 8.9 by a applying it to our coreset S. The runtime of this algorithm strongly depends on the weights of the given data set S, namely, by a factor $(\mathrm{w}^{(S)}_{\max}/\mathrm{w}^{(S)}_{\min})^K$. Unfortunately, we can only guarantee that $\mathrm{w}^{(S)}_{\max}/\mathrm{w}^{(S)}_{\min}\leq|X|$, for the original input data set X.

12.5 Analysis

For our analysis we make use of various relaxed notions of coresets. In Section 12.5.1, we introduce these relaxed notions formally and explain why they are useful in our analysis. Then, in Section 12.5.2, we give a more detailed outline of our analysis.

12.5.1 The Key Ideas

A (strong) coreset guarantees that the coreset property holds for each possible solution. That is, for each solution, the cost with respect to the coreset differs from the cost with respect to the original data set by only a small factor.

Weak Coresets

A weak coreset guarantees the coreset property only for a certain set of solutions, which has to be an ϵ-approximate mean set.

Definition 12.4 (weak ϵ-coresets)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $\epsilon \in [0,1]$, and $K \in \mathbb{N}$.*

Consider a set of solutions $\Theta \subseteq (\mathbb{R}^D)^{\leq K}$ and a data set $S \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$. The tuple (S, Θ) is a weak ϵ-coreset of X for the r-fuzzy K-means problem if

$$\forall C \in \Theta : \ \phi_S^{(r)}(C) \in [1 \pm \epsilon] \cdot \phi_X^{(r)}(C)$$

and if Θ is an ϵ-approximate mean set of X for the r-fuzzy K-means problem.

In contrast to the definition of weak coresets for the K-means problem (Feldman et al., 2007), we consider elements C of a given set of solutions Θ instead of subsets of a set of candidate means. This is just a slight generalization that allows us to choose the set of solutions more precisely.

To utilize a weak coreset (S, Θ) of X, we need an approximation algorithm that solves the r-fuzzy K-means problem with respect to the data set S and the set of solutions Θ:

Observation 12.5 (the use of weak coresets)**.** *Assume we have an algorithm that can solve the r-fuzzy K-means approximation problem, restricted to a small set of solutions Θ, with respect to the small data set S in reasonable time. That is, it finds some solution $C_{alg} \in \Theta$ with $\phi_S^{(r)}(C_{alg}) \leq (1+\epsilon)\phi_S^{(r)}(C_{S,\Theta}^{opt})$ where $C_{S,\Theta}^{opt} \in \arg\min\left\{\phi_S^{(r)}(C') \,\middle|\, C' \in \Theta\right\}$.*

Assume that (S, Θ) is a weak coreset of the large data set X. This means that the coreset property holds for all solutions from Θ. Hence, we have $\phi_X^{(r)}(C_{alg}) \leq 1/(1-\epsilon)\phi_S^{(r)}(C_{alg})$. Moreover, it means that Θ is an approximate mean set. Hence, there exists a $C^* \in \Theta$ satisfying $\phi_X^{(r)}(C^*) \leq (1+\epsilon)\phi_{(X,K,r)}^{OPT}$.

Observe that $\phi_S^{(r)}(C_{S,\Theta}^{opt}) \leq \phi_S^{(r)}(C^*)$ due to the definition of $C_{S,\Theta}^{opt}$ and the fact that $C^* \in \Theta$. Besides that, observe that $\phi_S^{(r)}(C^*) \leq (1+\epsilon)\phi_X^{(r)}(C^*)$ due to the fact that the coreset property applies to $C^* \in \Theta$. Therefore, we have $\phi_S^{(r)}(C_{S,\Theta}^{opt}) \leq \phi_S^{(r)}(C^*) \leq (1+\epsilon)\phi_X^{(r)}(C^*)$.

By combining these observations, we obtain

$$\phi_X^{(r)}(C_{alg}) \leq \tfrac{1}{1-\epsilon}\phi_S^{(r)}(C_{alg}) \leq \tfrac{1+\epsilon}{1-\epsilon}\phi_S^{(r)}(C_{S,\Theta}^{opt}) \leq \tfrac{(1+\epsilon)^2}{1-\epsilon}\phi_X^{(r)}(C^*) \leq \tfrac{(1+\epsilon)^3}{1-\epsilon}\phi_{(X,K,r)}^{OPT} \ .$$

Solutions with Non-Negligible Clusters

The notion of a weak coreset is only useful if we can identify a suitable ϵ-approximate mean set Θ. For each (original) data set X, we consider a special set of solutions $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$:

Definition 12.6. *For each $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $\mathbf{i}_r$-increase-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $K \in \mathbb{N}$, and $\epsilon \in [0,1]$, we let*

$$\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X) := \left\{ C \in (\mathbb{R}^D)^{\leq K} \;\middle|\; \begin{array}{c} \text{There is an } r\text{-fuzzy clustering of } X \text{ induced by } C \\ \text{that has no } (\mathbf{i}_r, K, \epsilon)\text{-negligible clusters} \end{array} \right\} .$$

It is easy to see that this set of solutions is an ϵ-approximate mean set for X.

Lemma 12.7. *For all $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $\mathbf{i}_r$-increase-bounded fuzzifier functions r, $K \in \mathbb{N}$, and $\epsilon \in [0,1]$, $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}$ is an ϵ-approximate mean set of X for the r-fuzzy K-means problem.*

Proof. Theorem 6.3 holds for optimal solutions, in particular. This yields the claim. □

The benefit of the restriction to this specific set of solutions $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ is that it guarantees that we can make use of the notion of non-negligible clusters, which is the analogon of non-empty clusters in a K-means hard clustering (cf. Section 6.2). This helps us to transfer the proof of Chen (2009), who showed that Algorithm 12 computes a (strong) coreset for the K-means problem, to the r-fuzzy K-means problem. More precisely, we will follow his line of arguments to show that Algorithm 12 computes a set S such that $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a *weak* coreset for the given set X

An Open Gap?

To utilize a weak coreset $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ of X, we need an approximation algorithm that solves the r-fuzzy K-means problem with respect to the set data set S and the restricted set of solutions $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ (see Observation 12.5). However, we do not know whether there is such an algorithm. In particular, given an approximation algorithm that solves the problem with respect to S and the complete set of solutions $(\mathbb{R}^D)^{\leq K}$, we cannot construct an approximative solution that is contained in $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ via Algorithm 3:

Observation 12.8 (we cannot use weak coresets)**.** *Assume, the tuple $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak coreset of X. Apply an $(1+\epsilon)$-approximation algorithm that solves the r-fuzzy K-means problem with respect to the set S and the complete set of solutions $(\mathbb{R}^D)^{\leq K}$. It returns some solution $C_{alg} \in (\mathbb{R}^D)^{\leq K} \supseteq \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ with $\phi_S^{(r)}(C_{alg}) \leq (1+\epsilon)\phi_{(S,K,r)}^{OPT}$.*

According to Lemma 6.4, we can apply Algorithm 3 to C_{alg} and X in order to compute a solution $\tilde{C}_{alg} \in \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ with $\phi_X^{(r)}(\tilde{C}_{alg}) \leq (1+\epsilon)\phi_X^{(r)}(C_{alg})$.

However, this does not *imply that $\phi_S^{(r)}(\tilde{C}_{alg}) \leq (1+\epsilon)\phi_{(S,K,r)}^{OPT}$. The reason for this is that $\tilde{C}_{alg} \subseteq C_{alg}$, and hence, $\phi_S^{(r)}(\tilde{C}_{alg}) \geq \phi_S^{(r)}(C_{alg})$ (see Lemma 5.11).*

Consequently, we cannot *conclude that $\phi_X^{(r)}(\tilde{C}_{alg}) \leq \mathrm{const} \cdot \phi_{(X,K,r)}^{OPT}$ as in Observation 12.5.*

To put it in a nutshell, we do not know whether one can utilize an arbitrary weak coreset $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ of X.

There is no Gap.

Fortunately, the set $S = ((s_m, v_m))_m$ constructed by Algorithm 12 takes a specific form. Its points s_m coincide with some points x_n from the given data set $X = ((x_n, w_n))_{n \in [N]}$. Moreover, the ϵ-approximate mean sets $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ have some useful properties.

To gain some intuition why we can exploit these properties, take note of the following: We consider solutions from $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$. This means that we consider solutions which induce non-negligible clusters with respect to X. The notion of a non-negligible cluster basically boils down to the fact that at least a single point x_n in X supports this cluster. The points s_n from S coincide with points x_n from X. To sum up, there is some connection between the notion of non-negligible clusters of S and non-negligible clusters of X. For more details, we refer to Section 12.5.7. There, we show that a weak coreset $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ of X where S has been constructed by Algorithm 12 is actually a *strong* coreset.

Weaker Coresets and a Discrete Search Space

While the notion of weak coresets is in principle useful, the next relaxed notion of a coreset is rather technical. To prove that Algorithm 12 computes a set S such that $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ is

a weak coreset for X, we follow the line of arguments from Chen (2009). In his proof, Chen implicitly makes use of another relaxed notion of a coreset.

Definition 12.9 (weaker ϵ-coresets)**.** *Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$, let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a fuzzifier function, $\epsilon \in [0,1]$, and $K \in \mathbb{N}$.*

Consider a set of solutions $\Theta \subseteq (\mathbb{R}^D)^{\leq K}$ and a data set $S \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)$. The tuple (S, Θ) is a weaker (than weak) ϵ-coreset of X for the r-fuzzy K-means problem if

$$\forall C \in \Theta : \ \phi_S^{(r)}(C) \in [1 \pm \epsilon] \cdot \phi_X^{(r)}(C) \ .$$

In this definition, Θ is not necessarily an ϵ-approximate mean set. Hence, we cannot use weaker coresets in the same way as in Observation 12.5.

In the following sections, we show that Algorithm 13 can be used to construct a set S such that $(S, \mathfrak{G}^K)$ is a weaker coreset where $\mathfrak{G}$ is a discrete search space (Definition 9.9). That is, unlike the discrete search space that we considered in Chapter 10, this set $\mathfrak{G}$ does *not* have the property that $\mathfrak{G}^K$ is an ϵ-approximate mean set of the given data set X. Nonetheless, a discrete search space has several useful properties, which we explained in Chapter 9. The coreset property helps us to exploit these properties of $\mathfrak{G}$.

12.5.2 Outline of the Analysis

In the first part of the analysis, we show that Algorithm 12 constructs a set S such that $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak coreset of the given data set X. The following three steps of this proof are similar to the proof from (Chen, 2009):

1. In Section 12.5.4, we show that, for an arbitrary but fixed unknown set of solutions Θ of some known size $|\Theta| \leq \gamma$, Algorithm 13 can be used to construct a data set S such that (S, Θ) is a weaker coreset of X, with constant probability.

2. In Section 12.5.5, we use this result to conclude that Algorithm 12 constructs a data set S such that $(S, \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak coreset of X. In particular, we construct a discrete search space $\mathfrak{G}$ with $|\mathfrak{G}|^K \leq \gamma$ (see Chapter 9) and exploit the fact that, due to our previous results, $(S, \mathfrak{G}^K)$ is a weaker coreset of X, with constant probability.

3. In Section 12.5.6, we analyse the size of the constructed data set S and the runtime of Algorithm 12.

In the second part of the analysis, we show that the data set S constructed by Algorithm 12 is a strong coreset, with constant probability.

4. In Section 12.5.7, we identify some useful properties of the construction and of the sets of solutions $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$. From these properties, we conclude that the data set S computed by Algorithm 12 actually is a strong coreset.

12.5.3 Preliminaries

In this section, we introduce some notation, take a close look at Algorithm 13, and note a basic observation that helps us to compare the cost of two different data sets.

Notation (K-Means)

We use the terms (strong) coreset, weak coreset and weaker (than weak) coreset also with respect to the K-means problem. In doing so, we refer to the respective definitions for the special case $r = \mathrm{id}$, where the r-fuzzy K-means problem corresponds to the K-means problem (Observation 5.23).

Properties of Algorithm 13

The following observations correspond to the notes that can be found in (Chen, 2009, pp. 927 (incl. footnote)).

Observation 12.10 (partition)**.** $\dot\cup_{l,j} X_{l,j} = \{x \mid (x,1) \in X\}$.

Proof. By construction and Definition 9.6, the sets $X_{l,j}$ are pairwise disjoint. It remains to show that $\{x \mid (x,1) \in X\} \subseteq \cup_{l,j} X_{l,j}$. Towards a contradiction, assume that there exists a point $(x,1) \in X$ with $x \notin \cup_{l,j} X_{l,j}$. This implies that

$$\mathtt{dist}(x,\mathfrak{M}) > 2^{\mathfrak{C}}\mathfrak{R} \geq \sqrt{\alpha |X|} \cdot \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{\alpha |X|}} = \mathrm{km}_X(\mathfrak{M}) \ ,$$

which contradicts the fact that $\mathrm{km}_X(\mathfrak{M}) \geq 1 \cdot \mathtt{dist}(y,\mathfrak{M})^2$ for all $(y,1) \in X$. □

Observation 12.11 (number of samples and their weights)**.** *For every set $S_{l,j}$ (with $X_{l,j} \neq 0$), observe the following:*

1. *In the (l,j)-th round the algorithm samples at most $|S_{l,j}| \leq 2 \cdot Q$ points.*
2. *The weight of all samples in $S^{\omega}_{l,j}$ sums up to $\sum_{(s,\omega_s) \in S^{\omega}_{l,j}} \omega_s = |X_{l,j}|$.*
3. *The weights of all data points in $S^{\omega}_{l,j} \in \mathrm{Dom}(\{x \mid (x,1) \in X\}, \mathbb{N})$ are natural numbers. In particular, if $\frac{|X_{l,j}|}{Q} \notin \mathbb{N}$, then each point in T^{ω}_{lj} is weighted by $\frac{|X_{l,j} \setminus T_{lj}|}{Q} = \left\lfloor \frac{|X_{l,j}|}{Q} \right\rfloor \in \mathbb{N}$.*

Proof. Assume $|X_{l,j}|/Q \in \mathbb{N}$. Then $\sum_{(s,\omega_s) \in S^{\omega}_{l,j}} \omega_s = q \cdot \frac{|X_{l,j}|}{Q} = |X_{l,j}|$ and $|S_{l,j}| = Q$.

Assume $|X_{l,j}|/Q \notin \mathbb{N}$. Then,

$$\sum_{(s,\omega_s) \in S^{\omega}_{l,j}} \omega_s = |R_{lj}| \frac{|X_{l,j} \setminus T_{lj}|}{Q} + |T_{lj}| \cdot 1 = Q \cdot \frac{|X_{l,j} \setminus T_{lj}|}{Q} + |T_{lj}| = |X_{l,j}| \ .$$

Moreover, $\hat{Q} = |X_{l,j}| - Q \cdot \left\lfloor \frac{|X_{l,j}|}{Q} \right\rfloor \leq Q$. Hence, $|S_{l,j}| = |T_{lj}| + |R_{lj}| = \hat{Q} + Q \leq 2 \cdot Q$. Furthermore, observe that $\hat{Q}$ is chosen such that

$$\frac{|X_{l,j} \setminus T_{lj}|}{Q} = \frac{|X_{l,j}| - \hat{Q}}{Q} = \frac{|X_{l,j}| - \left(|X_{l,j}| - Q \cdot \left\lfloor \frac{|X_{l,j}|}{Q} \right\rfloor\right)}{Q} = \left\lfloor \frac{|X_{l,j}|}{Q} \right\rfloor \in \mathbb{N} \ .$$

□

Observation 12.12 (size of S and runtime)**.** *The algorithm constructs a set S of the size*

$$|S| \in \mathcal{O}\left(\log(|X|)\log(\log(|X|)) \cdot \mathbf{c}_r(K)^{-2} \cdot K^2 \cdot \epsilon^{-2} \cdot \log(\gamma) \cdot \log\left(\delta^{-1}\right)\right) \ .$$

in time $\mathcal{O}(|X|DK + |S|)$.

Proof. From Algorithm 13 and Observation 12.11 (Item 1), we can conclude that S is a union of $L \cdot \mathfrak{C}$ sets of size $\leq 2 \cdot Q$ where $L \in \mathcal{O}(K)$, $\mathfrak{C} \in \mathcal{O}(\log(|X|)$, and

$$Q \in \mathcal{O}\left(\left(\frac{1}{\epsilon \mathbf{c}_r(K)}\right)^2 \log\left(\frac{K\mathfrak{C}\gamma^K}{\delta}\right)\right) \subseteq \mathcal{O}\left(\epsilon^{-2} \cdot \mathbf{c}_r(K)^{-2} \cdot \log(\log(|X|)) \cdot K \cdot \log(\gamma) \cdot \log\left(\delta^{-1}\right)\right) \ .$$

Combining these bounds yields the claim. □

Comparing the Cost of Two Different Data Sets

The following general observation will be helpful in Section 12.5.5.

Lemma 12.13 (different data sets)**.** *Let $C = (\mu_l)_{l\in[L]} \subseteq \mathbb{R}^D$. Let $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a [0,1]-reducing fuzzifier function. Consider two data sets $X = ((x_n,1))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$ and $Y = ((y_n,1))_{n\in[N]} \in \mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$ of the same size N. Then, for all $\hat{\epsilon} \in [0,1]$, it holds*

$$\left|\phi_X^{(r)}(C) - \phi_Y^{(r)}(C)\right| \leq \left(1+\frac{1}{\hat{\epsilon}}\right)\sum_{n=1}^{N}\|x_n - y_n\|_2^2 + \hat{\epsilon}\cdot\min\left\{\phi_Y^{(r)}(C),\ \phi_X(C)\right\}\ .$$

Proof. Let $P_X = (p_{nk})_{k\in[K]}$ and $P_Y = (\tilde{p}_{nk})_{k\in[K]}$ be the r-fuzzy clusterings induced by $(\mu_k)_{k\in[K]}$ with respect to X and Y, respectively. Let

$$\mathcal{E} \coloneqq \left|\phi_X^{(r)}(C) - \phi_Y^{(r)}(C)\right| = \left|\sum_{l=1}^{L}\sum_{n=1}^{N}\left(r(p_{nl})\left\|x_n-\mu_l\right\|_2^2 - r(\tilde{p}_{nl})\left\|y_n-\mu_l\right\|_2^2\right)\right|\ .$$

In the following, we distinguish two cases.

Case 1: If $\phi_X^{(r)}(C) \geq \phi_Y^{(r)}(C)$, then

$$\begin{aligned}
\mathcal{E} &= \phi_X^{(r)}(C) - \phi_Y^{(r)}(C)\\
&= \phi_X^{(r)}(C,(p_{nk})_{n,k}) - \phi_Y^{(r)}(C,(\tilde{p}_{nk})_{n,k})\\
&\leq \phi_X^{(r)}(C,(\tilde{p}_{nk})_{n,k}) - \phi_Y^{(r)}(C,(\tilde{p}_{nk})_{n,k})\\
&= \sum_{l=1}^{L}\sum_{n=1}^{N} r(\tilde{p}_{nl})\left(\left\|x_n-\mu_l\right\|_2^2 - \left\|y_n-\mu_l\right\|_2^2\right)\\
&\leq \sum_{l=1}^{L}\sum_{n=1}^{N} r(\tilde{p}_{nl})\left(\|x_n-y_n\|_2^2 + 2\|x_n-y_n\|_2\left\|y_n-\mu_l\right\|_2\right) && \text{(Lemma 9.4)}\\
&= \sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(\tilde{p}_{nl})\right)\|x_n-y_n\|_2^2 + 2\sum_{l=1}^{L}\sum_{n=1}^{N} r(\tilde{p}_{nl})\|x_n-y_n\|_2\left\|y_n-\mu_l\right\|_2\ .
\end{aligned}$$

Observe that, for all $a \in \mathbb{R}_+$ and $x,y \in \mathbb{R}$, we have $0 \leq (ax - \frac{1}{a}y)^2 = a^2x^2 - 2xy + \frac{1}{a^2}y^2$, which means that $2xy \leq a^2x^2 + \frac{1}{a^2}y^2$. Hence, for all $\hat{\epsilon} \in [0,1]$, we have

$$\begin{aligned}
2\sum_{n=1}^{N}\sum_{l=1}^{L} r(\tilde{p}_{nl})\left\|y_n-\mu_l\right\|_2\|x_n-y_n\|_2 &\leq \sum_{n=1}^{N}\sum_{l=1}^{L} r(\tilde{p}_{nl})\left(\hat{\epsilon}\left\|y_n-\mu_l\right\|_2^2 + \frac{1}{\hat{\epsilon}}\|x_n-y_n\|_2^2\right)\\
&= \hat{\epsilon}\cdot\phi_Y^{(r)}(C) + \frac{1}{\hat{\epsilon}}\sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(\tilde{p}_{nl})\right)\|x_n-y_n\|_2^2\ .
\end{aligned}$$

Combining these inequalities gives

$$\begin{aligned}
\mathcal{E} &\leq \left(\sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(\tilde{p}_{nl})\right)\|x_n-y_n\|_2^2\right) + \left(\hat{\epsilon}\cdot\phi_Y^{(r)}(C) + \frac{1}{\hat{\epsilon}}\sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(\tilde{p}_{nl})\right)\|x_n-y_n\|_2^2\right)\\
&= \left(1+\frac{1}{\hat{\epsilon}}\right)\sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(\tilde{p}_{nl})\right)\|x_n-y_n\|_2^2 + \hat{\epsilon}\cdot\phi_Y^{(r)}(C)\ .
\end{aligned}$$

Case 2: If $\phi_X^{(r)}(C) < \phi_Y^{(r)}(C)$, then we analogously obtain

$$\mathcal{E} \leq \left(1+\frac{1}{\hat{\epsilon}}\right)\sum_{n=1}^{N}\left(\sum_{l=1}^{L} r(p_{nl})\right)\|x_n-y_n\|_2^2 + \hat{\epsilon}\cdot\phi_X^{(r)}(C)\ .$$

Unfortunately, we do not know whether $\sum_{l=1}^{L} r(p_{nl}) \geq \sum_{l=1}^{L} r(\tilde{p}_{nl})$ for some $n \in [N]$, or not. However, given that r is [0,1]-reducing, we know that in both cases

$$\forall n \in [N]\colon \sum_{l=1}^{L} r(p_{nl}) \leq 1 \ \wedge\ \sum_{l=1}^{L} r(\tilde{p}_{nl}) \leq 1\ .$$

This yields the claim. □

12.5.4 Weaker Coreset for a Fixed Number of Arbitrary Solutions

Fix an arbitrary set of solutions $\Gamma \subset \mathbb{R}^D$. Assume that Γ is *unknown* to us, except for its size which, as we know, is upper bounded by some $\gamma \in \mathbb{N}$. With the help of Algorithm 15, we can construct a data set S such that $(S, \Gamma^{\leq K})$ is a weaker coreset of X, with constant probability.

Algorithm 15 Weaker Coreset

Require: $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\mathbf{c}_r(K) \in (0,1]$, $\epsilon \in (0,1]$, $\delta \in (0,1)$, $\gamma \in \mathbb{N}$.

1: Use the algorithm of Aggarwal et al. (2009) to compute an (α, β)-bicriteria approximation $\mathfrak{M} = (\mathfrak{m}_l)_{l \in [L]} \subseteq \mathbb{R}^D$ for the K-means problem with respect to X with $\alpha, \beta \geq 1$.
2: Apply Algorithm 13 to X, K, ϵ, δ, γ, α, β, and $\mathfrak{M}$ to compute a data set S.
3: **return** S

Theorem 12.14. *Let $\gamma \in \mathbb{N}$. Fix an arbitrary $\Gamma \subseteq \mathbb{R}^D$ with $|\Gamma| \leq \gamma$. Let $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$, $K \in \mathbb{N}$, a $\mathbf{c}_r$-contribution-bounded fuzzifier function $r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\epsilon \in (0,1]$, and $\delta \in (0,1)$.*

Then, given $X, K, r, \mathbf{c}_r(K), \epsilon, \delta$ and γ, Algorithm 15 computes a set $S \in \mathrm{Dom}(\{x \mid (x,1) \in X\}, \mathbb{N})$ such that, with a probability of $1-\delta$, the tuple $(S, \Gamma^{\leq K})$ is

1. *a weaker ϵ-coreset of X for the r-fuzzy K-means problem, and*
2. *a weaker $(\epsilon \cdot \mathbf{c}_r(K))$-coreset of X for the K-means problem.*

The set S has the size

$$|S| \in \mathcal{O}\left(\log(|X|)\log(\log(|X|)) \cdot \mathbf{c}_r(K)^{-2} \cdot K^2 \cdot \epsilon^{-2} \cdot \log(\gamma) \cdot \log\left(\delta^{-1}\right)\right) .$$

The algorithms' runtime is

$$\mathcal{O}\left(|X| D K \log(\delta^{-1}) + |S|\right) .$$

We point out that the form $\Theta = \Gamma^{\leq K}$ that the set of solutions takes here is only for the sake of simplicity (we will set $\Gamma = \mathfrak{G}$ for some discrete search space $\mathfrak{G}$ in the next chapter). Analogously, one can show that the result holds true for $\Theta \subseteq (\mathbb{R}^D)^{\leq K}$ with $|\Theta| \leq \gamma^K$. Moreover, one could combine Theorem 12.14 with our results from Section 10.3.

From Lemma 5.12 from (Chen, 2009, p. 938), we already know that Item 2 of Theorem 12.14 is satisfied, with constant probability. However, together with Lemma 6.1, this observation does not directly imply that Algorithm 15 computes a weaker coreset for the r-fuzzy K-means problem. Therefore, we have to refine the analysis from (Chen, 2009). To this end, we need the following concentration bound by Haussler (1992).

Lemma 12.15. *Let $\epsilon, \delta > 0$, $X \subseteq \mathbb{R}^D$ a finite set, $f : X \to \mathbb{R}$, and let $F \in \mathbb{R}$ be such that $\forall x \in X : 0 \leq f(x) \leq F$. Let $S \subset X$ be a uniform sample multi-set of size $|S| \geq \frac{1}{2}\epsilon^{-2}\ln(2/\delta)$. Then*

$$\Pr\left[\left|\frac{1}{|X|}\sum_{x \in X} f(x) - \frac{1}{|S|}\sum_{s \in S} f(s)\right| \leq \epsilon F\right] \geq 1 - \delta.$$

With this lemma at hand, we can now prove Theorem 12.14.

Proof of Theorem 12.14. This proof follows the lead of Chen (2009).

Fix an arbitrary $\Gamma \subset \mathbb{R}^D$ with $|\Gamma| \leq \gamma$. Consider a single run of the algorithm as described in the theorem. Consider an arbitrary but fixed $C \subseteq \Gamma$ with $|C| \leq K$.

Due to Observation 12.10 and due to the construction of S, we have $\dot{\cup}_{l,j} X_{l,j} = X$ and $\dot{\cup}_{l,j} S^{\omega}_{l,j} = S$. Hence, with the the triangle inequality, we can conclude

$$\left|\phi^{(r)}_X(C) - \phi^{(r)}_S(C)\right| \leq \sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}} \left|\phi^{(r)}_{X_{l,j}}(C) - \phi^{(r)}_{S^{\omega}_{l,j}}(C)\right| , \tag{12.6}$$

where we identify $X_{l,j} \subseteq \mathbb{R}^D$ with an unweighted data set, as explained in Definition 2.3, and let $S_{l,j} := T_{lj} := R_{lj} := \emptyset$ if $X_{l,j} = \emptyset$.

Consider an arbitrary but fixed summand $\left|\phi^{(r)}_{X_{l,j}}(C) - \phi^{(r)}_{S^{\omega}_{l,j}}(C)\right|$ with $l \in [L]$ and $j \in \{0,1,\dots,\mathfrak{E}\}$. Our goal is to derive a probabilistic upper bound on this summand with the help of Lemma 12.15. To this end, we distinguish two cases:

Case 1: First, consider the case that $X_{l,j} \neq \emptyset$ with $|X_{l,j}|/q \in \mathbb{N}$.

Then, $\omega_s = \frac{|X_{l,j}|}{q} \in \mathbb{N}$ for each $(s,\omega_s) \in S^{\omega}_{l,j}$ and $|S_{l,j}| = q$. Hence, $\frac{1}{|X_{l,j}|} \cdot \omega_s = \frac{1}{|S_{l,j}|}$. Therefore, we have

$$\begin{aligned}\frac{1}{|X_{l,j}|}\phi^{(r)}_{S^{\omega}_{l,j}}(C) &= \frac{1}{|X_{l,j}|}\sum_{(s,\omega_s)\in S^{\omega}_{l,j}} \phi^{(r)}_{(s,\omega_s)}(C) \\ &= \sum_{(s,\omega_s)\in S^{\omega}_{l,j}} \left(\frac{1}{|X_{l,j}|}\omega_s\right)\phi^{(r)}_{(s,1)}(C) \\ &= \frac{1}{|S_{l,j}|}\sum_{s\in S_{l,j}} \phi^{(r)}_{(s,1)}(C)\,.\end{aligned}$$

Consequently, we can write

$$\begin{aligned}\left|\phi^{(r)}_{X_{l,j}}(C) - \phi^{(r)}_{S^{\omega}_{l,j}}(C)\right| &= |X_{l,j}|\left|\frac{1}{|X_{l,j}|}\sum_{x\in X_{l,j}} \phi^{(r)}_{(x,1)}(C) - \frac{1}{|S_{l,j}|}\sum_{s\in S_{l,j}} \phi^{(r)}_{(s,1)}(C)\right| \\ &= |X_{l,j}|\left|\frac{1}{|X_{l,j}|}\phi^{(r)}_{X_{l,j}}(C) - \frac{1}{|S_{l,j}|}\phi^{(r)}_{S_{l,j}}(C)\right|\,, \end{aligned} \tag{12.7}$$

where we identify $X_{l,j}$ and $S_{l,j}$ with unweighted data sets, as explained in Definition 2.3.

We want to apply Lemma 12.15 with $f(\cdot) = \phi^{(r)}_{(\cdot)}(C)$: Obviously, $\phi^{(r)}_{(x,1)}(C) \geq 0$ for all $x \in \mathbb{R}^D$. Next, we need to determine an upper bound $F(l,j) \in \mathbb{R}$ with $\phi^{(r)}_{(x,1)}(C) \leq F(l,j)$ for all $x \in X_{l,j}$. Due to Lemma 6.1, we have $\phi^{(r)}_{(x,1)}(C) \leq \mathtt{dist}(x,C)^2$ for all $x \in \mathbb{R}^D$. Fix some

$$\mathfrak{c}_{l,j} \in \arg\min\{\mathtt{dist}(x,C) \,|\, x \in X_{l,j}\}\,. \tag{12.8}$$

Then, for each $x \in X_{l,j}$, we can bound

$$\begin{aligned}\mathtt{dist}(x,C)^2 &\leq 2\left(\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2 + \big\|\mathfrak{c}_{l,j} - x\big\|_2^2\right) && \text{(Lemma A.3 and Lemma A.2)} \\ &\leq 4\left(\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2 + \big\|\mathfrak{c}_{l,j} - \mathfrak{m}_k\big\|_2^2 + \|\mathfrak{m}_k - x\|_2^2\right) && \text{(Lemma A.2)} \\ &\leq 4\left(\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2 + 2^{2j+1}\mathfrak{R}^2\right)\,,\end{aligned}$$

where the last inequality is due to the fact that $X_{l,j} \subseteq \mathfrak{U}_{l,j} \subseteq \mathtt{B}\big(\mathfrak{m}_l, 2^j\mathfrak{R}\big)$ (cf. Definition 9.6). Hence, we can set

$$F(l,j) := 4\left(\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2 + 2^{2j+1}\mathfrak{R}^2\right)\,.$$

Let $\epsilon' := \epsilon \cdot \mathfrak{c}_r(K)/(44\alpha)$ and $\delta' := \delta/\big(2L\mathfrak{E}(\gamma+1)^K\big)$. Note that $S_{l,j}$ is a uniform sample multi-set of size

$$Q \geq \frac{1}{2}(\epsilon')^{-2}\ln(2/\delta')$$

from $X_{l,j}$ (assuming Q_{const} is sufficiently large). So we can apply Lemma 12.15 with ϵ' instead of ϵ and δ' instead of δ. Recall (12.7). We obtain that, with a probability of at least

$1-\delta/\big(2L\mathfrak{E}(\gamma+1)^K\big)$,

$$\begin{aligned}\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{S^{\omega}_{l,j}}(C)\right| &\leq \left|X_{l,j}\right|\cdot\epsilon' F(l,j)\\ &= 4\epsilon'\left(\left|X_{l,j}\right|\,\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2+\left|X_{l,j}\right|2^{2j+1}\mathfrak{R}^2\right)\,. \end{aligned}\tag{12.9}$$

Consider the first term in (12.9). Due to (12.8),

$$\left|X_{l,j}\right|\,\mathtt{dist}\big(\mathfrak{c}_{l,j},C\big)^2\leq\sum_{x\in X_{l,j}}\mathtt{dist}(x,C)^2=\mathrm{km}_{X_{l,j}}(C)\,.$$

Consider the second term in (12.9). For $j=0$, we know

$$\left|X_{l,j}\right|2^{2j+1}\mathfrak{R}^2=\left|X_{l,j}\right|2\mathfrak{R}^2=\frac{2}{\alpha}\frac{\left|X_{l,j}\right|}{|X|}\mathrm{km}_X(\mathfrak{M})\leq 2\frac{\left|X_{l,j}\right|}{|X|}\mathrm{km}_X(\mathfrak{M})\,.$$

For $j\geq 1$, observe that $X_{l,j}\subseteq\mathfrak{U}_{l,j}\cap A_l$ where $\mathfrak{U}_{l,j}\cap\mathtt{B}\big(\mathfrak{m}_l,2^{j-1}\mathfrak{R}\big)=\emptyset$ (cf. Definition 9.6) and where A_l is the l-th K-means hard cluster induced by $\mathfrak{M}$. Hence, for all $x\in X_{l,j}$, we have $(2^{j-1}\mathfrak{R})^2=2^{2j-2}\mathfrak{R}^2\leq\|x-\mathfrak{m}_l\|_2^2=\mathtt{dist}(x,\mathfrak{M})^2$. Hence, for $j\geq 1$,

$$\left|X_{l,j}\right|2^{2j+1}\mathfrak{R}^2\leq 2^3\sum_{x\in X_{l,j}}\|x-\mathfrak{m}_k\|_2^2=8\,\mathrm{km}_{X_{l,j}}(\mathfrak{M})\,.$$

By combining the upper bounds (for the case $j=0$ and $j\geq 1$) with (12.9), we obtain that

$$\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{S^{\omega}_{l,j}}(C)\right|\leq 4\epsilon'\left(\mathrm{km}_{X_{l,j}}(C)+8\,\mathrm{km}_{X_{l,j}}(\mathfrak{M})+2\frac{\left|X_{l,j}\right|}{|X|}\mathrm{km}_X(\mathfrak{M})\right)\tag{12.10}$$

with a probability of at least $1-\delta/(2L\mathfrak{E}(\gamma+1)^K)^{-1}$.

Case 2: Second, consider the case that $X_{l,j}\neq\emptyset$ and $\left|X_{l,j}\right|/Q\notin\mathbb{N}$.

Then, $S_{l,j}=T_{lj}\cup R_{lj}$. Observe that, by construction, $T_{lj}\cup R_{lj}\subseteq X_{l,j}$. Moreover, by construction, each data point $(t,\omega_t)\in T^{\omega}_{lj}$ is weighted by $\omega_t=1$. Recall that the points in $X\in\mathrm{Dom}\big(\mathbb{R}^D,\{1\}\big)$ are also weighted by 1. Therefore, we can write

$$\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{S^{\omega}_{l,j}}(C)\right|=\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{T^{\omega}_{lj}}(C)-\phi^{(r)}_{R^{\omega}_{lj}}(C)\right|=\left|\phi^{(r)}_{X_{l,j}\setminus T_{lj}}(C)-\phi^{(r)}_{R^{\omega}_{lj}}(C)\right|\,.\tag{12.11}$$

Observe that R_{lj} contains Q uniform samples from $X_{l,j}\setminus T_{lj}$. In R^{ω}_{lj}, each of these samples from R_{lj} is weighted by $\omega=\left|X_{l,j}\setminus T_{lj}\right|/Q$. From Observation 12.11 (Item 3), we know that $\omega\in\mathbb{N}$. Moreover, we can apply the same line of argument as in the first case (with $X_{l,j}$ replaced by $X_{l,j}\setminus T_{lj}$ and $S^{\omega}_{l,j}$ replaced by R^{ω}_{lj}). Combining this result with (12.11) yields the correctness of a result that is analog to (12.10) (with the same probability).

To sum up, in either case, we know that (12.10) holds true with a probability of $1-\delta/(2L\mathfrak{E}(\gamma+1)^K)^{-1}$.

Due to Boole's inequality, we know that (12.10) holds simultaneously for every $l\in[L]$ and $j=\{0,1,\ldots,\mathfrak{E}\}$ with a probability of at least $1-\delta/(2(\gamma+1)^K)$. Due to Observation 12.10, we have $\sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}}\left|X_{l,j}\right|=|X|$. Hence, by taking the sum on both sides of (12.10), we obtain

$$\sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}}\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{S_{l,j}}(C)\right|\leq 4\epsilon'\left(\mathrm{km}_X(C)+10\,\mathrm{km}_X(\mathfrak{M})\right)\,.\tag{12.12}$$

Due to Lemma 6.1 and $|C|\leq K$, it holds that $\mathrm{km}_X(C)\leq\frac{1}{\mathfrak{c}_r(K)}\phi^{(r)}_X(C)$ and $\mathrm{km}_X(\mathfrak{M})\leq\alpha\,\mathrm{km}^{OPT}_{(X,K)}\leq\alpha\frac{1}{\mathfrak{c}_r(K)}\phi^{(r)}_X(C)$.

We can conclude

$$\sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}}\left|\phi^{(r)}_{X_{l,j}}(C)-\phi^{(r)}_{S_{l,j}}(C)\right| \leq 4\epsilon'\cdot\frac{(1+10\alpha)}{\mathbf{c}_r(K)}\phi^{(r)}_X(C) \leq \epsilon\cdot\phi^{(r)}_X(C)\ ,$$

where the last inequality is due to $\epsilon' = \epsilon\cdot\mathbf{c}_r(K)/(44\alpha)$ and $\alpha \geq 1$.

Recall (12.6). We just showed that, for a fixed $C\subseteq\Gamma$, $|C|\leq K$, with a probability of at least $1-\delta/(2(\gamma+1)^K)$, it holds

$$\left|\phi^{(r)}_X(C)-\phi^{(r)}_S(C)\right| \leq \epsilon\cdot\phi^{(r)}_X(C)\ . \tag{12.13}$$

There are at most $(|\Gamma|+1)^K \leq (\gamma+1)^K$ different vectors $C\subseteq\Gamma$ with $|C|\leq K$. Thus, by Boole's inequality, with a probability of at least $1-\delta/2$, (12.13) holds simultaneously for all $C\subseteq\Gamma$ with $|C|\leq K$.

Now consider the case that $r=\mathrm{id}$. Recall that $\phi^{(\mathrm{id})}_X(C) = \mathrm{km}_X(C)$ for all $X\in\mathrm{Dom}\left(\mathbb{R}^D,\mathbb{R}_+\right)$ and $C\subseteq\mathbb{R}^D$. Follow the same line of arguments as before, up to (12.12). With $r=\mathrm{id}$, this inequality reads

$$\sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}}\left|\mathrm{km}_{X_{l,j}}(C)-\mathrm{km}_{S_{l,j}}(C)\right| \leq 4\epsilon'\left(\mathrm{km}_X(C)+10\,\mathrm{km}_X(\mathfrak{M})\right)\ .$$

Recall that $\mathrm{km}_X(\mathfrak{M}) \leq \alpha\,\mathrm{km}^{OPT}_{(X,K)} \leq \alpha\,\mathrm{km}_X(C)$ since $|C|\leq K$. Hence,

$$\begin{aligned}
|\mathrm{km}_X(C)-\mathrm{km}_S(C)|\sum_{l=1}^{L}\sum_{j=0}^{\mathfrak{E}}\left|\mathrm{km}_{X_{l,j}}(C)-\mathrm{km}_{S_{l,j}}(C)\right| & \\
\leq 4\epsilon'(1+10\alpha)\mathrm{km}_X(C) & \\
\leq \epsilon\cdot\mathbf{c}_r(K)(1/(11\cdot\alpha)+10/11)\mathrm{km}_X(C) & \qquad (\epsilon'=\epsilon\cdot\mathbf{c}_r(K)/(44\alpha)) \\
\leq \epsilon\cdot\mathbf{c}_r(K)\mathrm{km}_X(C)\ . & \qquad (\alpha\geq 1)
\end{aligned}$$

By using Boole's inequality to combine our result (for the given r) with this special case ($r=\mathrm{id}$), we obtain that both results hold simultaneously with a probability of at least $1-\delta$. This yields the claim. □

12.5.5 Weak Coreset

In this section, we show that Algorithm 12 computes a set S such that $(S,\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak coreset for the given data set X.

Theorem 12.16. *Given an unweighted data set $X\in\mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$, $K\in\mathbb{N}$, a [0,1]-reducing fuzzifier function $r:\mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$, $\mathbf{i}_r$-increase-bounded and $\mathbf{c}_r$-contribution-bounded, the values $\mathbf{c}_r(K)\in(0,1]$ and $\mathbf{i}_r\in[1,\infty)$, $\epsilon\in(0,1)$ and $\delta\in(0,1)$, Algorithm 12 computes a data set $S\in\mathrm{Dom}\left(\{x\mid(x,w)\in X\},\mathbb{N}\right)$ such that, with a probability of at least $1-\delta$, the tuple $(S,\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak ϵ-coreset of X for the r-fuzzy K-means problem.*

The proof of this theorem is basically an analogon to the proof from (Chen, 2009, pp.935) for (strong) coresets for the K-means problem.

Preliminaries

Consider arbitrary but fixed $X=(x_n)_{n\in[N]}\in\mathrm{Dom}\left(\mathbb{R}^D,\{1\}\right)$, $K\in\mathbb{N}$, fuzzifier function r that is $\mathbf{i}_r$-increase-bounded ($\mathbf{i}_r\in[1,\infty)$), $\mathbf{c}_r$-contribution-bounded ($\mathbf{c}_r(K)\in(0,1]$), and [0,1]-reducing, $\epsilon\in(0,1)$, and $\delta\in(0,1)$. We analyse a single run of Algorithm 12 given X, K, r, $\mathbf{c}_r(K)$, $\mathbf{i}_r$, ϵ,

and δ. We denote the (α,β)-approximation computed in the first step of the algorithm by $\mathfrak{M} = (\mathfrak{m}_l)_{l\in[L]}$. That is, $\alpha,\beta \in \mathcal{O}(1)$ and

$$L = |\mathfrak{M}| \leq \beta \cdot K \quad \text{and} \quad \mathrm{km}_X(\mathfrak{M}) \leq \alpha\, \mathrm{km}_{(X,K)}^{OPT} \,. \tag{12.14}$$

Fix an arbitrary solution $C \in \Theta_{(r,\mathbf{i}_r,K,\epsilon)}$. That is, $C = (\mu_t)_{t\in[T]} \subset \mathbb{R}^D$ contains

$$T = |C| \leq K \tag{12.15}$$

mean vectors and induces some r-fuzzy clustering $(p_{nt})_{n\in[N],t\in[T]}$ of X that has no $(\mathbf{i}_r,K,\epsilon)$-negligible clusters. Hence,

$$\forall t \in [T]\ \exists n \in [N]:\ p_{nt} \geq \frac{\epsilon}{2\mathbf{i}_r K^2} \,. \tag{12.16}$$

Choose a Large (Discrete) Search Space

Additionally, we define the search space

$$\mathfrak{U} := \mathfrak{U}(\mathfrak{E}', \mathfrak{R}, \mathfrak{M}) \subset \mathbb{R}^D$$

where

$$\mathfrak{E}' := \left\lfloor \frac{1}{2} \log\left(9 \cdot \alpha \cdot \frac{20}{\tilde{\epsilon}^2 \cdot r\left(\frac{\epsilon}{2\mathbf{i}_r K^2}\right)} \cdot |X| \right) \right\rfloor \,, \quad \mathfrak{R} = \sqrt{\frac{\mathrm{km}_X(\mathfrak{M})}{\alpha\,|X|}} \quad \text{and} \quad \tilde{\epsilon} = \frac{\epsilon \cdot \mathbf{c}_r(K)}{504\alpha} \,. \tag{12.17}$$

Note that $\mathfrak{R}$ and $\tilde{\epsilon}$ take the values described in Algorithm 12.

Observation 12.17 (large search space)**.** *From Lemma 9.10, we know that the search space $\mathfrak{U}$ covers the data set X well:*

$$\bigcup_{x\in X} \mathsf{B}(x,\mathfrak{r}) \subseteq \mathfrak{U} \quad \textit{where} \quad \mathfrak{r} = 2\sqrt{\frac{20}{\tilde{\epsilon}^2 \cdot r\left(\frac{\epsilon}{2\mathbf{i}_r K^2}\right)} \cdot \mathrm{km}_X(\mathfrak{M})} \,.$$

For those points inside the search space $\mathfrak{U}$, function $\mathfrak{g}$ defines a representative in $\mathfrak{G}$. Let $\mathfrak{g}$ be a representative function described by $(\mathfrak{E}', \mathfrak{R}, \mathfrak{M})$ and $(\tilde{\epsilon}/36)$ and let $\mathfrak{G}$ be the corresponding discrete search space

$$\mathfrak{G} := \{\mathfrak{g}(x)\ |x \in \mathfrak{U}\} \,.$$

Weaker Coreset for r-Fuzzy K-Means and Strong Coreset for K-Means

With our results from Section 12.5.4, we can conclude that $(S, \mathfrak{G}^{\leq K})$ is a weaker coreset, with high probability:

Claim 12.18 (weaker coresets)**.** *With a probability of at least $1-\delta$, $(S,\mathfrak{G}^{\leq K})$ is a weaker $(\mathbf{c}_r(K)\cdot\tilde{\epsilon})$-coreset of X for the K-means problem and a weaker $\tilde{\epsilon}$-coreset of X for the r-fuzzy K-means problem.*

Proof. Recall that the discrete search space $\mathfrak{G}$ is described by $(\mathfrak{E}', \mathfrak{R}, \mathfrak{M})$ and the precision $\tilde{\epsilon}/36$, where $|\mathfrak{M}| \leq \beta K$. From Lemma 9.17, the chosen parameters (12.17), and the definition of γ in Algorithm 12, we know that

$$|\mathfrak{G}| \leq |\mathfrak{M}| \cdot (\mathfrak{E}'+1) \cdot \left(\frac{16\sqrt{\pi \cdot e}}{\tilde{\epsilon}/36}\right)^D \leq \beta K \cdot (\mathfrak{E}'+1) \cdot \left(\frac{47\cdot 36}{\tilde{\epsilon}}\right)^D = \gamma \,.$$

Applying Theorem 12.14 yields the claim. □

So, from now on, let us assume that this is the case: $(S, \mathfrak{G}^{\leq K})$ is a weaker $(\mathbf{c}_r(K)\cdot\tilde{\epsilon})$-coreset of X for the K-means problem and a weaker $\tilde{\epsilon}$-coreset of X for the r-fuzzy K-means problem. In particular, this means that

$$\forall \tilde{C} \in \mathfrak{G}^{\leq K} : \left|\phi_X^{(r)}(\tilde{C}) - \phi_S^{(r)}(\tilde{C})\right| \leq \tilde{\epsilon}\cdot\phi_X^{(r)}(\tilde{C}) \,. \tag{12.18}$$

Under this assumption, then we know that S already does part of the trick:

Claim 12.19. *S is a $(\mathbf{c}_r(K)\cdot\tilde{\epsilon})$-coreset of X for the K-means problem. That is,*

$$\forall \tilde{C} \in (\mathbb{R}^D)^{\leq K} : \left|\mathrm{km}_X\left(\tilde{C}\right) - \mathrm{km}_S\left(\tilde{C}\right)\right| \leq \tilde{\epsilon}\cdot\mathbf{c}_r(K)\cdot\mathrm{km}_X\left(\tilde{C}\right) \,. \tag{12.19}$$

Proof. Consider the work of (Chen, 2009, pp. 935) under the assumption that $(S, \mathfrak{G}^{\leq K})$ is a weaker $(\mathbf{c}_r(K)\cdot\tilde{\epsilon})$-coreset of X for the K-means problem. □

Case 1: Inside the Search Space

Following the same line of arguments as in (Chen, 2009, p. 937), we first consider the case that all means from C lie inside the search space $\mathfrak{U}$. In this case, we can make use of the fact that for all means from $C \subseteq \mathfrak{U}$ there exist representatives in $\mathfrak{g}(C) \subseteq \mathfrak{G}$ and that, as we showed in Section 9.4, these representatives are similar to the original means C. This is useful simply because we know that S exhibits the coreset property with respect to means from $\mathfrak{G}$.

Claim 12.20. *If $C \subseteq \mathfrak{U}$, then*

$$\phi_S^{(r)}(C) \leq \epsilon\cdot\phi_X^{(r)}C \,.$$

Proof. Start by observing that

$$\left|\phi_S^{(r)}(C) - \phi_X^{(r)}(C)\right| \tag{12.20}$$

$$\leq \left|\phi_S^{(r)}(C) - \phi_S^{(r)}(\mathfrak{g}(C))\right| + \left|\phi_S^{(r)}(\mathfrak{g}(C)) - \phi_X^{(r)}(\mathfrak{g}(C))\right| + \left|\phi_X^{(r)}(\mathfrak{g}(C)) - \phi_X^{(r)}(C)\right| \,. \tag{12.21}$$

Recall that $\mathfrak{g}$ is a representative function that defines a representative for each point in $\mathfrak{U} = \mathfrak{U}(\mathfrak{E}', \mathfrak{R}, \mathfrak{M})$ with precision $\tilde{\epsilon}/36$. Since $C \subseteq \mathfrak{U}$, we can apply Corollary 9.16 with respect to C and $\tilde{\epsilon}/36$.

Moreover, recall that $|C| = T \leq K$ and $|\mathfrak{M}| = L \leq K$. Hence, $\mathrm{km}_X(\mathfrak{M}) \leq \alpha\,\mathrm{km}_{(X,K)}^{OPT}$ and

$$\mathrm{km}_X(\mathfrak{M}) \leq \alpha\,\mathrm{km}_{(X,K)}^{OPT} \leq \alpha\,\mathrm{km}_X(C) \,. \tag{12.22}$$

Besides that, since S is a strong $(\tilde{\epsilon}\cdot\mathbf{c}_r)$-coreset for the K-means problem (12.19), we can conclude that

$$\mathrm{km}_S(\mathfrak{M}) \leq (1+\tilde{\epsilon}\cdot\mathbf{c}_r)\mathrm{km}_X(\mathfrak{M}) \quad \text{and} \quad \mathrm{km}_S(C) \leq (1+\tilde{\epsilon}\cdot\mathbf{c}_r)\mathrm{km}_X(C) \,. \tag{12.23}$$

First, we consider the last summand of the upper bound from (12.21). We have

$$\begin{aligned}
\left|\phi_X^{(r)}(\mathfrak{g}(C)) - \phi_X^{(r)}(C)\right| &\leq 36\tilde{\epsilon}\left(\mathrm{km}_X(C) + \mathrm{km}_X(\mathfrak{M}) + |X|\mathfrak{R}^2\right) \\
&\leq 36\tilde{\epsilon}(\mathrm{km}_X(C) + (1+1/\alpha)\mathrm{km}_X(\mathfrak{M})) && \text{(Equation (12.17))} \\
&\leq 36\tilde{\epsilon}(\alpha+2)\mathrm{km}_X(C) && \text{(Equation (12.22))} \\
&\leq \tilde{\epsilon}\cdot\frac{36(\alpha+2)}{\mathbf{c}_r(K)}\phi_X^{(r)}(C) && \text{(Lemma 6.1)} \\
&\leq \tilde{\epsilon}\cdot\frac{108\alpha}{\mathbf{c}_r(K)}\phi_X^{(r)}(C) \,. && (\alpha \geq 1)
\end{aligned}$$

Second, consider the first summand of the upper bound from (12.21). Similarly, we have

$$\begin{aligned}
&\left|\phi_S^{(r)}(C) - \phi_S^{(r)}(\mathfrak{g}(C))\right| \\
&\leq 36\tilde{\epsilon}(\mathrm{km}_S(C) + \mathrm{km}_S(\mathfrak{M}) + |S|\mathfrak{R}) \\
&\leq 36\tilde{\epsilon}(\mathrm{km}_S(C) + \mathrm{km}_S(\mathfrak{M}) + \mathrm{km}_X(\mathfrak{M})/\alpha) && (|S| \leq |X| \text{ and Equation (12.17)}) \\
&\leq 36\tilde{\epsilon}((1+\tilde{\epsilon}\mathbf{c}_r(K))\mathrm{km}_X(C) + (1+\tilde{\epsilon}\mathbf{c}_r(K) + 1/\alpha)\mathrm{km}_X(\mathfrak{M})) && (\text{Equation (12.19)}) \\
&\leq 36\tilde{\epsilon}(1+\tilde{\epsilon}\mathbf{c}_r(K) + \alpha + \alpha\tilde{\epsilon}\mathbf{c}_r(K) + 1)\mathrm{km}_X(C) && (\text{Equation (12.22)}) \\
&\leq 180\alpha \cdot \tilde{\epsilon} \cdot \mathrm{km}_X(C) && (\tilde{\epsilon}, \mathbf{c}_r(K) \leq 1, \alpha \geq 1) \\
&\leq \tilde{\epsilon}\frac{180\alpha}{\mathbf{c}_r(K)} \cdot \phi_X^{(r)}(C)\,. && (\text{Lemma 6.1})
\end{aligned}$$

To bound the second summand, we use our assumption (12.18) that $(S, \mathfrak{G}^{\leq K})$ is an $\tilde{\epsilon}$-weaker coreset for the r-fuzzy K-means problem with respect to X. Observe that $\mathfrak{g}(C) \subseteq \mathfrak{G}$. Hence, with (12.18), we can directly bound

$$\left|\phi_S^{(r)}(\mathfrak{g}(C)) - \phi_X^{(r)}(\mathfrak{g}(C))\right| \leq \tilde{\epsilon} \cdot \phi_X^{(r)}(\mathfrak{g}(C))\,.$$

With our bound on the last summand of the upper bound from (12.21), we can conclude that

$$\begin{aligned}
\left|\phi_S^{(r)}(\mathfrak{g}(C)) - \phi_X^{(r)}(\mathfrak{g}(C))\right| &\leq \tilde{\epsilon} \cdot \left(1 + \tilde{\epsilon} \cdot \frac{108\alpha}{\mathbf{c}_r(K)}\right)\phi_X^{(r)}(C) \\
&\leq \tilde{\epsilon} \cdot \frac{216\alpha}{\mathbf{c}_r(K)} \cdot \phi_X^{(r)}(C)\,. && (\tilde{\epsilon}, \mathbf{c}_r(K) \leq 1,\ \alpha \geq 1)
\end{aligned}$$

By combining our bounds on the single summands of the upper bound from (12.21) and using the fact that $\tilde{\epsilon} = \epsilon \cdot \mathbf{c}_r(K)/(504\alpha)$, we obtain

$$\left|\phi_S^{(r)}(C) - \phi_X^{(r)}(C)\right| = \tilde{\epsilon} \cdot \frac{504\alpha}{\mathbf{c}_r(K)}\phi_X^{(r)}(C) \leq \epsilon \cdot \phi_X^{(r)}(C)\,.$$

□

Case 2: Outside the Search Space

Next, we consider the case that one of the mean vectors from C is not contained in the large search space $\mathfrak{U}$. In this case, we can exploit the fact that the resulting r-fuzzy K-means cost of C is rather large. Moreover, we can make use of the specific form of the coreset construction. Take note of the following observation:

Observation 12.21 (exploit the natural weights)**.** *From Observation 12.11 (Item 2 and Item 3) we can conclude that there exists a function* $\mathfrak{s} : \{x \mid (x,1) \in X\} \mapsto \{x \mid (x,1) \in X\}$ *that satisfies*

$$\forall l \in [L]\ \forall j \in \{0, 1, \ldots, \mathfrak{E}\} :\ \mathfrak{s}(X_{l,j}) = S_{l,j} \subseteq X_{l,j}$$

and

$$\forall (s, \omega_s) \in S :\ \left|\mathfrak{s}(s)^{-1}\right| = \omega_s\,.$$

We stress the fact that this result exploits the specific form of the constructed sets: Algorithm 13 computes data sets $S_{l,j}^{\omega}$ whose data points are weighted by *natural numbers* and that the sum of these weights is equal to the number of points in $X_{l,j} = \mathfrak{U}_{l,j} \cap A_l$.

In the following, let us fix a function $\mathfrak{s}$ as described in Observation 12.21. With this function, we can now bound the cost between the given data set X and the presumed coreset S as follows.

Claim 12.22 (upper bound)**.** *For all* $\hat{\epsilon} \in [0,1]$*, we have*

$$\left|\phi_X^{(r)}(C) - \phi_S^{(r)}(C)\right| \leq \left(1 + \frac{1}{\hat{\epsilon}}\right) \sum_{n=1}^{N} \|x_n - \mathfrak{s}(x_n)\|_2^2 + \hat{\epsilon} \cdot \min\left\{\phi_S^{(r)}(C), \phi_X^{(r)}(C)\right\} .$$

Proof. Let $\tilde{S}$ be the unweighted data set that contains ω_s copies of $(s,1)$ for each of the $|S|$ data points $(s,\omega_s) \in S$. Recall from Corollary 8.7 that $\phi_S^{(r)}(C) = \phi_{\tilde{S}}^{(r)}(C)$ for all $C \subseteq \mathbb{R}^D$.

Observe that $\left|\tilde{S}\right| = \sum_{(s,\omega_s)\in S} \omega_s = |X|$ due to Observation 12.10 and Observation 12.11 (Item 2). Moreover, recall that r is $[0,1]$-reducing. Hence, we can apply Lemma 12.13 to X and $\tilde{S}$. This yields the claim. □

Consider the first summand of this upper bound. Analogously to (Chen, 2009, p. 934), we can upper bound this sum in terms of $\mathrm{km}_X(\mathfrak{M})$.

Claim 12.23.

$$\sum_{n=1}^{N} \|x_n - \mathfrak{s}(x_n)\|_2^2 \leq 20 \cdot \mathrm{km}_X(\mathfrak{M})$$

Proof. Consider an arbitrary $x_n \in X$. By construction, there exists a set $X_{l,j}$ such that $x_n, \mathfrak{s}(x_n) \in X_{l,j}$. Recall that $X_{l,j} = A_l \cap \mathfrak{U}_{l,j}$. With Lemma 9.11, we can conclude that

$$\|x_n - \mathfrak{s}(x_n)\|_2 \leq \mathtt{diam}(\mathfrak{U}_{l,j}) \leq 2\max\left\{2(\|x_n - \mathfrak{m}_l\|_2, \mathfrak{R}\right\} = 2\max\{2\,\mathtt{dist}(x_n, M), \mathfrak{R}\} .$$

Hence,

$$\|x_n - \mathfrak{s}(x_n)\|_2^2 \leq 4\max\left\{4\,\mathtt{dist}(x_n, M)^2, \mathfrak{R}^2\right\} .$$

By summing over all $x_n \in X$, we obtain

$$\begin{aligned}
\sum_{n=1}^{N} \|x_n - \mathfrak{s}(x_n)\|_2^2 &\leq 4 \sum_{n=1}^{N} \max\left\{4\,\mathtt{dist}(x_n, M)^2, \mathfrak{R}^2\right\} \\
&\leq 4 \sum_{n=1}^{N} 4\,\mathtt{dist}(x_n, M)^2 + \mathfrak{R}^2 \\
&= 4\left(4\,\mathrm{km}_X(\mathfrak{M}) + N \cdot \mathfrak{R}^2\right) \\
&= 4(4 + 1/\alpha)\,\mathrm{km}_X(\mathfrak{M}) && \text{(Equation (12.2))} \\
&\leq 20\,\mathrm{km}_X(\mathfrak{M}) . && (\alpha \geq 1)
\end{aligned}$$

□

This upper bound seems to be very large. However, since one mean lies outside the (particularly large) search space $\mathfrak{U}$ and since this cluster has non-negligible support (by assumption (12.16), there is no negligible cluster), the r-fuzzy K-means cost of C must be large. Note that this is the only argument in the whole proof that requires the notion of non-negligible clusters.

Claim 12.24. *If* $C \not\subseteq \mathfrak{U}$*, then*

$$\mathrm{km}_X(\mathfrak{M}) \leq \frac{\tilde{\epsilon}^2}{20} \phi_X^{(r)}(C) .$$

Proof. Let

$$b := \frac{20}{\tilde{\epsilon}^2 \cdot r\left(\frac{\epsilon}{2\mathfrak{i}_r K^2}\right)} . \tag{12.24}$$

Fix an arbitrary $\mu_t \in C \setminus \mathfrak{U}$. From Observation 12.17, we know that

$$\mathtt{dist}\left(\mu_t, \mathfrak{M}\right)^2 > 4b \cdot \mathrm{km}_X(\mathfrak{M}) . \tag{12.25}$$

Hence, for all $(x,1) \in X$, we have

$$\begin{aligned}
\|x-\mu_t\|_2 &\geq \mathtt{dist}(\mu_t, \mathfrak{M}) - \|x-\mathfrak{m}_l\|_2 && \text{(triangle inequality, } \mathfrak{m}_l \in \mathfrak{M}) \\
&\geq \sqrt{4b \cdot \mathrm{km}_X(\mathfrak{M})} - \|x-\mathfrak{m}_l\|_2 && \text{(Equation (12.25))} \\
&\geq \sqrt{4b \cdot \mathrm{km}_X(\mathfrak{M})} - \sqrt{\mathrm{km}_X(\mathfrak{M})} && ((x,1) \in X) \\
&= \left(2\sqrt{b}-1\right) \cdot \sqrt{\mathrm{km}_X(\mathfrak{M})} \\
&\geq \sqrt{b} \cdot \sqrt{\mathrm{km}_X(\mathfrak{M})} \,. && (b \geq 1)
\end{aligned}$$

Combining these inequalities gives

$$\phi_X^{(r)}(C) \geq \sum_{n=1}^{N} r(p_{nt}) \|x-\mu_t\|_2^2 \geq \left(\sum_{n=1}^{N} r(p_{nt})\right) b \cdot \mathrm{km}_X(\mathfrak{M}) = \frac{\sum_{n=1}^{N} r(p_{nt})}{r\left(\frac{\epsilon}{2\mathbf{i}_r K^2}\right)} \cdot \frac{20}{\tilde{\epsilon}^2} \mathrm{km}_X(\mathfrak{M}) \,.$$

Finally, observe that due to (12.16) there exists some $n(t) \in [N]$ such that $p_{n(t)t} \geq \frac{\epsilon}{2\mathbf{i}_r K^2}$. Since a fuzzifier function is non-negative and strictly increasing, we can conclude that $\sum_{n=1}^{N} r(p_{nt}) \geq r(p_{n(t)t}) \geq r\left(\frac{\epsilon}{2\mathbf{i}_r K^2}\right)$. This yields the claim. □

A combination of these results yields the desired bound:

Claim 12.25. *If $C \not\subseteq \mathfrak{U}$, then*

$$\left|\phi_X^{(r)}(C) - \phi_S^{(r)}(C)\right| \leq \epsilon \cdot \phi_X^{(r)}(C) \,.$$

Proof. First, observe that, due to Claim 12.23 and Claim 12.24, we have

$$\sum_{n=1}^{N} \|x_n - \mathfrak{s}(x_n)\|_2^2 \leq 20 \cdot \mathrm{km}_X(\mathfrak{M}) \leq \tilde{\epsilon}^2 \cdot \phi_X^{(r)}(C) \,.$$

With Claim 12.22, we can conclude that

$$\begin{aligned}
\left|\phi_X^{(r)}(C) - \phi_S^{(r)}(C)\right| &\leq \left(1+\frac{1}{\tilde{\epsilon}}\right) \sum_{n=1}^{N} \|x_n - \mathfrak{s}(x_n)\|_2^2 + \tilde{\epsilon} \cdot \min\left\{\phi_S^{(r)}(C), \phi_X^{(r)}(C)\right\} \\
&\leq \left(1+\frac{1}{\tilde{\epsilon}}\right) \cdot \tilde{\epsilon}^2 \cdot \phi_X^{(r)}(C) + \tilde{\epsilon} \cdot \phi_X^{(r)}(C) \\
&= \left(\tilde{\epsilon}^2 + 2\tilde{\epsilon}\right) \phi_X^{(r)}(C) \leq \epsilon \phi_X^{(r)}(C) \,. && (\tilde{\epsilon} \leq \epsilon/3)
\end{aligned}$$

□

Conclusion

Combining the results of both cases yields the correctness of the approximation bound stated in Theorem 12.16.

12.5.6 Size of S and Runtime

Consider the setting from Theorem 12.2. From Observation 12.12, we know that

$$|S| \in \mathcal{O}\left(\log(|X|)\log(\log(|X|)) \cdot \mathbf{c}_r(K)^{-2} \cdot K^2 \cdot \epsilon^{-2} \cdot \log(\gamma) \cdot \log(\delta^{-1})\right) \text{ , where}$$

$$\gamma = \beta K \cdot \left(\frac{1}{2}\log\left(9\alpha|X| \cdot \frac{20}{\tilde{\epsilon}^2 r\left(\frac{\epsilon}{2 \cdot \mathbf{i}_r K^2}\right)}\right) + 1\right) \cdot \left(\frac{1692}{\tilde{\epsilon}}\right)^D \text{ , } \tilde{\epsilon} = \frac{\epsilon \cdot \mathbf{c}_r(K)}{504\alpha} \text{ , and } \alpha, \beta \in \mathcal{O}(1).$$

Hence,

$$\begin{aligned}\log(\gamma) &\in \mathcal{O}\left(\log(K)+\log\left(\log(|X|)+\log\left(\frac{1}{\tilde{\epsilon}^2}\right)+\log\left(r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)^{-1}\right)\right)+D\log\left(\frac{1}{\tilde{\epsilon}}\right)\right)\\ &\subseteq \mathcal{O}\left(\log(K)+\log(\log(|X|))+\log\log\left(r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)^{-1}\right)+D\log\left(\frac{1}{\epsilon \mathbf{c}_r(K)}\right)\right)\\ &\subseteq \mathcal{O}\left(\log(K)\cdot\log(\log(|X|))\cdot D\cdot\log\left(\frac{1}{\epsilon}\right)\cdot\log\log\left(r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)^{-1}\right)\cdot\log\left(\frac{1}{\mathbf{c}_r(K)}\right)\right).\end{aligned}$$

Putting these bounds together gives the desired bound

$$|S| \in \mathcal{O}\left(\log(|X|)\log(\log(|X|))^2\cdot K^3\cdot D\cdot\epsilon^{-3}\cdot\log\log\left(\frac{1}{r\left(\frac{\epsilon}{2\cdot \mathbf{i}_r K^2}\right)}\right)\cdot \mathbf{c}_r(K)^{-3}\cdot\log(\delta^{-1})\right).$$

Next, consider the runtime. From Observation 12.12, we know that the time needed to apply Algorithm 13 is $\mathcal{O}(|X|DK+|S|)$. The time needed to apply the (α,β)-approximation algorithm from (Aggarwal et al., 2009) is just $\mathcal{O}(|X|DK\log(1/\delta))$. A combination of these observations yields the claim on the runtime.

12.5.7 These Weak Coresets Are Not Weak

With a little additional analysis of the chosen approximate mean sets, it is easy to see that our weak coresets are actually strong coresets. In the following, we first give an overview of the main arguments and an informal outline of the proof. After that, we provide a formal proof of Theorem 12.2.

Recall that our weak coresets guarantee the coreset property only with respect to a restricted set of solutions:

Observation 12.26 (weak coreset)**.** *If $(S,\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X))$ is a weak ϵ-coreset of X for the r-fuzzy K-means problem, then*

$$\forall \tilde{C}\in\Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X):\ \phi_S^{(r)}(\tilde{C})\in[1\pm\epsilon]\phi_X^{(r)}(\tilde{C})\,. \tag{12.26}$$

Recall from Definition 12.6 that $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(Y)$ is the set of solutions that do not contain negligible clusters with respect to Y. So this restriction is rather negligible in regard to Y:

Observation 12.27 (negligible)**.** *Due to Theorem 6.3, for all $Y\in\mathrm{Dom}\left(\mathbb{R}^D,\mathbb{R}_+\right)$ and all $C\in(\mathbb{R}^D)^{\leq K}$, we have*

$$\exists C'\subseteq C:\ \ C'\in\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(Y)\ \ \textit{and}\ \ \phi_Y^{(r)}(C')\leq(1+\epsilon)\cdot\phi_Y^{(r)}(C)\,.$$

Consider an arbitrary solution $C\in(\mathbb{R}^D)^{\leq K}$. This observation tells us that, for a given data set X as well as the presumed coreset S, there exist solutions $C'_X, C'_S\subseteq C$ with $C'_X\in\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ and $C'_S\in\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(S)$ whose costs are not much worse than the costs of C with respect to X and S, respectively.

Moreover, we already know from Observation 12.26 that the costs of $C'_X\in\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$ with respect to X and with respect to S do not differ much. We do not know yet how the cost of $C'_S\in\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(S)$ with respect to X and with respect to S relate to each other. Fortunately, due to the way S is constructed, we know the following:

Lemma 12.28 (transitivity (due to construction))**.** *For all $X,S\in\mathrm{Dom}\left(\mathbb{R}^D,\mathbb{R}_+\right)$ we have*

$$S\in\mathrm{Dom}(\{x\mid(x,w)\in X\},\mathbb{R}_+)\quad\Rightarrow\quad\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(S)\subseteq\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)\,.$$

Proof. Let $S = ((s_m, v_m))_{m\in[M]}$ be some data set. Let $X = ((x_n, w_n))_{n\in[N]}$ be a data set such that $S \in \mathrm{Dom}(\{x \mid (x,w) \in X\}, \mathbb{R}_+)$. This means that, for each $m \in [M]$, there exists an $n(m) \in [N]$ such that $s_m = x_{n(m)}$.

Recall from Lemma 5.17 that an r-fuzzy clustering of a data set that is induced by some means consists of soft clusterings of the single data points that depend only on the respective point and the given means. In particular, the soft assignments of the single data points do not depend on the weight of the data points (Corollary 5.18). Hence, there exists an r-fuzzy L-clustering $P = (p_{nk})_{n\in[N],k\in[K]}$ of X induced by C and an r-fuzzy clustering $\tilde{P} = (\tilde{p}_{nk})_{n\in[N],k\in[K]}$ of S induced by C such that, for all $(x_n, w_n) \in X$ and $(s_m, v_m) \in S$ with $x_n = s_m$ we have $p_{nk} = \tilde{p}_{mk}$. With our first observation, we can conclude that, for each $m \in [M]$, there exists an index $n(m) \in [N]$ such that $\forall k \in [K]\colon\ p_{mk} = \tilde{p}_{n(m)k}$.

Let $c \in \mathbb{R}_+$ be some constant. Assume that, for each $k \in [K]$, there exists some $m(k) \in [M]$ such that $\tilde{p}_{m(k)k} \geq c$. Then, for each $k \in [K]$, we have $p_{n(m(k))k} = \tilde{p}_{m(k)k} \geq c$. □

Hence, $\Theta_{(r,\mathbf{i}_r,K,\epsilon)}(S) \subseteq \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(X)$. So, due to Observation 12.26, we also know that the costs of $C'_S \in \Theta_{(r,\mathbf{i}_r,K,\epsilon)}(S)$ with respect to X and with respect to S do not differ much.

Now it remains to analyse how the cost with respect to X changes when we replace C by C'_S and how the cost with respect to S changes when we replace C by C'_X. Here, our knowledge on the specific form of C'_S and C'_X comes in handy. Recall that both of these sets are subsets of C.

Observation 12.29 (monotonicity)**.** *Due to Lemma 5.11, for all $C', C \in (\mathbb{R}^D)^{\leq K}$, we have*

$$C' \subseteq C \quad \Rightarrow \quad \forall Z \in \mathrm{Dom}\left(\mathbb{R}^D, \mathbb{R}_+\right)\colon\ \phi_Z^{(r)}(C') \geq \phi_Z^{(r)}(C)\,.$$

Hence, $\phi_X^{(r)}(C'_S) \geq \phi_X^{(r)}(C)$ and $\phi_S^{(r)}(C'_X) \geq \phi_S^{(r)}(C)$. Given these observations, we can now prove the main result of this chapter.

Proof of Theorem 12.2. Let $\tilde{\epsilon} = \epsilon/4$. From Theorem 12.16, we know that, with probability $1-\delta$, the tuple $(S, \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X))$ is a weak ϵ-coreset of X for the r-fuzzy K-means problem. In the following, we assume that this is the case. This means that

$$\forall \tilde{C} \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X)\colon\ \phi_S^{(r)}(\tilde{C}) \in [1 \pm \epsilon]\phi_X^{(r)}(\tilde{C})\,. \tag{12.27}$$

Now consider an arbitrary but fixed $C \in (\mathbb{R}^D)^{\leq K} \setminus \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}$. First, we prove the upper bound. From Observation 12.27, we know there exists a $C' \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X)$ with $\phi_X^{(r)}(C') \leq (1+\tilde{\epsilon})\phi_X^{(r)}(C)$. Thus, we have

$$\begin{aligned}
\phi_S^{(r)}(C) &\leq \phi_S^{(r)}(C') && (C' \subseteq C \text{ and Observation 12.29})\\
&\leq (1+\tilde{\epsilon})\phi_X^{(r)}(C') && (C' \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X) \text{ and (12.27)})\\
&\leq (1+\tilde{\epsilon})^2\phi_X^{(r)}(C) && (\text{choice of } C')\\
&= (1+\epsilon)\phi_X^{(r)}(C)\,. && (\tilde{\epsilon} = \epsilon/4)
\end{aligned}$$

Next, we prove the lower bound. From Observation 12.27, we know there exists an $C'' \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(S)$ with $\phi_S^{(r)}(C'') \leq (1+\tilde{\epsilon})\phi_S^{(r)}(C)$. Due to Lemma 12.28, we know that $C'' \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X) \cap \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(S)$. Hence,

$$\begin{aligned}
\phi_S^{(r)}(C) &\geq \frac{1}{1+\tilde{\epsilon}}\phi_S^{(r)}(C'') && (\text{choice of } C'')\\
&\geq \frac{1-\tilde{\epsilon}}{1+\tilde{\epsilon}}\phi_X^{(r)}(C'') && (C'' \in \Theta_{(r,\mathbf{i}_r,K,\tilde{\epsilon})}(X) \text{ and (12.27)})\\
&\geq \frac{1-\tilde{\epsilon}}{1+\tilde{\epsilon}}\phi_X^{(r)}(C) && (C'' \subseteq C \text{ and Observation 12.29})\\
&\geq (1-4\tilde{\epsilon})\phi_X^{(r)}(C) = (1-\epsilon)\phi_X^{(r)}(C)\,. && (\tilde{\epsilon} = \epsilon/4)
\end{aligned}$$

□

"Hypotheses are what we lack the least."

Henri Poincaré[1]

Chapter 13

Summary & Conclusion

In this chapter, we first review the main steps of our analysis and give an overview of our approximation algorithms. Then we discuss our results and suggest ideas for future work.

13.1 Review

Basically, we derived our notion of fuzzifier functions r with the goal that the resulting r-fuzzy K-means clustering shares basic properties with a K-means clustering (Section 5.2.2). We found that our fuzzifier functions guarantee two particularly useful properties: First, there is a connection between the r-fuzzy K-means and the K-means objective function. More precisely, we showed that objective values of solutions induced by the same means differ by at most a factor $\mathbf{c}_r^*(K)$, which only depends on the number of clusters K and the fuzzifier r (Section 6.1). Second, the notion of an empty hard cluster has a counterpart in r-fuzzy K-means clustering. If a hard cluster is empty, then it is not supported by any point at all, and so we can effectively remove its mean from the hard clustering without changing its K-means cost. If an r-fuzzy cluster has negligible support, then there is no point that supports this cluster sufficiently, and so we can remove the mean of this cluster without significantly increasing the overall r-fuzzy K-means cost (Section 6.2).

Besides these two properties, we found that an r-fuzzy K-means clustering exhibits a locality property. Points are more assigned to means nearby than they are assigned to means far away (Section 5.2.2). However, this locality property is inherently *soft* and so we completely lose the notion of a hard clustering (Section 4.2.2). In Chapter 8 we tried to close this gap between r-fuzzy clusterings and hard clusterings via a Monte Carlo method. By exploiting the probabilistic interpretation of r-fuzzy clusterings, we showed that there exist hard clusters whose statistics are similar to those of the r-fuzzy clusters. Unfortunately, apart from the probabilistic interpretation, these hard clusters do not exhibit useful structural properties: They do not exhibit any locality property, their convex hulls might overlap, and they do not even cover the whole data set.

Nevertheless, due to our identification of all these properties of the r-fuzzy K-means problem, we were able to apply algorithmic techniques that are known from K-means clustering: the algorithm of Hasegawa et al. (1993), the coreset construction by Chen (2009), the notion of ϵ-approximate mean sets by Matoušek (2000), and the superset sampling technique used by Ackermann et al. (2010). With slight adaptions (e.g. a larger sampling rates and evaluation of a different cost function) and a more detailed analysis (e.g. the duplication of points), we were able to show that these techniques can be used to compute approximations and coresets with respect to the r-fuzzy K-means problem as well. This shows that these techniques primarily rely on the form of the K-means cost function and

[1] Source: Henri Poincaré, Science and Hypothesis (1905)

the notion of empty clusters. For instance, the superset sampling technique is a clever way of applying uniform sampling to identify candidates for the (mean) statistics of (nearly) arbitrary hard clusters. In contrast, we were not able to apply the K-means++ algorithm by Arthur and Vassilvitskii (2007) as this technique relies on much more specific locality properties of K-means *hard* clusterings.

13.2 Overview of Our Algorithms

First, let us briefly compare our algorithms with the corresponding algorithm for the K-means problem: Algorithm 4 is the basically the same as the algorithm by Hasegawa et al. (1993). It also computes a 2-approximation and has runtime $\mathcal{O}(|X|^{K+1}DK)$. Algorithm 8 uses the superset sampling technique introduced by Inaba et al. (1994), whose result was enhanced by Kumar et al. (2004). Its runtime is slightly better than that the runtime $|X|^{\mathcal{O}(DK)}$ of the *exact* algorithm by Inaba et al. (1994) and slightly worse than the runtime $\mathcal{O}(|X|D2^{\text{poly}(K/\epsilon)})$ of the approximation algorithm by Kumar et al. (2004). Recall that Algorithm 9 does not guarantee a proper approximation bound. Algorithm 14 follows the idea of the algorithm by Chen (2009), whose runtime of $\mathcal{O}(|X|D)+D\cdot\text{poly}(1/\epsilon)+2^{\tilde{\mathcal{O}}(K/\epsilon)}$ is slightly better than that of our algorithm.

Next, let us consider the dependencies on the notions from K-means clustering and on the properties of the fuzzifier function: The simplest method, Algorithm 4, does not rely on the additional properties that we introduced for fuzzifier functions. Our approach that constructs a set of soft clusterings (Algorithm 5) relies on the increase-bounded property as, simply speaking, it distorts soft assignments. All the algorithms that are based on the work by Chen (2009) heavily depend on notions from K-means clustering. In particular, Algorithm 14 depends on a bound $\mathbf{c}_r(K)$ on the minimum contribution, which relates the r-fuzzy K-means objective function to the K-means objective function. In contrast, Algorithm 8, which is based on the superset sampling technique, does not depend on the minimum contribution bound $\mathbf{c}_r(K)$.

13.3 Discussion

We have generalized the fuzzy K-means problem and identified a large class of related problems, the so-called r-fuzzy K-means problems. Despite the fact that all of these problems search for soft clusterings, we have shown that there are some aspects that are similar to the K-means hard clustering problem. We have identified and characterized these aspects. They justified the application of techniques known from K-means clustering. Thereby, we derived the very first algorithms for the fuzzy K-means problem with approximation guarantees.

However, unsupervised clustering is about identifying unknown structures in data sets. This raises the following questions: What are the structural differences between a $(1+\epsilon)$-approximation to the r-fuzzy K-means problem and a $(1+\epsilon)$-approximation to the K-means problem? How can we describe these differences? Is there an algorithm that exploits these structural properties? That is, is there an algorithm that does not generate nearly the same set of candidate solutions and just uses a different cost function to evaluate these candidates? On a slightly different note, can we identify properties of the fuzzy K-means algorithm that explains why it is used in practise? It is the use of this algorithm that motivated us to consider the fuzzy K-means problem after all.

Besides that, in our analysis it becomes apparent that we are missing statistical assumptions and an interpretation of the fuzzified soft assignments. Recall that the latter gap led us to (and was our one and only motivation for) the definition of probabilistic membership values (Section 3.3.1). In particular, it is the reason why our soft-to-hard-cluster technique

reference	technique	randomized[c]	approximation factor	runtime wrt. $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$[a] and p_m-fuzzy[d] K-means	depends[b] on … $\mathbf{i}_r$	$\mathbf{c}_r$	$\frac{\mathrm{w}^{(X)}_{\max}}{\mathrm{w}^{(X)}_{\min}}$	reducing
Algorithm 4	means from X		2	$\lvert X\rvert^{K+1} \cdot DK$				
Theorem 7.4	K-means c-approximation[e]	–	$c \cdot \mathbf{c}_r^{-1}(K)$	–			–	
Algorithm 5	round assignments		$(1+\epsilon)$	$D \cdot 2^{\mathcal{O}(\lvert X\rvert K m \epsilon)}$	✗			
Algorithm 8	superset sampling		$(1+\epsilon)$	$D \cdot \lvert X\rvert^{\mathcal{O}(K^3 \epsilon^{-2} m^2)}$	✗		✗	✗
Algorithm 9	superset sampling	✗	–[f]	$\lvert X\rvert \cdot D \cdot 2^{\mathcal{O}(K^2 m \epsilon^{-1} \alpha^{-1})}$	✗		✗	✗
Corollary 8.12	superset sampling	✗	$(1+\epsilon)$	$D \cdot \lvert X\rvert^{\mathcal{O}(K^2 \epsilon^{-2} m^2)}$	✗		✗	✗
Algorithm 10	discretize means		$(1+\epsilon)$	$\lvert X\rvert \cdot \log(\lvert X\rvert)^K \cdot 2^{\mathcal{O}(K^2 \mathbf{D} \epsilon^{-1} m)}$	✗	✗	✗	✗
Algorithm 11	+ dimension reduction	✗	$(1+\epsilon)$	$\mathbf{D} \cdot \lvert X\rvert^{\mathcal{O}(K^2 \epsilon^{-3} m)}$	✗	✗	✗	✗
Algorithm 14	+ coresets	✗	$(1+\epsilon)$	$\lvert X\rvert \cdot (\log(\lvert X\rvert) \cdot \mathbf{D})^{\mathcal{O}(K^2 \epsilon^{-3} m)}$	✗	✗	–[g]	✗

Table 13.1: Overview of our approximation algorithms.

[a] For the sake of simplicity, we only state the runtime of the algorithms with respect to *un-weighted* data sets $X \in \mathrm{Dom}\left(\mathbb{R}^D, \{1\}\right)$.

[b] "✗" marks an algorithm that depends on the respective quantity.

[c] "✗" marks a randomized algorithm (with a constant probability of success).

[d] For the fuzzifier function p_m with $m \in (1, \infty)$, we have $\mathbf{c}_{\mathrm{p}_m}(K) = 1/K^{m-1}$ and $\mathbf{i}_{\mathrm{p}_m} = 4m$.

[e] "–" marks information that depends on the choice of the concrete c-approximation K-means algorithm.

[f] There is no proper guarantee. For more information, we refer to Section 8.6.2.

[g] Algorithm 14 can only be applied to unweighted data sets.

cannot guarantee the existence of a hard *clustering* that imitates an r-fuzzy clustering well (Chapter 3).

In Section 4.4 we already pointed out the current line of work that deals with the notion of clusterability and the idea that clustering is difficult only when it does not matter. With this in mind, one can sum up the previous questions as follows: When is a data set r-fuzzy-K-means-clusterable?

13.4 Future Work

First of all, one can tackle the general problem criticized in the previous section. As explained in Section 4.4, there is no definition of clusterability that is agreed upon and, to the best of our knowledge, these definitions focus on hard clusterings. Nonetheless, we propose to study the notion of clusterability with respect to soft K-means problems. In particular, the derivation of algorithms that are more specific to the r-fuzzy K-means problem might be a step towards the analysis of clusterability and a better understanding of fuzzy clusterings. Such algorithms might exist for relaxed versions of the r-fuzzy K-means approximation problem. Therefore, we propose to pursue the study of constant factor and constant bi-factor approximation algorithms. This is not to say that these algorithms are less useful or less meaningful. For instance, the K-means++ algorithm by Arthur and Vassilvitskii (2007) is very popular in practise although it (is only proven that it) yields a $\mathcal{O}(\log(K))$-approximation in expectation. Moreover, the "correct" number of clusters is hardly ever known. Hence, allowing for a larger number of clusters might, in practise, be more appealing than allowing for a poor approximation factor for the presumed correct number of clusters. Besides that, we propose to study the properties of the fuzzy K-means algorithm in order to explain its popularity and help us understand the fuzzy K-means clustering approach. Last but not least, we presume that the K-means++ algorithm also works provably well with respect to the fuzzy K-means problem.

Part III

Clustering with Gaussian Mixture Models

"Everyone believes it, however, as M. Lippmann told me one day, because the experimenters imagine it is a mathematical theorem, and the mathematicians that it is an experimental fact."

Henri Poincare[1]

Chapter 14

Introduction

Training the parameters of statistical models to describe a given data set is a central task in the field of data mining and machine learning. In the following, we give a short introduction to the topic. For more information we refer to the work of Bilmes (1998), Mackay (2003), Bishop (2006), and Cover and Thomas (2006).

14.1 Gaussian Mixture Models (GMMs)

Among the most widely used families of statistical models are mixture models, especially, mixtures of Gaussian distributions.

14.1.1 Density Function

Gaussian (or normal) distributions are probably the most common distributions used in natural sciences.

Definition 14.1 (Gaussian)**.** *The probability density function* $\mathcal{N}_D(\cdot|\mu,\Sigma)\colon \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ *of a* D*-variate Gaussian distribution with mean* $\mu \in \mathbb{R}^D$ *and covariance* $\Sigma \in \mathbb{R}^{D\times D}$ *is given by*

$$\mathcal{N}_D(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

where $|\Sigma|$ *denotes the determinant of the matrix* $\Sigma \in \mathbb{R}^{D\times D}$.

This distribution is non-degenerated if the covariance Σ *is non-degenerated, i.e., , symmetric and positive definite). If* Σ *takes the form* $\sigma^2 I_D$*, where* I_D *denotes the* $(D\times D)$*-identity matrix and* $\sigma^2 \in (0,\infty)$*, then we call this matrix and the distribution* spherical.

In this thesis, we consider Gaussian mixture models where the number of components $K \in \mathbb{N}$ is predefined and where each of the Gaussian components is described by its own mean and covariance.

Definition 14.2 (GMM)**.** *The probability density function* $\mathrm{p}(\cdot|\theta)\colon \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ *of a* D*-variate Gaussian mixture model (GMM) with parameters* $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$*, where* $(w_k)_{k\in[K]} \in \Delta_{K-1}$*,* $\mu_k \in \mathbb{R}^D$ *and* $\Sigma_k \in \mathbb{R}^{D\times D}$ *for all* $k \in [K]$*, is given by*

$$\mathrm{p}(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}_D(x|\mu_k,\Sigma_k)\ .$$

It is non-degenerated if none of its components is degenerated.

[1] Source: Duplantier and Rivasseau (2015). Henri Poincaré, 1912-2012. Poincaré Seminar 2012. ISBN: 978-3-0348-0834-7. (p. 187)

Whenever we talk about covariance matrices or Gaussian mixtures, we will implicitly assume that they are non-degenerated, unless stated otherwise. For the sake of simplicity, we write $\mathcal{N}_D(\mu,\Sigma)$ to denote the function $\mathcal{N}_D(\cdot|\mu,\Sigma)$. Moreover, we refer to a GMM with parameters θ simply as the GMM θ.

Besides that, as with data sets, the indexation of the Gaussian components does not matter. That is, given a permutation π of $[K]$, the densities $\mathrm{p}(\cdot|\theta)$ and $\mathrm{p}(\cdot|\theta_\pi)$ with $\theta_\pi = \big((w_k,\mu_k,\Sigma_k^2)\big)_{k\in(\pi(1),\dots,\pi(K))}$ are the same. Nonetheless, to keep our notation uncluttered, we stick with our vector notation from Section 2.1.

14.1.2 Generating Observations

When we draw an observation according to a Gaussian mixture model (GMM), we implicitly follow a two-step process.

Process. Drawing an observation X_n according to a GMM $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$ can be described as follows:

1. Sample an indicator vector $Z_n = (Z_{nk})_{k\in[K]}$ with $\sum_{k=1}^K Z_{nk} = 1$ from $[0,1]^K$ according to $\Pr(Z_{nk}=1) = w_k$.
2. Sample an observation X_n from $\mathbb{R}^D$ according to the k-th component $\mathcal{N}_D(\mu_k,\Sigma_k)$ with the index $k\in[K]$ that satisfies $Z_{nk}=1$.

When we draw N observations $X = (X_n)_{n\in[N]}$ according to a GMM θ, then we draw each observation X_n independently according to this process described above. Consequently, the set of random variables $\{X_n\}_{n\in[N]}$ is mutually independent.

By the conditional function $\mathrm{p}(Y|\theta')$ we denote the probability density function of Y under the assumption that the random variables $X_1,\dots,X_N$ and $Z_1,\dots,Z_N$ (on which Y usually depends) have been generated according to the process described by θ'.

That is, for $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$, we write

$$\begin{aligned} \mathrm{p}(Z_{nk}=1|\theta) &= w_k\ , \\ \mathrm{p}(X_n,Z_{nk}=1|\theta) &= \mathcal{N}_D(X_n|\mu_k,\Sigma_k)\ , \text{ and} \\ \mathrm{p}(X_n|\theta) &= \textstyle\sum_{k=1}^K \mathrm{p}(Z_{nk}=1|\theta)\cdot\mathrm{p}(X_n,Z_{nk}=1|\theta)\ . \end{aligned} \tag{14.1}$$

Moreover, due to the independence of the observations, we can write

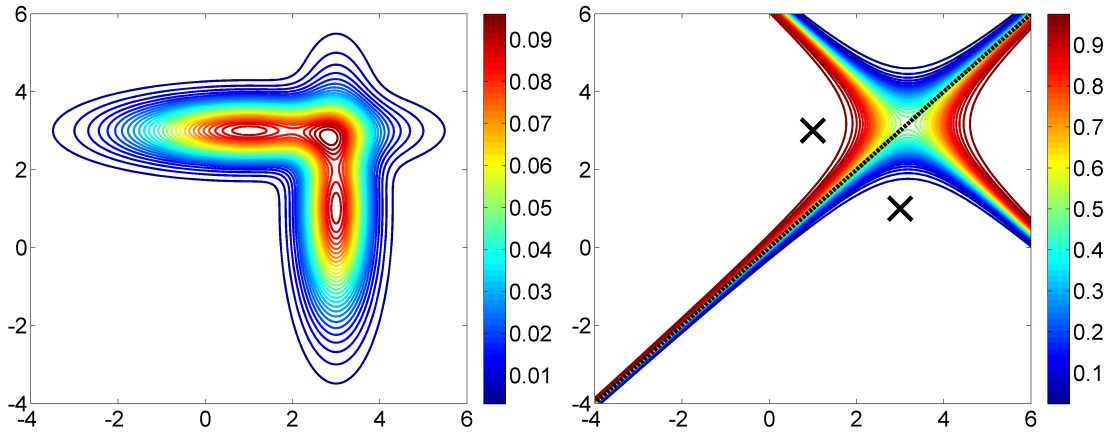
$$\mathrm{p}(X|\theta) = \textstyle\prod_{n=1}^N \mathrm{p}(X_n|\theta) \quad\text{and}\quad \mathrm{p}(X,Z|\theta) = \prod_{n=1}^N \mathrm{p}(X_n,Z_n|\theta)\ .$$

Hidden Variables (Hard Assignments). As we only observe the value $x_n\in\mathbb{R}^D$ that X_n takes, the indicator $(Z_{nk})_{k\in[K]}$ is called a hidden (or latent) random variable in this two-step process. Likewise, we call the realization z_n of Z_n a hidden variable. Observe that the matrix $(Z_{nk})_{n\in[N],k\in[K]}$ describes a hard K-clustering of X.

Posterior Probabilities (Soft Assignments). Given an observation $X_n = x_n\in\mathbb{R}^D$ that has been generated according to a GMM $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$, we can compute the posterior probability that the k-th component of θ has generated this observation during the second step of the process. This probability is given by

$$\begin{aligned} p_{nk} \coloneqq \mathrm{p}(Z_{nk}=1|X_n=x_n,\theta) &= \frac{\mathrm{p}(Z_{nk}=1|\theta)\cdot\mathrm{p}(X_n=x_n|Z_{nk}=1,\theta)}{\mathrm{p}(X_n=x_n|\theta)} && \text{(Bayes')} \\ &= \frac{w_k\cdot\mathcal{N}_D(x_n|\mu_k,\Sigma_k)}{\mathrm{p}(x_n|\theta)}\ , && (14.2) \end{aligned}$$

where the last equality is due to (14.1). We call $(p_{nk})_{n,k}\in\Delta_{N,K-1}$ the soft clustering of X induced by the GMM θ.



(a) Contour plot of the density of the GMM θ.

(b) Contour plot of the soft assignments p_{n1}.

Figure 14.1: Consider the GMM $\theta = \big((0.5, \mu_1, \Sigma_1), (0.5, \mu_2, \Sigma_2)\big)$ where $\mu_1 = (1\ 3)^T$, $\mu_2 = (3\ 1)^T$, Σ_1 is a diagonal matrix with entries 3 and 0.25, and where Σ_2 is a diagonal matrix with entries 0.25 and 3. In Figure 14.1a, we see that the density actually has more modes than the mixture has components (at $(1,3)$, $(3,1)$, and $(3,3)$). Figure 14.1b depicts a contour plot of the posterior probability $p_{n1} := \mathrm{p}(Z_{nk} = 1 | \theta, X_n = x_n) = 0.5 \cdot \mathcal{N}_2(x_n | \mu_1, \Sigma_1) / \mathrm{p}(x_n | \theta)$ for each $x_n \in [-4,6]^2$. The two crosses mark the means μ_1 and μ_2 and the dashed line marks the perpendicular bisector of the line segment between these means. There are points (e.g. $(6,2)^T$) that are closer to μ_2 but whose soft assignment to μ_1 is larger than 0.5.

14.1.3 Remarks

Unbounded. A density is no probability. It can take values larger than 1. In fact, it can even be arbitrarily large: Consider a Gaussian distribution $\mathcal{N}_D(x, \sigma^2 \cdot I_D)$ with some $x \in \mathbb{R}^D$ and $\sigma^2 > 0$. For $\sigma^2 > 0$, this density is non-degenerated. Nonetheless, if σ converges to 0, then the Gaussian density at x diverges to ∞.

Observation 14.3. *For $|\Sigma| \to 0^+$ with $\Sigma \neq 0_{D,D}$, we have $\mathcal{N}_D(x|x, \Sigma) = \frac{1}{(2\pi)^D |\Sigma|^{1/2}} \cdot 1 \to \infty$.*

Highly Multi-Modal. A mixture with $K \geq 2$ Gaussian components does not necessarily have K modes. It can have more or less than K modes. An example for a fairly simple mixture of 2 Gaussians with 3 modes is given in Figure 14.1a.

No Locality (Soft Assignments). Consider the soft assignment $(p_{nk})_{k \in [K]}$ of some point $x_n \in \mathbb{R}^D$ induced by some GMM θ, which we derived in (14.2). Depending on the covariances of θ, this soft assignment does *not* necessarily assign x_n more to a cluster whose mean is closer than to a cluster whose mean is farther away. An example is depicted in Figure 14.1b. Hence, the soft clusterings induced by GMMs are inherently different from the r-fuzzy K-means clusterings, which we considered in Section 4.2.1.

No Monotonicity. In an r-fuzzy K-means clustering, adding another representative to a solution is never a bad idea. If we add a mean vector to a set of mean vectors, then the r-fuzzy K-means cost of this set does not increase (Lemma 5.11). This observation cannot be directly transferred to the clustering with GMMs.

An obvious example that illustrates this fact is the following: Consider an arbitrary GMM θ and observations X. Add a new component $(\mu_{K+1}, \Sigma_{K+1})$ to θ with weight $w_{K+1} \approx 1$ and reduce the weights of the other components accordingly. Let the mean μ_{K+1} be a point that lies farther away from all points in X than all means in θ. Choose a sufficiently small

covariance matrix Σ_{K+1} (i.e., $|\Sigma_{K+1}|$ close enough to zero). Then, the likelihood of the resulting GMM is much smaller than the likelihood of the original GMM θ.

No Identifiability. Even if we were able to evaluate the density of an unknown GMM at every point, we could still not identify the correct number of its components. As an example, consider a mixture $\theta_2 = \big((w_1,\mu,\sigma_1 \cdot I_D),(w_2,\mu,\sigma_2 \cdot I_D)\big)$ where $\sigma_1^2 \neq \sigma_2^2$ and $w_1, w_2 > 0$. Then, $\mathrm{p}(x|\theta) = \mathcal{N}_D(x|\mu,(\sigma_1^2+\sigma_2^2)/2 \cdot I_D)$. Hence, there is no difference between drawing points from the mixture θ_2 with 2 different components and the single Gaussian distribution $\mathcal{N}_D(\mu,(\sigma_1^2+\sigma_2^2)/2 \cdot I_D)$ (i.e., a mixture with one component). This also means that given only the resulting observations X, we cannot identify the exact parameters that have been used to create X. Besides that, as already pointed out in Section 14.1.1, the indexation of the Gaussian components is not identifiable.

Curse & Blessing of High-Dimensionality (Sampling). In high dimension, the probability mass of a Gaussian density is concentrated in a thin shell around the mean (Hopcroft and Kannan, 2017): Nearly all of the probability mass of a D-variate Gaussian with spherical covariance matrix $\sigma^2 I_D$ is contained in a shell around the mean which has radius $r = \sqrt{D}\sigma$ and whose thickness is proportional to $r/\sqrt{D}$. For example, for $D = 1000$ and $\sigma = 1$, 90% of the probability mass is contained in a shell with the radius $r = 31.6$ and thickness 2.8 (Mackay, 2003, p. 309). Nevertheless, the density itself is still largest at the mean vector. In the example, the density at the mean is $e^{1000/2} \approx 10^{217}$ times larger than the density of a point contained in the shell. This unintuitive behaviour is usually collectively referred to as the curse of dimensionality. However, as the tight concentration of points in the shell can be a very useful property, it is also referred to as a blessing. For instance, Anderson et al. (2014) use this fact to derive an algorithm that determines the means of an unknown Gaussian mixture in high dimension.

14.2 Likelihood Approach

We are given a set of observations $X = (x_n)_{n\in[N]}$ and presume that these observations have been drawn according to a GMM θ^*. Our goal is to estimate the parameters of the GMM θ^*. There are various approaches to estimate such parameters. In this thesis, we focus on the method of maximum likelihood estimation. This method estimates θ^* by the GMM θ_{ML} which has most likely generated X.

In the following, we formalize the notion of likelihood, show how likelihoods can be compared with one another, and discuss this approach in general. In Section 14.3, we deal with methods for finding the most likely solution θ_{ML}.

14.2.1 Likelihood

We are given some observations $X = (x_n)_{n\in[N]} \subseteq \mathbb{R}^D$. We assume that each x_n has been generated independently according to the two-step process described by some unknown GMM θ^*. Under this assumption, consider some arbitrary but fixed GMM θ. How likely is it that θ has generated the observations X?

One can measure this likelihood in terms of the density of the observations under the (assumed) distribution θ.

Definition 14.4 (likelihood)**.** *Let $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$ be a D-variate GMM and let $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$. The likelihood of θ, given X, is*

$$\mathcal{L}_X(\theta) := \prod_{n=1}^{N} \mathrm{p}(x_n|\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} w_k \mathcal{N}_D(x_n|\mu_k,\Sigma_k) \,.$$

This definition might seem a bit cumbersome as it just gives the joint density $p(X|\theta)$ a different name. Yet, it marks a change in the paradigm: We consider a function in the model parameters.

Log-Likelihood. Often, it is easier to work with the natural logarithm of the likelihood instead of the likelihood. We use the short notation "log-likelihood" to refer to the natural logarithm of a likelihood.

14.2.2 Likelihood Ratio

Assume we are given two GMMs θ_1 and θ_2. We prefer one GMM over the other if its likelihood is larger. However, they might still be very similar to each other. How do we measure the difference of the quality of two GMMs?

To this end, one considers the ratio between these likelihoods.

Definition 14.5 (likelihood ratio)**.** *The likelihood ratio between the GMM θ_1 and the GMM θ_2, given observations X, is given by*

$$\Lambda_X(\theta_1,\theta_2) := \frac{\mathcal{L}_X(\theta_1)}{\mathcal{L}_X(\theta_2)} ,$$

where we use the convention that $\mathcal{L}_X(\theta_1)/0 = \infty$, if $\mathcal{L}_X(\theta_1) > 0$, and $0/0 = 1$.

Example 14.6 (likelihood ratio between Gaussians)**.** *The likelihood of a spherical Gaussian $\mathcal{N}_D(\mu,\sigma^2 I_D)$ with respect to observations $X = (x_n)_{n\in[N]}$ is given by*

$$\prod_{n=1}^{N} \mathcal{N}_D(x_n|\mu,\sigma^2) = (2\pi)^{-ND/2}\cdot\sigma^{-ND}\cdot\exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\|x_n-\mu\|_2^2\right) .$$

Let $\theta_1 := (1,\mu_1,\sigma^2 I_D)$ and $\theta_2 := (1,\mu_2,\sigma^2 I_D)$ be two Gaussians with identical covariance matrix $\sigma^2 I_D$. Then,

$$\begin{aligned}
\Lambda_X(\theta_1,\theta_2) &= \frac{\prod_{n=1}^{N}\mathcal{N}_D(x_n|\mu_1,\sigma^2)}{\prod_{n=1}^{N}\mathcal{N}_D(x_n|\mu_2,\sigma^2)} \\
&= \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}\|x_n-\mu_1\|_2^2\right)\cdot\exp\left(+\frac{1}{2\sigma^2}\sum_{n=1}^{N}\|x_n-\mu_2\|_2^2\right) \\
&= \exp\left(\frac{1}{2\sigma^2}\left(\sum_{n=1}^{N}\|x_n-\mu_2\|_2^2-\sum_{n=1}^{N}\|x_n-\mu_1\|_2^2\right)\right) \\
&= \exp\left(\frac{N}{2\sigma^2}\left(\|\mathbf{m}(X)-\mu_2\|_2^2-\|\mathbf{m}(X)-\mu_1\|_2^2\right)\right) . \qquad \text{(Lemma 2.20)}
\end{aligned}$$

14.2.3 Scale Invariance of the Likelihood-Ratio

Suppose we are given observations $X = (x_n)_{n\in[N]}$ and a GMM θ. We scale the observations by some constant factor $c \in \mathbb{R}_+$. That is, we consider $X_c := (c\cdot x_n)_{n\in[N]}$ instead of X. How can we transfer the GMM θ from X to X_c?

Intuitively, we should scale the mean vectors by the same factor c, the covariance matrix by a factor c^2, and leave the weights as they are. This intuition is confirmed by the following result.

Lemma 14.7 (scale)**.** *Let $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, let $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$ be a D-variate GMM, and $c \in \mathbb{R}_+$. Set $\theta_c := \big((w_k, c\cdot\mu_k, c^2\cdot\Sigma_k\big)_{k\in[K]}$ and $X_c := (c\cdot x_n)_{n\in[N]}$. Then,*

$$\mathcal{L}_{c\cdot X}(\theta_c) = \frac{1}{c^{D|X|}}\cdot\mathcal{L}_X(\theta) .$$

Proof. For all $x_n \in X$ and $k \in [K]$, we have

$$\begin{aligned}
\mathcal{N}_D(c \cdot x_n | c\mu_k, c^2\Sigma_k^2) &= \frac{1}{(2\pi)^{D/2}\left|c^2 \cdot \Sigma\right|^{1/2}} \cdot \exp\left(-\frac{1}{2}(c \cdot x_n - c \cdot \mu)^T (c^2 \cdot \Sigma)^{-1}(c \cdot x_n - c \cdot \mu)\right) \\
&= \frac{1}{(2\pi)^{D/2} c^D \cdot |\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2} \cdot c(x_n - \mu)^T \left(c^{-2} \cdot \Sigma^{-1}\right) \cdot c(x_n - \mu)\right) \\
&= \frac{1}{c^D} \cdot \frac{1}{(2\pi)^{D/2} \cdot |\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2}(x_n - \mu)^T \cdot \Sigma^{-1} \cdot (x_n - \mu)\right) \\
&= \frac{1}{c^D} \cdot \mathcal{N}_D(x_n | \mu_k, \Sigma_k^2) \,.
\end{aligned}$$

Hence, $\mathrm{p}(c \cdot x|\theta_c) = \frac{1}{c^D} \cdot \mathrm{p}(x|\theta)$ and $\prod_{n=1}^N \mathrm{p}(c \cdot x_n|\theta_c) = \prod_{n=1}^N \left(\frac{1}{c^D}\,\mathrm{p}(x_n|\theta)\right) = \frac{1}{c^{D \cdot N}} \prod_{n=1}^N \mathrm{p}(x_n|\theta)$. This yields the claim. □

Consequently, the likelihood ratio is invariant against this kind of scaling.

Corollary 14.8 (scale invariance)**.** *Let $X = (x_n)_{n\in[N]} \subseteq \mathbb{R}^D$ and let $\hat{\theta} = \left((\hat{w}_k, \hat{\mu}_k, \hat{\Sigma}_k\right)_{k\in[K]}$ and $\tilde{\theta} = \left((\tilde{w}_k, \tilde{\mu}_k, \tilde{\Sigma}_k\right)_{k\in[K]}$ be D-variate GMMs. Fix some $c \in \mathbb{R}_+$. Set $X_c := (cx_n)_{n\in[N]}$, $\theta_c = \left((\hat{w}_k, c\hat{\mu}_k, c^2\hat{\Sigma}_k\right)_{k\in[K]}$, and $\theta_c = \left((\tilde{w}_k, c\tilde{\mu}_k, c^2\tilde{\Sigma}_k\right)_{k\in[K]}$. Then, $\Lambda_X(\hat{\theta}, \tilde{\theta}) = \Lambda_{X_c}(\hat{\theta}_c, \tilde{\theta}_c)$.*

14.2.4 Maximum Likelihood Estimator for $K \geq 2$

We are given some observations $X = (x_n)_{n\in[N]}$ and presume that these observations have been drawn according to a GMM θ^*. We want to approximate θ^*. There are several approaches to this problem. In this thesis, we focus on the method of maximum likelihood estimation. That is, we want to estimate θ^* by a GMM θ_{ML} which has most likely generated X. We call θ_{ML} a maximum likelihood estimator.

In the following, we assume that we know the number of components K of the underlying GMM θ^*. That is, we focus on determining a mixture θ_{ML} with a predefined number of components $K \in \mathbb{N}$. An obvious formulation of our goal is the following:

Problem 14.9 (a meaningless MLE problem)**.** *We are given $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, and $K \geq 2$. We want to find a GMM $\theta = \left((w_k, \mu_k, \Sigma_k)\right)_{k\in[K]}$ with the maximum likelihood $\mathrm{p}(X|\theta)$.*

It is a well-known fact that Problem 14.9 is no sensible formulation of our problem. For instance, this has already been noted by Day (1969). The reason is simply that for $K \geq 2$ the likelihood function is unbounded from above, which means that there are infinitely many solutions with the likelihood ∞.

Lemma 14.10 (unboundedness for $K \geq 2$)**.** *Let $K \geq 2$, and $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$. Fix arbitrary $\mu_2, \ldots, \mu_K \in \mathbb{R}^D$. Fix arbitrary non-degenerated covariance matrices $\Sigma_2, \ldots, \Sigma_K \in \mathbb{R}^D$.*

Let $\theta(\Sigma) := \left((1/K, x_1, \Sigma), \left(1/K, \mu_2, \Sigma_2\right), \ldots, \left(1/K, \mu_K, \Sigma_K\right)\right)$ for all $p \in \mathbb{N}$.

Then, we have $\mathcal{L}(X|\theta(\Sigma)) \to \infty$ for $|\Sigma| \to 0$ with $\Sigma \neq 0_{D,D}$.

Proof. Observe that

$$\begin{aligned}
\mathcal{L}(X|\theta(\Sigma)) &= \prod_{n=1}^N \left(\frac{1}{K}\mathcal{N}_D(x_n|x_1, \Sigma) + \frac{1}{K}\sum_{k=2}^K \mathcal{N}_D(x_n|\mu_k, \Sigma_k)\right) \\
&= \frac{1}{K^N} \prod_{n=1}^N \left(\mathcal{N}_D(x_n|x_1, \Sigma) + t_{\text{const}}^{(n)}\right) \\
&= \left(\frac{1}{K^N} \prod_{n=2}^N t_{\text{const}}^{(n)}\right) \cdot \mathcal{N}_D(x_1|x_1, \Sigma) + t_{\text{non-neg.}} \,,
\end{aligned}$$

where the $t_{\text{const}}^{(n)}$ are non-negative terms that are *independent* of Σ and $t_{\text{non-neg.}}$ is a non-negative term (that depends on Σ). With Observation 14.3 the claim follows. □

In particular, this lemma holds true for $K = 2$. It does not hold true for $K = 1$, though.

Example 14.11 (boundedness for $K = 1$)**.** *Let $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$. Consider an arbitrary $\sigma^2 > 0$. The likelihood of the Gaussian $\mathcal{N}_D(x_1, \sigma^2 I_D)$ with respect to X computes to*

$$\prod_{n=1}^{N} \mathcal{N}_D(x_n|x_1, \sigma^2 I_D) = \text{const} \cdot \sigma^{-ND} \cdot \exp\left(-\frac{1}{2\sigma^2} \sum_{n\in[N]} \|x_1 - x_n\|_2^2\right)$$

where const *denotes a non-negative term that is independent of σ. Observe that, for $\sigma \to 0$, the term $\exp\left(-\frac{1}{2\sigma^2}\sum_{n\in[N]} \|x_1 - x_n\|_2^2\right)$ converges faster to 0 than σ^{-ND} diverges to ∞.*

14.2.5 Maximum Likelihood Estimator for $K = 1$

It is easy to determine a single Gaussian with maximum likelihood:

Lemma 14.12. *Let $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$ with some $x_n \neq \mathbf{m}(X)$. Then, the vector $\mu \in \mathbb{R}^D$ and the matrix $\Sigma \in \mathbb{R}^{D\times D}$ maximizing $\prod_{n=1}^N \mathcal{N}_D(x_n|\mu, \Sigma)$ satisfy*

$$\mu = \mathbf{m}(X) \quad \text{and} \quad \Sigma = \mathbf{cov}(X) \ .$$

Besides that, the vector $\mu \in \mathbb{R}^D$ and the value $\sigma^2 \in [0, \infty)$ that maximize the likelihood $\prod_{n=1}^N \mathcal{N}_D(x_n|\mu, \sigma^2 I_D)$ satisfy

$$\mu = \mathbf{m}(X) \quad \text{and} \quad \sigma^2 = \frac{\mathbf{var}(X)}{D} \ .$$

Proof. Recall our definitions from Section 2.3 and Section 2.1, where we stated that we identify $X = (x_n)_{n\in[N]}$ with the data set $((x_n, 1))_{n\in[N]}$. A proof of the first claim can be found in (Bishop, 2006, pp. 93), for instance. The second claim follows analogously. Consider the log-likelihood

$$\ln\left(\prod_n \mathcal{N}_D(x_n|\mu, \sigma^2 I_D)\right) = \sum_{n=1}^{N} \ln\left(\mathcal{N}_D(x_n|\mu, \sigma^2 I_D)\right) = \text{const.} - ND\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{n=1}^{N} \left\|x_n - \mu\right\|_2^2 \ .$$

Observe that the optimal choice of the mean vector μ does not depend on the value of σ^2. Due to Lemma 2.20, we know that the optimal choice is $\mu = \mathbf{m}(X)$. For arbitrary but fixed μ, the first derivative in the direction of σ computes to $-\frac{ND}{\sigma} + \frac{1}{\sigma^2}\sum_{n=1}^N \left\|x_n - \mu\right\|_2^2$. Setting this derivative to zero and solving the resulting equation yields the claim. □

Nonetheless, if there are too few different points in X, then the problem degenerates.

Observation 14.13. *Assume that there are at most $D-1$ different points in X. We know that $\Sigma = \frac{\mathbf{ucov}(X)}{N}$ is positive definite if for all $x \in \mathbb{R}^D$, $x \neq 0$, we have $x^T\Sigma x > 0$ (Golub and Loan, 1996, p. 140). Observe that, due to the number of different points, there exists a vector $x \in \mathbb{R}^D$ which is orthogonal to all vectors $(x_1 - \mathbf{m}(X)), \ldots, (x_N - \mathbf{m}(X))$ (i.e., $\forall n : \langle x, (x_n - \mathbf{m}(X))\rangle = 0$). Hence, $x^T\Sigma x = \frac{1}{N}\sum_{n=1}^N x^T(x_n - \mathbf{m}(X))(x_n - \mathbf{m}(X))^T x = \frac{1}{N}\sum_{n=1}^N \langle x, (x_n - \mathbf{m}(X))\rangle^2 = 0$. This shows that we need at least $|X| \geq D$ different observations to obtain a non-degenerated covariance.*

14.2.6 Constrained Maximum Likelihood Estimation

To turn Problem 14.9 into a reasonable problem, one could impose additional constraints on the covariances or add a regularization term to the likelihood function. In the following, we briefly explain the former approach.

Hathaway (1985) proposed the following formalization for one-dimensional mixtures:

Problem 14.14 (constrained one-dimensional MLE problem)**.** *We are given* $N \in \mathbb{N}$*,* $X = (x_n)_{n\in[N]} \subset \mathbb{R}$*,* $K \geq 2$*, and some additional constant* $c \in \mathbb{R}_+$*. We want to find a GMM* $\theta = \left((w_k, \mu_k, \sigma_k)\right)_{k\in[K]}$ *maximizing* $\mathrm{p}(X|\theta)$ *subject to* $\sigma_k^2/\sigma_l^2 \geq c$ *for all* $k, l \in [K]$ *with* $k \neq l$*.*

Hathaway (1985) showed that this problem is well-defined. That is, there exists a global maximizer θ_{ML} to the constrained optimization problem with $\mathcal{L}(\theta_{ML}) < \infty$.

To gain the intuition behind this result, observe the following: The constraint ensures that all σ_i differ by at most a factor c. Hence, if we consider a sequence of mixture models $\left(\left((w_k, \mu_k, \sigma_k^p)\right)_{k\in[K]}\right)_{p\in\mathbb{N}}$, where the variances satisfy the constraint and where $\sigma_k^p \to 0$ for $p \to \infty$ for *some* $k \in [K]$, then we can conclude that $\sigma_l^p \to 0$ for $p \to \infty$ for *all* $l \in [K]$. Then, we basically have the same situation as in Example 14.11. Hence, the resulting likelihood is finite. This is the main idea behind the constraint in Problem 14.14.

García-Escudero et al. (2015) generalized this approach for D-dimensional Gaussian mixtures in a straightforward manner.

Problem 14.15 (constrained D-dimensional MLE problem)**.** *We are given* $N \in \mathbb{N}$*,* $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$*,* $K \geq 2$*, and* $c \in \mathbb{R}_+$*. We want to find a GMM* $\theta = \left((w_k, \mu_k, \Sigma_k)\right)_{k\in[K]}$ *maximizing* $\mathrm{p}(X|\theta)$ *subject to* $\frac{\min\{\lambda_d(\Sigma_k) \mid d\in[D]\}}{\max\{\lambda_d(\Sigma_l) \mid d\in[D]\}} \geq c$ *for all* $k, l \in [K]$ *with* $k \neq l$*, where* $\lambda_d(\Sigma)$ *denotes the* d*-th eigenvalue of* Σ*.*

Unfortunately, to the best of our knowledge, there are no theoretical guidelines on how to choose the parameter c. Hence, we will not work on this problem or Problem 14.14.

14.2.7 Remarks

Unbounded. Problem 14.9 is no sensible problem formulation because the likelihood function is unbounded. We already discussed this issue in Section 14.2.

Spurious Maxima. The constraints imposed by Hathaway and Bezdek (1986) and García-Escudero et al. (2015), which we described in Section 14.2.6, also tackle the problem of spurious maximizers. The issue of spurious maxima is discussed in (McLachlan and Krishnan, 2008, p. 65). He argues, "Consideration has to be given to the problem of a relatively large local maxima that occur as a consequence of a fitted component having a very small (but nonzero) variance for multivariate data. Such a component corresponds to a cluster containing a few data points either relatively close together or almost lying in a lower dimensional subspace in the case of multivariate data. There is thus a need to monitor the relative size of [...] component variances [...] in an attempt to identify these spurious local maximizers". More generally, Day (1969) noted that, "[...] each sample point generates a singularity in the likelihood function. Similarly, any pair of sample points which are sufficiently close together will generate a local maximum, as will triplets, quadruplets and so on which are sufficiently close. Maximum likelihood clearly breaks down".

The constraints imposed by Hathaway and Bezdek (1986) and García-Escudero et al. (2015) do not rule out the possibility of spurious maximizers. In Problem 14.14 and Problem 14.15, a constant c has to be chosen in advance and in accordance to our notion of spurious maxima. Nonetheless, we have reason to believe that the constraints allow for less spurious maxima than there would be without this constraint.

Consistency. Despite the fact that the likelihood is unbounded, the maximum likelihood estimator is consistent (Hathaway and Bezdek, 1986; Kiefer and Wolfowitz, 1956). That is, given a sufficient amount of data that has been drawn according to a Gaussian mixture, there is a local maximizer of the likelihood function that is close to the underlying distribution. More formally, for each $N \in \mathbb{N}$, let X_N be a set of N points drawn according to a one-dimensional Gaussian mixture θ^*. Then, there is a sequence of local maximizers θ_N of the

likelihood function $\mathcal{L}_{X_N}(\cdot)$ such that $\theta_N \to \theta^*$ for $N \to \infty$. For the case $K = 1$, it is easy to see that, for $N \to \infty$, the expected maximum likelihood estimator takes the desired value:

Lemma 14.16. *Let $X = (X_n)_{n\in[N]}$ be a vector of N random variables that have been drawn independently according to the Gaussian $\mathcal{N}_D(\mu, \Sigma)$. Then,*

$$\mathrm{E}[\mathbf{m}(X)] = \mu \quad \textit{and} \quad \mathrm{E}[\mathbf{cov}(X)] = \frac{N-1}{N}\Sigma\,.$$

Proof. Observe that each X_n is distributed identically with

$$\mathrm{E}[X_n] = \mu \qquad \text{and } \mathrm{E}\left[(X_n-\mu)(X_n-\mu)^T\right] = \Sigma \tag{14.3}$$

for each $n \in [N]$ (Bishop, 2006, p. 83). Due to linearity of expectation, $\mathrm{E}[\mathbf{m}(X)] = \mathrm{E}[X_n]$ for all $n \in [N]$. With (14.3), we can conclude that the first claim holds true. To prove the second claim, observe that

$$\begin{aligned}
\mathrm{E}[\mathbf{cov}(X)] &= \mathrm{E}\left[\frac{\mathbf{ucov}(X)}{N}\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}\left[(X_n-\mathbf{m}(X))(X_n-\mathbf{m}(X))^T\right] && \text{(linearity)} \\
&= \mathrm{E}\left[(X_n-\mathbf{m}(X))(X_n-\mathbf{m}(X))^T\right] && \text{(identical distribution)} \\
&= \mathrm{E}\left[(X_n-\mu+\mu-\mathbf{m}(X))(X_n-\mu+\mu-\mathbf{m}(X))^T\right] \\
&= \mathrm{E}\left[(X_n-\mu)(X_n-\mu)^T + (X_n-\mu)(\mu-\mathbf{m}(X))^T\right. \\
&\quad \left. + (\mu-\mathbf{m}(X))(X_n-\mu)^T + (\mu-\mathbf{m}(X))(\mu-\mathbf{m}(X))^T\right] \\
&= \mathrm{E}\left[(X_n-\mu)(X_n-\mu)^T\right] + \mathrm{E}\left[(X_n-\mu)(\mu-\mathbf{m}(X))^T\right] \\
&\quad + \mathrm{E}\left[(\mu-\mathbf{m}(X))(X_n-\mu)^T\right] + \mathrm{E}\left[(\mu-\mathbf{m}(X))(\mu-\mathbf{m}(X))^T\right]\,. && \text{(linearity)}
\end{aligned}$$

Let us consider the single summands separately. Observe that

$$\begin{aligned}
&\mathrm{E}\left[(X_n-\mu)(\mu-\mathbf{m}(X))^T\right] \\
&= -\mathrm{E}\left[(X_n-\mu)(\mathbf{m}(X)-\mu)^T\right] \\
&= -\mathrm{E}\left[(X_n-\mu)\left(\sum_{m=1}^{N}\frac{X_m-\mu}{N}\right)^T\right] && \text{(Definition 2.14)} \\
&= -\frac{1}{N}\sum_{m\in[N]}\mathrm{E}\left[(X_n-\mu)\left(X_m-\mu\right)^T\right] && \text{(linearity)} \\
&= -\frac{1}{N}\mathrm{E}\left[(X_n-\mu)\left(X_n-\mu\right)^T\right] - \frac{1}{N}\sum_{m\in[N]\setminus\{n\}}\mathrm{E}\left[X_n-\mu\right]\mathrm{E}\left[X_m-\mu\right]^T && \text{(independence)} \\
&= -\frac{1}{N}\mathrm{E}\left[(X_n-\mu)\left(X_n-\mu\right)^T\right] - \frac{1}{N}\sum_{m\in[N]\setminus\{n\}}(\mathrm{E}[X_n]-\mu)(\mathrm{E}[X_m]-\mu)^T && \text{(linarity)} \\
&= -\frac{1}{N}\Sigma - \frac{N-1}{N}0_{D,D} && \text{(Equation (14.3))} \\
&= -\frac{1}{N}\Sigma\,. && (14.4)
\end{aligned}$$

So we also have $\mathrm{E}\left[(\mu-\mathbf{m}(X))(X_n-\mu)^T\right] = \mathrm{E}\left[(X_n-\mu)(\mu-\mathbf{m}(X))^T\right]^T = -\frac{1}{N}\Sigma$. Moreover,

$$\mathrm{E}\left[(\mu-\mathbf{m}(X))(\mu-\mathbf{m}(X))^T\right] = \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}\left[(\mu-X_n)(\mu-\mathbf{m}(X))^T\right] \qquad \text{(Definition 2.14)}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} \mathrm{E}\left[(X_n - \mu)(\mu - \mathbf{m}(X))^T\right] \qquad \text{(linearity)}$$

$$= \frac{1}{N}\Sigma\ . \qquad \text{(Equation (14.4))}$$

Hence, $\mathrm{E}\left[\frac{\mathbf{ucov}(X)}{N}\right] = (1 - 2\frac{1}{N} + \frac{1}{N})\Sigma = \frac{N-1}{N}\Sigma$, which yields the claim. □

Curse of Dimensionality (Parameters). Last but not least, one should be aware of the curse of dimensionality with regard to the space of the parameters. We consider multivariate Gaussian mixtures with a fixed number of components K where each component has its own set of parameters: a weight, a D-dimensional mean vector, and a symmetric $(D \times D)$-matrix as covariance. This means that we actually have

$$K \cdot \left(1 + D + \frac{D \cdot (D+1)}{2}\right)$$

real-valued parameters in total. This number of parameters is *quadratic* in the dimension. We need a large number of observations to correctly estimate all these parameters. There are several approaches that tackle this problem, such as dimension reduction, regularization of parameters, constrained and parsimonious clustering. For an overview, we refer to Bouveyron and Brunet-Saumard (2014).

14.3 Expectation-Maximization (EM)

We are given a set of observations $X = (x_n)_{n\in[N]}$ and assume that these observations have been drawn according to a GMM θ^*. We want to estimate θ^* via the GMM θ_{ML} that has most likely generated X.

The standard approach to this problem is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Bishop, 2006; McLachlan and Krishnan, 2008), which is actually a very general framework.

14.3.1 General Framework

The following problem is a straightforward generalization of Problem 14.9 where we replace the family of GMMs by some arbitrary family of parameterized models.

Problem 14.17 (general MLE problem)**.** *We are given a parameterized statistical model* $(\mathcal{X}, \{\mathrm{p}(\cdot|\theta) \mid \theta \in \Theta\})$ *and observations* $X \in \mathcal{X}$*. Find*

$$\theta_{ML} \in \arg\max\{\mathrm{p}(X|\theta) \mid \theta \in \Theta\}\ .$$

The EM algorithm that is described in Algorithm 16 is a heuristic for this problem. It leverages the notion of hidden random variables Z. The functioning of this algorithm is explained by the following fundamental result. For the sake of simplicity, let us assume that the hidden variables are discrete.

Lemma 14.18 (decomposition of the likelihood)**.** *For all parameters* $\theta \in \Theta$ *and all distributions* q *over the hidden variables* Z*, it holds*

$$\ln(\mathrm{p}(X|\theta)) = \mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))] + \mathcal{H}(q) + \mathrm{KLD}(\mathrm{p}(\cdot|X,\theta)\|q)\ ,$$

where

$$\mathcal{H}(q) := -\sum_{Z} q(Z)\ln(q(Z)) \geq 0$$

Algorithm 16 EM Update Step

Require: observations X, parameters $\theta^{old} \in \Theta$

Expectation Step: Determine a distribution q over the latent variables Z:

$$q(Z) := \mathrm{p}(Z|X,\theta^{old})$$

Maximization Step: Determine the parameter $\theta \in \Theta$ that maximizes

$$\mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))] \ .$$

return θ

Algorithm 17 EM Update Step for GMMs

Require: observation $X = (x_n)_{n\in[N]} \subseteq \mathbb{R}^D$, parameters $\{(w_k^{old}, \mu_k^{old}, \Sigma_k^{old})\}_{k\in[K]}$

for all $n \in [N]$ and $k \in [K]$ **do**
Determine the posterior probabilities

$$p_{nk} := \frac{w_k^{old}\mathcal{N}_D(x_n|\mu_k^{old},\Sigma_k^{old})}{\sum_{l=1}^K w_l^{old}\mathcal{N}_D(x_n|\mu_l^{old},\Sigma_l^{old})} \ .$$

for all $k \in [K]$ **do**

$$w_k := \frac{1}{N}\sum_{n=1}^N p_{nk}$$

$$\mu_k := \frac{\sum_{n=1}^N p_{nk}x_n}{\sum_{n=1}^N p_{nk}}$$

$$\Sigma_k := \frac{\sum_{n=1}^N p_{nk}(x_n-\mu_k)(x_n-\mu_k)^T}{\sum_{n=1}^N p_{nk}}$$

return $\big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$

and

$$\mathrm{KLD}(\mathrm{p}(\cdot|X,\theta)\|q) := \sum_Z \mathrm{p}(Z|X,\theta)\ln\left(\frac{\mathrm{p}(Z|X,\theta)}{q(Z)}\right) \geq 0$$

with the convention that $\ln(0) = 0$, $0\cdot\ln(0/p) = 0$, *and* $p\cdot\ln(p/0) = \infty$ *for all* $p \neq 0$. *Besides that,* $\mathrm{KLD}(\mathrm{p}(\cdot|X,\theta)\|q) = 0$ *if and only if* $\mathrm{p}(\cdot|X,\theta) = q$.

Proof. A proof can be found in (Bishop, 2006, pp. 450), for instance. □

With this fundamental result, it is easy to see that an EM update step computes parameters θ that are not less likely than the given parameters θ^{old}.

Observation 14.19 (monotonicity)**.** *Consider a run of Algorithm 16, given some observations* $X \in \mathcal{X}$ *and parameters* $\theta^{old} \in \Theta$. *Due to Lemma 14.18, we know that*

$$\begin{aligned}
&\ln(\mathrm{p}(X|\theta^{old})) \\
&= \mathrm{E}_{Z\sim q(Z)}\left[\ln\left(\mathrm{p}(X,Z|\theta^{old})\right)\right] + \mathcal{H}(q) + \mathrm{KLD}\left(\mathrm{p}(\cdot|X,\theta^{old})\middle\|q\right) && \text{(arbitrary distribution } q\text{)} \\
&= \mathrm{E}_{Z\sim \mathrm{p}(Z|X,\theta^{old})}\left[\ln\left(\mathrm{p}(X,Z|\theta^{old})\right)\right] + \mathcal{H}(\mathrm{p}(\cdot|X,\theta^{old})) + 0 && (q(Z) = \mathrm{p}(Z|X,\theta^{old})) \\
&\leq \mathrm{E}_{Z\sim \mathrm{p}(Z|X,\theta^{old})}[\ln(\mathrm{p}(X,Z|\theta))] + \mathcal{H}(\mathrm{p}(\cdot|X,\theta^{old})) + 0 \\
&\leq \mathrm{E}_{Z\sim \mathrm{p}(Z|X,\theta^{old})}[\ln(\mathrm{p}(X,Z|\theta))] + \mathcal{H}(\mathrm{p}(\cdot|X,\theta^{old})) + \mathrm{KLD}\left(\mathrm{p}(\cdot|X,\theta)\middle\|\mathrm{p}(\cdot|X,\theta^{old})\right) && (\mathrm{KLD} \geq 0) \\
&= \ln(\mathrm{p}(X|\theta)) \ .
\end{aligned}$$

Besides the monotonicity of the sequence of likelihood values produced by the EM algorithm, little is guaranteed. For instance, (McLachlan and Krishnan, 2008, pp. 79) noted the following: The sequence of likelihoods $(\mathcal{L}_X(\theta_r))_r$ produced by the EM algorithm might diverge to ∞ if the likelihood is not bounded from above (cf. Section 14.2.4). If the likelihood is bounded, then the sequence $(\mathcal{L}_X(\theta_r))_r$ converges monotonically to some $\mathcal{L}_X(\theta) < \infty$. However, the sequence of models $(\theta_r)_r$ computed by the EM algorithm is not guaranteed to converge, even if the sequence of likelihoods $(\mathcal{L}_X(\theta_r))_r$ converges. If the sequence of likelihoods $(\mathcal{L}_X(\theta_r))_r$ computed by the EM algorithm converges to some $\mathcal{L}_X(\theta) < \infty$, then $\mathcal{L}_X(\theta)$ is *not* necessarily a local maximum of the likelihood. Nonetheless, (Wu, 1983) shows that under some regularity conditions the EM algorithm converges to a local maximum.

14.3.2 EM Algorithm for GMMs

The EM algorithm for GMMs takes the form described in Algorithm 17. As already explained in the last section, the EM algorithm never decreases the likelihood. However, it is not guaranteed that a sequence produced by Algorithm 17 converges to a non-degenerated GMM. Besides that, the EM algorithm has two major drawbacks: On the one hand, the convergence of the EM algorithm can be very slow. In particular if the mixture components are "not well separated", as observed by Xu and Jordan (1996). On the other hand, the EM algorithm is prone to get trapped in (poor) stationary points of the likelihood function (McLachlan and Krishnan, 2008, p. 228). In Chapter 15, we analyse a stochastic version of the EM algorithm which is known to suffer less from these drawbacks. In Chapter 16, we derive initialization methods for the EM algorithm for GMMs which, hopefully, prevent the algorithm from getting trapped in a poor solution.

14.4 Overview

The following three chapters deal with three different topics:

Chapter 15 provides a theoretical comparison of the classical EM algorithm and a stochastic variant thereof, with respect to certain mixture models. Making use of our results from Chapter 3, we show that under certain conditions the update formulas of both algorithms yield similar results with high probability.

Chapter 16 deals with the problem of initializing the EM algorithm for GMMs. We propose new initialization methods that are based on the well-known K-means++ algorithm by Arthur and Vassilvitskii (2007) for the K-means problem and the algorithm of Gonzalez (1985) for the so-called discrete radius clustering problem. We compare these new methods with a large number of existing methods via experiments with respect to artificial data sets and real-world data sets.

Chapter 17 deals with an approach towards a theoretical analysis of the maximum likelihood estimation (MLE) problem. We consider a special case of the MLE problem where the weights and covariances of the Gaussian mixture models (GMMs) are fixed in advance and so the only degrees of freedom that are left to be determined are mean vectors. We propose and discuss different variants of this problem and an approach towards a theoretical analysis.

"Mathematics, according to D. Hilbert (1862-1943), is nothing more than a game played according to certain simple rules with meaningless marks on paper."

E.T. Bell[1]

Chapter 15

A Non-Asymptotic Comparison of EM and SEM Algorithms

A major downside of the EM algorithm for mixture models is that there are cases where the algorithm converges very slowly and gets attracted to unstable stationary points of the likelihood function. A variant of the EM algorithm that tackles this problem is the stochastic EM (SEM) algorithm, which is also known as a special case of the Monte Carlo EM (MCEM) algorithm. It replaces the expectation step of the EM algorithm by a stochastic expectation step: Recall that the goal of the expectation step is to compute the distribution q which determines the expected complete-data log-likelihood $\mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))]$. The stochastic EM algorithm approximates this expected value via a sample $\ln(\mathrm{p}(X,Z=z|\theta))$ where z is drawn according to $q(Z=z)$. Intuitively, the SEM algorithm imitates the EM algorithm if the data set is sufficiently large. In this chapter, we aim to quantify this intuition with respect to EM and SEM algorithms for mixture models.

Overview. In Section 15.1, we introduce the SEM algorithm formally. In Section 15.2, we give an overview of the mixture models with respect to which we compare EM and SEM (like) algorithms. In Section 15.3 and Section 15.4, we summarize related work and our contribution. Section 15.5 contains our theoretical comparison. In Section 15.6, we provide some examples and illustrate our bounds.

Publication. We generalize and discuss the result from Blömer et al. (2014).

15.1 Introduction

Recall that an update step of the EM algorithm, which is described in Algorithm 18, consists of two steps: In the expectation step, the algorithm computes a distribution q over the hidden variables Z. Together with the observations X, this distribution q determines the expected complete-data log-likelihood $f(\theta) := \mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))]$. Then, in the maximization step, the expected complete-data log-likelihood $f(\theta)$ is maximized with respect to θ.

The stochastic EM (SEM) algorithm is described in Algorithm 19. Instead of determining the expected complete-data log-likelihood $f(\theta)$ exactly, the stochastic EM (SEM) algorithm approximates $f(\theta)$ via sampling. More precisely, it approximates the expectation $f(\theta) = \mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))]$ via a sample $f_z(\theta) = \ln(\mathrm{p}(X,Z=z|\theta))$ where z is drawn according to $q(Z=z)$. That is, the expectation step of the EM is replaced by a stochastic expectation step where the SEM algorithm samples realizations of the hidden variables. Then, in the maximization step, the SEM algorithm proceeds in the same way as the EM algorithm, but

[1] Source: E.T. Bell, Mathematics (Queen and Servant of Science). G. Bell & Sons Ltd. 1952 (p. 21)

Algorithm 18 EM Update Step

Require: X, $\theta^{old} \in \Theta$

Expectation Step: Determine a distribution q over the latent variables Z:

$$q(Z) := \mathrm{p}(Z|X,\theta^{old})$$

Maximization Step: Determine the parameter $\theta \in \Theta$ that maximizes

$$\mathrm{E}_{Z\sim q(Z)}[\ln(\mathrm{p}(X,Z|\theta))] \ .$$

return θ

Algorithm 19 SEM Update Step

Require: X, $\theta^{old} \in \Theta$

Stochastic Expectation Step: Sample a realization z according to

$$z \sim \mathrm{p}(Z = z|X,\theta^{old})$$

Maximization Step: Determine the parameter $\theta \in \Theta$ that maximizes

$$\ln(\mathrm{p}(X, Z = z|\theta)) \ .$$

return θ

with $f(\theta)$ replaced by $f_z(\theta)$. That is, it determines the parameters θ maximizing $f_z(\theta)$. Note that it is possible to improve the estimate of the expectation $f(\theta)$ via repeated sampling. This generalization of the SEM algorithm is known as the Monte Carlo EM (MCEM) algorithm (McLachlan and Krishnan, 2008, p. 227).

A downside of the SEM algorithm is that we have to give up the monotonicity property of the EM algorithm (cf. Observation 14.19). That is, one round of the SEM algorithm might produce a model with smaller likelihood than the given model. However, a major advantage of the SEM algorithm is that it prevents us from "staying near an unstable stationary point of the likelihood function" (McLachlan and Krishnan, 2008, p. 228). Thereby it avoids cases where the EM algorithm (for mixture models) is known to converge very slowly. Last but not least, note that sometimes the EM algorithm is simply no alternative to the SEM algorithm because the expectation step of the EM algorithm is analytically intractable for some probabilistic models (McLachlan and Krishnan, 2008, p. 224). For more information, we refer to Celeux and Diebolt (1985), Bishop (2006), and McLachlan and Krishnan (2008).

15.2 Scope of Our Comparison

In the following, we focus on mixture models that are parameterized by a mean vector $\mu \in \mathbb{R}^D$ and, possibly, a covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$.

Definition 15.1. *We consider a family of parameterized D-variate density functions* $\mathcal{P} = \{\mathrm{p}(\cdot|\mu,\Sigma) \mid \mu \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D\times D}\}$. *The probability density function* $\mathrm{p}(\cdot|\theta) : \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ *of a mixture model with K components from $\mathcal{P}$ is given by* $\mathrm{p}(x|\theta) = \sum_{k=1}^{K} w_k \, \mathrm{p}(x|\mu_k,\Sigma_k)$, *where* $\theta = \big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$, $(w_k)_{k\in[K]} \in \Delta_{K-1}$, $\mu_k \in \mathbb{R}^D$ *and* $\Sigma_k \in \mathbb{R}^{D\times D}$ *for all* $k \in [K]$

For these mixtures, we analyse EM-like and SEM-like algorithms that take the form described in Algorithm 20 and Algorithm 21, respectively. This covers the following distributions and algorithms:

First and foremost, we consider the EM and SEM algorithm for Gaussian mixture models: A Gaussian distribution is parameterized by a mean vector $\mu \in \mathbb{R}^D$ and a covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$. As already explained in Section 14.1, its probability density function takes the form

$$\mathcal{N}_D(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}\,|\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2}(x_n-\mu)^T\Sigma^{-1}(x_n-\mu)\right) \ .$$

For mixtures of Gaussian distributions, the component update of the corresponding EM and SEM algorithm take the form depicted in Algorithm 20 and Algorithm 21, respectively, with all ζ_{nk} values set to 1 (see Bishop (2006) and Algorithm 17).

Algorithm 20 EM* Update Step

Require: data set $X = (x_n)_{n\in[N]}$,
$\theta^{old} = \left((w_k^{old}, \mu_k^{old}, \Sigma_k^{old})\right)_{k\in[K]}$,
a density function $\mathrm{p}(\cdot|\theta^{old})$,
some function $\zeta_{(X,\theta^{old})} : [N] \times [K] \to \mathbb{R}_+$

for all $n \in [N]$ and $k \in [K]$ **do**
$p_{nk} := \frac{w_k^{old}\,\mathrm{p}(x_n|\mu_k^{old},\Sigma_k^{old})}{\sum_{l=1}^K w_l^{old}\,\mathrm{p}(x_n|\mu_l^{old},\Sigma_l^{old})}$
$\zeta_{nk} := \zeta_{(X,\theta^{old})}(n,k)$

for all $k \in [K]$ **do**
$w_k := \frac{1}{N}\sum_{n=1}^N p_{nk}$
$\mu_k := \frac{\sum_{n=1}^N p_{nk}\zeta_{nk}x_n}{\sum_{n=1}^N p_{nk}\zeta_{nk}}$
$\Sigma_k := \frac{\sum_{n=1}^N p_{nk}\zeta_{nk}y_{nk}}{\sum_{n=1}^N p_{nk}}$
where $y_{nk} = (x_n - \mu_k)(x_n - \mu_k)^T$

return $\left((w_k, \mu_k, \Sigma_k)\right)_{k\in[K]}$

Algorithm 21 SEM* Update Step

Require: data set $X = (x_n)_{n\in[N]}$,
$\theta^{old} = \left((w_k^{old}, \mu_k^{old}, \Sigma_k^{old})\right)_{k\in[K]}$,
a density function $\mathrm{p}(\cdot|\theta^{old})$,
some function $\zeta_{(X,\theta^{old})} : [N] \times [K] \to \mathbb{R}_+$

for all $n \in [N]$ and $k \in [K]$ **do**
$p_{nk} := \frac{w_k^{old}\,\mathrm{p}(x_n|\mu_k^{old},\Sigma_k^{old})}{\sum_{l=1}^K w_l^{old}\,\mathrm{p}(x_n|\mu_l^{old},\Sigma_l^{old})}$
$\zeta_{nk} := \zeta_{(X,\theta^{old})}(n,k)$

for all $n \in [N]$ **do**
Sample $(Z_{nk})_k \in \{0,1\}^K$ with
$\sum_k Z_{nk} = 1$ s.t. $\Pr(Z_{nk} = 1) = p_{nk}$.

for all $k \in [K]$ **do**
$W_k := \frac{1}{N}\sum_{n=1}^N Z_{nk}$
$M_k := \frac{\sum_{n=1}^N Z_{nk}\zeta_{nk}x_n}{\sum_{n=1}^N Z_{nk}\zeta_{nk}}$
$S_k := \frac{\sum_{n=1}^N Z_{nk}\zeta_{nk}Y_{nk}}{\sum_{n=1}^N Z_{nk}}$
where $Y_{nk} = (x_n - M_k)(x_n - M_k)^T$

return $((W_k, M_k, S_k))_{k\in[K]}$

Second, our work covers heuristics for mixtures of multivariate power exponential (MPE) distributions, which are generalizations of multivariate Gaussian distributions (Gómez et al., 1998). MPE distributions have an additional shape parameter $s \in \mathbb{R}_+$. For $s = 1$ one obtains Gaussians, while $s = 1/2$ yields Laplacians. The probability density function of an MPE distribution with shape $s \in \mathbb{R}_+$, mean vector $\mu \in \mathbb{R}^D$, and covariance $\Sigma \in \mathbb{R}^{D\times D}$ is given by

$$\mathcal{M}_s(x|\mu,\Sigma) = \mathcal{C}\cdot|\Sigma|^{-\frac{1}{2}}\exp\left(-\tfrac{1}{2}\left((x-\mu)^T\Sigma^{-1}(x-\mu)\right)^s\right),$$

where $\mathcal{C} = D\Gamma\left(\frac{D}{2}\right)/\left(\pi^{\frac{D}{2}}\Gamma\left(1+\frac{D}{2s}\right)2^{1+\frac{D}{2s}}\right)$ and where $\Gamma(\cdot)$ denotes the gamma function. In the following, we assume that the shape parameter s is some given fixed constant. For mixtures of MPE distributions, there is no implementation of the EM or SEM algorithm known since there is no known implementation of the maximization step of either algorithm. However, Zhang and Liang (2010) introduced an EM-like heuristic which was later corrected by Dang et al. (2015). The EM-like heuristic corresponds to Algorithm 20 with the ζ_{nk} values set to

$$\zeta_{nk} := s\left((x_n - \mu_k^{old})^T(\Sigma_k^{old})^{-1}(x_n - \mu_k^{old})\right)^{s-1}.$$

Dang et al. (2015) showed that the covariance update of this heuristic is reasonable if $s \in (0,1]$. More precisely, he showed that, given fixed weights and means, an update of the covariance matrices does not decrease the expected complete-data log-likelihood which shall be maximized in the last step of the EM algorithm (cf. Dang et al. (2015), Bishop (2006)).

Third, our analysis also covers the EM and SEM Algorithm for mixtures of regular exponential distributions. These distributions are only parameterized by a mean vector $\mu \in \mathbb{R}^D$. Their density functions are of the form

$$\mathcal{R}_\psi(x|\mu) = \exp\left(\langle x,\mu\rangle - \psi(\mu)\right)\mathcal{R}_0(x),$$

where $\psi(\cdot)$ and $\mathcal{R}_0(\cdot)$ denote given functions. For instance, Banerjee et al. (2005) showed that the component update of the corresponding EM algorithm takes the form depicted in Algorithm 20 (excluding the update of the (non-existing) covariance) and with all the ζ_{nk} values set to 1. For more information, we refer to (McLachlan and Krishnan, 2008, pp. 22).

15.3 Related Work

There are various theoretical comparisons of the EM and SEM algorithm that deal with asymptotic theory, for instance, Celeux et al. (1995), Nielsen (2000a) and Nielsen (2000b). For mixtures of distributions from the exponential family, Ip (1994) shows that the sequence of models generated by iterations of the SEM update step is an ergodic Markov chain that converges weakly to a stationary distribution over models. Furthermore, he shows that, under appropriate assumptions, the mean of this stationary distribution converges to the maximum likelihood estimator.

An experimental comparison between the EM and SEM algorithm for mixtures of Gaussian distributions can be found in Dias and Wedel (2004), for instance. The algorithms are applied to two small one-dimensional data sets (containing 150 and 174 points) and use Gaussian mixtures with 3 and 2 components, respectively. The authors evaluate the log-likelihood of the sequence of produced solutions and the log-likelihood surface in the neighborhood of these solutions. Furthermore, they compare the final Gaussian mixture models returned by the algorithms. Their results indicate that the SEM algorithm converges faster and more reliably than the EM algorithm. More comparisons can be found in Celeux et al. (1996), for instance.

15.4 Contribution

We analyse an SEM* update step in comparison to an EM* update step, with respect to different mixture models. We show that these single update steps, with high probability, yield similar results if the given data set X and the given model θ^{old} satisfy certain properties.

15.5 Theoretical Comparison

In this section, we state probabilistic bounds on the differences between an EM* and SEM* update step. First, we state and prove these bounds in the most general form. Then we consider the specific case of Gaussian mixture models. Finally, we discuss our results.

15.5.1 A Non-Asymptotic Bound

With our results from Chapter 3 we can derive the following probabilistic bound on the proximity between Algorithm 20 and Algorithm 21.

Theorem 15.2 (Proximity of Update Steps)**.** *Consider a single run of Algorithm 20 and Algorithm 21, given the same observations $X = (x_n)_{n\in[N]} \subseteq \mathbb{R}^D$, mixture model θ^{old} with K components, density function $\mathrm{p}(\cdot|\theta^{old})$, and function $\zeta_{(X,\theta^{old})}$, respectively.*

Let $\delta \in (0,1)$,

$$a_\delta := 3\ln(2/\delta) \quad \text{and} \quad b_\delta := \sqrt{2e\ln(2/\delta)}$$

For all $k \in [K]$, let

$$\zeta_k^{max} = \max\{\zeta_{nk} \mid n \in [N]\} \ .$$

If for all $k \in [K]$ we have

$$r_k := \sum_{n=1}^{N} p_{nk} \geq a_\delta \quad \text{and} \quad u_k := \sum_{n=1}^{N} p_{nk}\zeta_{nk} \geq a_\delta \zeta_k^{max} \ , \tag{15.1}$$

then, with probability $1-K\cdot\left(2+D+\frac{D(D+1)}{2}\right)\cdot\delta$, *for all* $k\in[K]$ *and* $d,i,j\in[D]$ *we have*

$$|W_k-w_k|\leq\frac{\sqrt{a_\delta}}{\sqrt{r_k}}\cdot w_k\ , \tag{15.2}$$

$$\left|\left(M_k-\mu_k\right)_d\right|\leq\frac{\lambda^{(\mu)}_{(kd)}}{\sqrt{u_k}-\sqrt{a_\delta\zeta_k^{max}}}\cdot\frac{\tau_{kd}}{\sqrt{u_k}}\ ,\ and \tag{15.3}$$

$$\left|(S_k-\Sigma_k)_{ij}\right|\leq\frac{\lambda^{(\Sigma)}_{(kij)}}{\sqrt{r_k}-\sqrt{a_\delta}}\cdot\frac{\rho_{kij}}{\sqrt{r_k}}+\frac{\left(\sqrt{u_k}+\sqrt{a_\delta\zeta_k^{max}}\right)}{\left(\sqrt{u_k}-\sqrt{a_\delta\zeta_k^{max}}\right)^2}\cdot\frac{\lambda^{(\mu)}_{(ki)}\lambda^{(\mu)}_{(kj)}}{\sqrt{r_k}-a_\delta}\cdot\frac{\tau_{ki}\tau_{kj}}{\sqrt{u_k r_k}}\ , \tag{15.4}$$

where

$$\begin{aligned}
\tau^2_{kd}&=\sum_{n=1}^N p_{nk}(1-p_{nk})\zeta^2_{nk}(x_n-\mu_k)^2_d\ ,\\
\lambda^{(\mu)}_{(kd)}&=\begin{cases}b_\delta & if\ \tau_{kd}\geq\frac{1}{e}b_\delta\cdot\zeta_k^{max}\mathrm{r}_d(X)\\ \frac{2b_\delta^2}{e}\cdot\frac{\zeta_k^{max}\mathrm{r}_d(X)}{\tau_{kd}} & otherwise\end{cases}\ ,\\
\rho^2_{kij}&=\sum_{n=1}^N p_{nk}(1-p_{nk})(\zeta_{nk}y_{nk}-\Sigma_k)^2_{ij}\quad with\ \ y_{nk}=(x_n-\mu_k)(x_n-\mu_k)^T\ ,\ and\\
\lambda^{(\Sigma)}_{(kij)}&=\begin{cases}b_\delta & if\ \rho_{kij}\geq\frac{1}{e}b_\delta\cdot\zeta_k^{max}\mathrm{r}_i(X)\mathrm{r}_j(X)\\ \frac{2b_\delta^2}{e}\cdot\frac{\zeta_k^{max}\mathrm{r}_i(X)\mathrm{r}_j(X)}{\rho_{kij}} & otherwise\end{cases}\ .
\end{aligned} \tag{15.5}$$

In the remainder of this section, we prove this theorem. For a discussion, we refer to the next section.

Because the covariance update does not take the same form as a covariance of a soft cluster, we cannot make use of Lemma 3.19. However, we can derive the following similar result:

Lemma 15.3. *Consider the setting from Theorem 15.2. Let* $\delta\in(0,1)$, $k\in[K]$, *and* $i,j\in[D]$. *We have*

$$\Pr\left(\left|\sum_{n=1}^N Z_{nk}\left(\zeta_{nk}y_{nk}-\Sigma_k\right)_{ij}\right|>\lambda_{kij}\cdot\rho_{kij}\right)\leq\delta\ . \tag{15.6}$$

Proof. The following proof is similar to the proof of Lemma 3.19 (the main differences are marked in boldface). For each $n\in[N]$, define the real random variable

$$S_{kijn}:=(Z_{nk}-p_{nk})(\zeta_{\mathbf{nk}}y_{nk}-\Sigma_k)_{ij}\ .$$

Since the Z_{nk} are binary random variables and since each membership p_{nk} lies in $[0,1]$, we have $|Z_{nk}-p_{nk}|\leq 1$. Since $\left(\mu_k\right)_d$ is a convex combination of the coordinates $(x_m)_d$ with $m\in[N]$, we know that $(x_n-\mu_k)_d\in[-\mathrm{r}_d(X),+\mathrm{r}_d(X)]$ for all $n\in[N]$. Hence, for all $n\in[N]$ and $i,j\in[D]$, we can conclude that $(y_{nk})_{ij}=(x_n-\mu_k)_i(x_n-\mu_k)_j\in\left[-\mathrm{r}_i(X)\cdot\mathrm{r}_j(X),\ +\mathrm{r}_i(X)\cdot\mathrm{r}_j(X)\right]$. As $(\mathbf{cov}_k)_{ij}$ is a convex combination of values in $(\zeta_{\mathbf{mk}}y_{mk})_{ij}$ with $m\in[N]$, it follows that

$$(\zeta_{\mathbf{nk}}y_{nk}-\mathbf{cov}_k)_{ij}\in\left[-2\cdot\zeta_{\mathbf{k}}^{\mathbf{max}}\mathrm{r}_i(X)\mathrm{r}_j(X),\ +2\cdot\zeta_{\mathbf{k}}^{\mathbf{max}}\mathrm{r}_i(X)\mathrm{r}_j(X)\right]$$

for all $n\in[N]$. Hence, $\left|(\zeta_{\mathbf{nk}}y_{nk}-\mathbf{cov}_k)_{ij}\right|\leq 2\zeta_{\mathbf{k}}^{\mathbf{max}}\mathrm{r}_i(X)\mathrm{r}_j(X)$ for all $n\in[N]$. Putting these inequalities together yields

$$\left|S_{kijn}\right|=|Z_{nk}-p_{nk}|\cdot\left|(\zeta_{\mathbf{nk}}y_{nk}-\mathbf{cov}_k)_{ij}\right|\leq 2\cdot\zeta_{\mathbf{k}}^{\mathbf{max}}\mathrm{r}_i(X)\mathrm{r}_j(X)\ .$$

With Lemma 3.12, we conclude that

$$\mathrm{E}[S_{kijn}] = 0 \quad \text{and} \quad \mathrm{Var}(S_{kijn}) = p_{nk}(1-p_{nk})(\zeta_{\mathbf{nk}} y_{nk} - \mathbf{covs}_k)_{ij}^2 \ .$$

Similarly to Observation 2.19, we can identify

$$S_{kij} \coloneqq \sum_{n=1}^{N} S_{kijn} = \sum_{n=1}^{N} Z_{nk}(\zeta_{\mathbf{nk}} y_{nk} - \mathbf{covs}_k)_i (\zeta_{\mathbf{nk}} y_{nk} - \mathbf{covs}_k)_j \ .$$

With our previous results, we get $\mathrm{E}[S_{kij}] = 0$ and $\mathrm{Var}(S_{kij}) = \rho_{kij}^2$.

Finally, applying Theorem 3.10 with $C \coloneqq 2\zeta_{\mathbf{k}}^{\mathbf{max}} \mathtt{r}_i(X)\mathtt{r}_j(X)$ yields the claim. □

Proof of Theorem 15.2. As they are given the same parameters, both algorithms compute the same soft clustering $P = (p_{nk})_{n\in[N],k\in[K]}$ and values $(\zeta_{nk})_{n\in[N],k\in[K]}$. Additionally, Algorithm 21 samples a hard assignment $(z_{nk})_k$ (according to $\Pr(z_{nk} = 1) = p_{nk}$), for each observation x_n. Let $Z \coloneqq (z_{nk})_{n\in[N],k\in[K]}$. Observe that $r_k = \mathbf{w}(A_k^{(X,P)})$, $w_k = \mathbf{w}(A_k^{(X,P)})/|X|$, and $W_k = \mathbf{w}(A_k^{(X,Z)})/|X|$. For each $k \in [K]$, define

$$W_k \coloneqq ((x_n, \zeta_{nk}))_{n\in[K]} \ .$$

Then we can identify $\mu_k = \mathbf{m}(A_k^{(W_k,P)})$, $M_k = \mathbf{m}(A_k^{(W_k,Z)})$, and $u_k = \mathbf{w}(A_k^{(W_k,P)})$. Moreover, the numerator of Σ_k is equal to $\mathbf{ucov}(A_k^{(W_k,P)})$ and that the numerator of S_k is equal to $\mathbf{ucov}(A_k^{(W_k,P)})$. Observe that $\mathtt{w}_{\max}^{(X)} = 1$, while $\mathtt{w}_{\max}^{(W_k)} = \zeta_k^{max}$ for all $k \in [K]$. Besides, $\mathtt{r}_i(W_k) = \mathtt{r}_i(X)$ for all $k \in [K]$ and $i \in [D]$.

Apply Lemma 3.14 with respect to X, for each $k \in [K]$. Apply Lemma 3.14 with respect to W_k, for each $k \in [K]$. Apply Lemma 3.16 with respect to W_k, for each $d \in [D]$ and $k \in [K]$. Apply Lemma 15.3 with respect to X, for each $k \in [K]$. Combine these bounds via the union bound. As we apply Lemma 3.14 for $2K$ times instead of K times, the overall probability of success becomes $1 - K \cdot (\mathbf{2} + D + D(D+1)/2) \cdot \delta$.

To prove (15.2), observe that, due to Lemma 3.14 (applied to X), we have

$$|W_k - w_k| = \frac{\left|\mathbf{w}\left(A_k^{(X,P)}\right) - \mathbf{w}\left(A_k^{(X,Z)}\right)\right|}{|X|} \leq \frac{\sqrt{a_\delta}\sqrt{\mathbf{w}(A_k^{(X,P)})}}{|X|} = \frac{\sqrt{a_\delta}}{\sqrt{r_k}} \cdot \frac{r_k}{|X|} = \frac{\sqrt{a_\delta}}{\sqrt{r_k}} w_k \ .$$

(15.3) follows analogously to the respective claim in Theorem 3.23. The proof of (15.4) is also similar to the proof of the respective claim in that theorem: Let $\mathfrak{m}_k \coloneqq (\mu_k - M_k)(\mu_k - M_k)^T$ and, as in the theorem, write $y_{nk} \coloneqq (x_n - \mu_k)(x_n - \mu_k)^T$ for each $n \in [N]$. Due to Lemma 2.21, we have

$$S_k = \frac{\mathbf{ucov}(A_k^{(W_k,Z)})}{\mathbf{w}(A_k^{(X,Z)})} = \frac{\mathbf{ucov}(A_k^{(W_k,Z)}, \mu_k)}{\mathbf{w}(A_k^{(X,Z)})} - \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \mathfrak{m}_k \ .$$

Hence,

$$\begin{aligned}
\left|(S_k - \Sigma_k)_{ij}\right| &= \left| \left(\frac{\mathbf{ucov}(A_k^{(W_k,Z)}, \mu_k)}{\mathbf{w}(A_k^{(X,Z)})} - \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \mathfrak{m}_k - \Sigma_k \right)_{ij} \right| \\
&= \left| \left(\frac{\sum_{n=1}^N z_{nk}\zeta_{nk}(x_n - \mu_k)(x_n - \mu_k)^T}{\mathbf{w}(A_k^{(X,Z)})} - \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \mathfrak{m}_k - \Sigma_k \right)_{ij} \right| \\
&= \left| \left(\frac{\sum_{n=1}^N z_{nk}\zeta_{nk} y_{nk}}{\mathbf{w}(A_k^{(X,Z)})} - \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \mathfrak{m}_k - \Sigma_k \right)_{ij} \right|
\end{aligned}$$

$$= \left| \left(\frac{\sum_{n=1}^{N} z_{nk} (\zeta_{nk} y_{nk} - \Sigma_k)}{\mathbf{w}(A_k^{(X,Z)})} - \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \mathfrak{m}_k \right)_{ij} \right|$$
$$\leq \frac{\left| \left(\sum_{n=1}^{N} z_{nk} (\zeta_{nk} y_{nk} - \Sigma_k) \right) \right|_{ij}}{\mathbf{w}(A_k^{(X,Z)})} + \frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}(A_k^{(X,Z)})} \left| (\mathfrak{m}_k)_{ij} \right| .$$

Due to Lemma 3.16 (applied to W_k), we have

$$\left| (\mathfrak{m}_k)_{ij} \right| = \left| (\mu_k - M_k)_i \right| \cdot \left| (\mu_k - M_k) \right|_j \leq \frac{\lambda_{(ki)}^{(\mu)} \lambda_{(kj)}^{(\mu)}}{\left(\sqrt{u_k} - \sqrt{a_\delta \zeta_k^{max}} \right)^2} \cdot \frac{\tau_{ki} \tau_{kj}}{u_k} .$$

Due to Lemma 3.14 (applied to X and W_k, respectively), we know that

$$\frac{\mathbf{w}\left(A_k^{(W_k,Z)}\right)}{\mathbf{w}\left(A_k^{(X,Z)}\right)} \leq \frac{u_k + a_\delta \zeta_k^{max} \sqrt{u_k}}{r_k - a_\delta \sqrt{r_k}} = \frac{\sqrt{u_k} + a_\delta \zeta_k^{max}}{\sqrt{r_k} - \sqrt{a_\delta}} \cdot \frac{\sqrt{u_k}}{\sqrt{r_k}}$$

and, in particular, $R_k \geq \left(\sqrt{r_k} - \sqrt{a_\delta} \right) \cdot \sqrt{r_k}$. Due to Lemma 15.3 (applied to X), we have

$$\left| \left(\sum_{n=1}^{N} z_{nk} (\zeta_{nk} y_{nk} - \Sigma_k) \right) \right|_{ij} \leq \lambda_{(kij)}^{(\Sigma)} \cdot \rho_{kij} .$$

A combination of these inequalities yields the claim.

□

15.5.2 Special Case: Gaussian Mixture Models (GMMs)

In the special case where the EM* and SEM* algorithm describe the classical EM and SEM algorithm for GMMs, Theorem 15.2 simplifies slightly. In this case, the given density function is Gaussian and we have $\zeta_{nk} = 1$ for all $n \in [N]$ and $k \in [K]$.

Observe that the posterior probabilities $P = (p_{nk})_{n \in [N], k \in [K]}$, which both algorithms compute, describe a soft K-clustering of X. More precisely, P is the soft clustering induced by the GMM θ^{old} (see Section 14.1.2).

First, let us consider the initial condition from (15.1). We have $\zeta_k^{max} = 1$ for all $k \in [K]$ and $\mathtt{w}_{\max}^{(X)} = 1$. Consequently,

$$\forall k \in [K] : \; r_k = u_k = \sum_{n=1}^{N} p_{nk} \cdot 1 = \mathbf{w}\left(A_k^{(X,P)}\right) .$$

This means that the two bounds from (15.1) coincide. Both demand that the weight of the k-th soft cluster of X given by P is

$$\mathbf{w}\left(A_k^{(X,P)}\right) \geq 2 \ln(2/\delta) = a_\delta$$

at least some value that (only) depends on the probability of success. To sum up, it is essential that each cluster does not have too small a weight.

Now, consider the difference of the parameter updates. First, consider the weight updates. With a certain probability, the difference is at most

$$|W_k - w_k| \leq \sqrt{\frac{2 \ln(2/\delta)}{\mathbf{w}\left(A_k^{(X,P)}\right)}} \cdot \mathbf{w}_k .$$

To sum up, the weight updates are similar if the cluster weights $\mathbf{w}\left(A_k^{(X,P)}\right)$ are large enough. Second, consider the bound on the mean updates. As in Section 3.5.3, observe that

$$\sum_{d=1}^{D} \tau_{kd}^2 = \sum_{n=1}^{N} p_{nk}(1-p_{nk}) \cdot 1 \cdot \left\| x_n - \mu_k \right\|_2^2 \leq \mathbf{d}\left(A_k^{(X,P)}\right). \tag{15.7}$$

Thus, the unit of measurement which Theorem 15.2 effectively uses to measure the squared distance $\left\| \mu_k - M_k \right\|_2^2$ is at most

$$\sum_{d=1}^{D} \left(\frac{\tau_{kd}}{\sqrt{u_k}} \right)^2 = \frac{\sum_{d=1}^{D} \tau_{kd}^2}{u_k} \leq \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} = \mathbf{var}\left(A_k^{(X,P)}\right)$$

the variance of the respective soft cluster. Moreover, observe that this unit of measurement is multiplied with the factor

$$\frac{(\lambda_{(kd)}^{(\mu)})^2}{(\sqrt{u_k} - a_\delta \zeta_k^{max})^2} = \frac{2e\ln(2/\delta)}{\left(\mathbf{w}\left(A_k^{(X,P)}\right) - \sqrt{2e\ln(2/\delta)}\right)^2}, \tag{15.8}$$

if the term τ_{kd}^2 is large enough. Intuitively, τ_{kd} should be large enough if the points in the soft cluster $A_k^{(X,P)}$ are scattered enough. That is, τ_{kd} should be large if the cost $\mathbf{d}\left(A_k^{(X,P)}\right)$ of the soft cluster is large enough. However, strictly speaking, τ_{kd}^2 is still a lower bound on the cost (see (15.7)). Nonetheless, the resulting factor from (15.8) is small if the weight $\mathbf{w}\left(A_k^{(X,P)}\right)$ of the soft cluster is large enough.

Third, consider the bound on a covariance update. Again, the bound becomes tighter when the weight $\mathbf{w}\left(A_k^{(X,P)}\right)$ of the soft cluster is large enough. However, to the best of our knowledge, there is no concise interpretation of the unit of measurement in this bound.

To sum up, it is essential that the weights of the soft clusters of X that are induced by the given GMM θ^{old} are large enough.

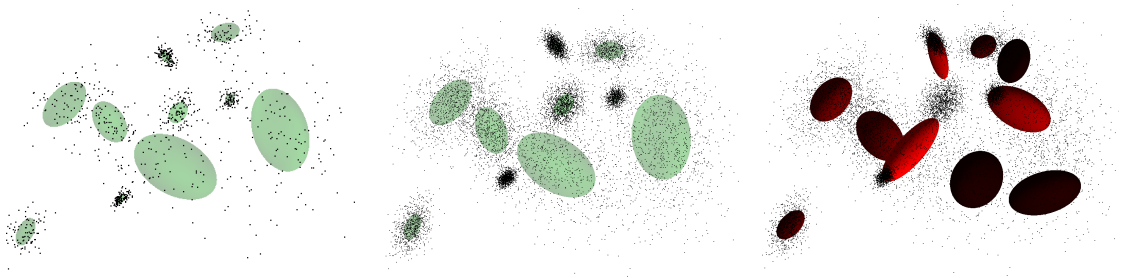
15.6 Some Concrete Examples

In this section, we illustrate our results with respect to the following two mixture models and the corresponding instantiations of Algorithm 20 and Algorithm 21: First, we consider Gaussian mixture models (GMM) and instantiations that correspond to the classical EM and SEM algorithm for GMMs. Second, we consider Laplacian mixture models (LMM) and instantiations that are EM-like and SEM-like heuristics for LMMs (see Section 15.2).

For our examples, we use the artificial data sets that are illustrated in Figure 15.1. In the following, $X_{K,D,N}$ denotes an artificial data set that has been generated by drawing N points according to a fixed D-variate mixture $\theta_{K,D}$ with K components. That is, for every $N \in \mathbb{N}$, the data set $X_{K,D,N}$ has been generated according to the same mixture model $\theta_{K,D}$. For more details on our data generation method, we refer to our explanations in Section 16.5.2.

A Direct Comparison

We start with a straightforward comparison: We consider the sequences of parameters computed by the EM* algorithm (Algorithm 20) and the SEM* algorithm (Algorithm 21) and their likelihoods. We feed both algorithms with the same initial solution and run each algorithm for 50 rounds. After each round, we compute the differences between their current solutions. More precisely, we compute the absolute difference $|w_k - W_k|$, the



(a) $X_{10,3,N}$, $N = 1\,000$ (b) $X_{10,3,N}$, $N = 10\,000$ (c) $X_{10,3,N}$, $N = 10\,000$, initial GMM

Figure 15.1: Illustration of our data sets. We depict orthogonal projections to randomly chosen 2-dimensional subspaces. The green ellipses indicate the covariance matrices of the underlying GMM. The red ellipses in Figure 15.1c indicate the covariance matrices of an initial solution (computed via the `Unif` method, as described in Section 16.3).

Euclidean distance $\|\mu_k - M_k\|_2$, and the Frobenius norm $\|\Sigma_k - S_k\|_F$. Moreover, we compute the negative log-likelihoods of the solutions, which we denote as "cost" in the following.

The value of a Euclidean distance between mean vectors and the value of a Frobenius norm of a difference between covariance matrices are completely meaningless, as long as we cannot compare them to something. To get a rough idea of how to interpret them, we compare them with the following values: We compare the Euclidean distances between mean vectors with the diameter $\texttt{diam}(X)$ of the respective data set X (see Definition 2.18). We compare the Frobenius norm of the difference between covariance matrices with the Frobenius norm of the covariance $\|\mathbf{cov}(X)\|_F$ of the given data set X (see Definition 2.15). The concrete values of $\texttt{diam}(X)$ and $\|\mathbf{cov}(X)\|_F$ of the respective data set X can be found in the caption of the figure that depicts the results.

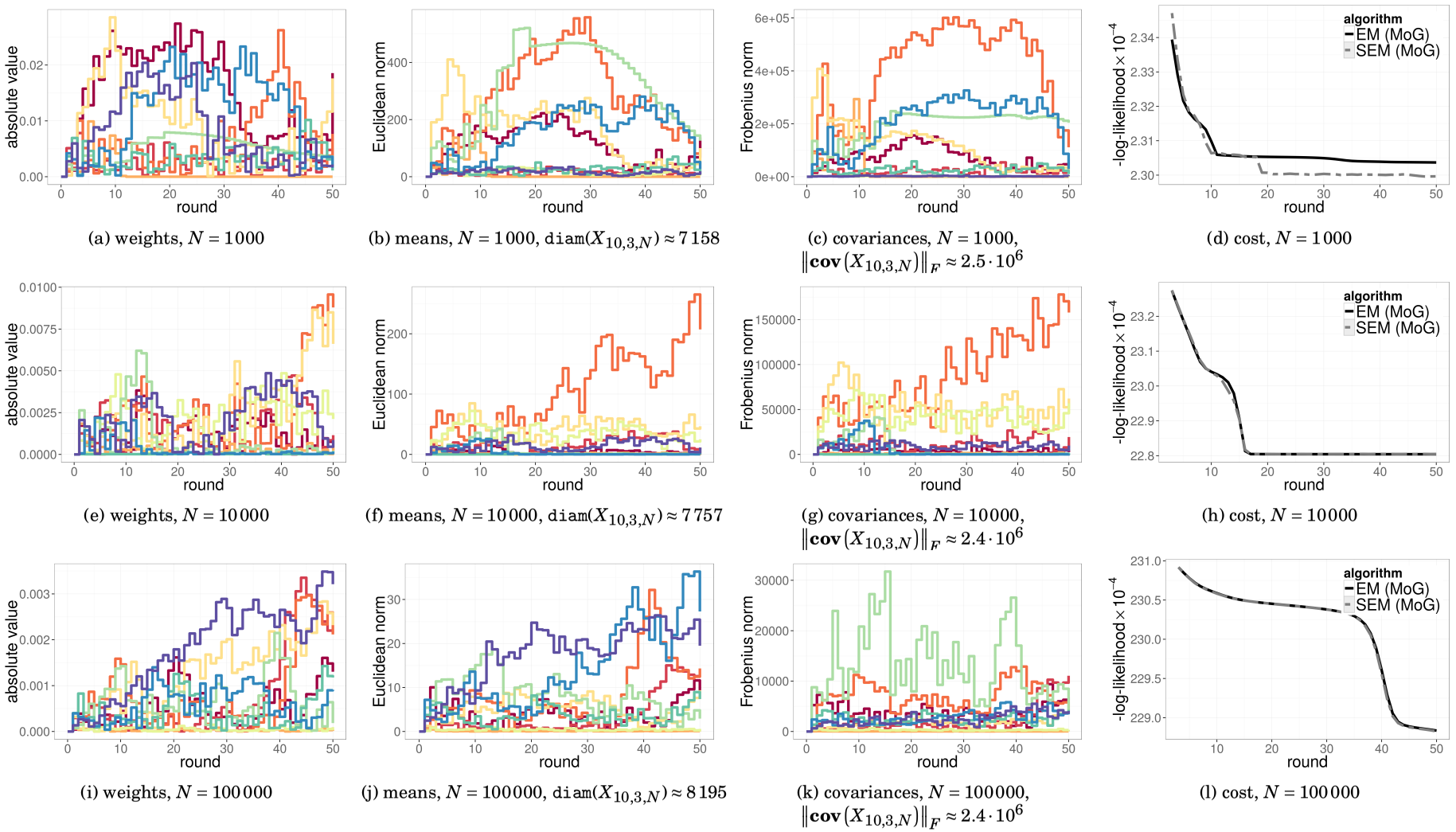
The following examples show that, as to be expected, an SEM* update step is very similar to an EM* update step if the data set is large. However, larger deviations between parameters do not necessary imply that a solution obtained via the SEM* algorithm has a smaller likelihood. Besides that, recall that the EM algorithm is guaranteed to produce solutions with a non-decreasing likelihood (Bishop, 2006). In the following examples, we observe the same for the EM-like EM* algorithm for LMMs, which is actually *no* instantiation of the EM algorithm for LMMs.

Figure 15.2 depicts a comparison of the EM and SEM algorithm for Gaussian mixture models (GMMs). In general, we observe that in comparison to $\texttt{diam}(X_{10,3,N})$ and $\|\mathbf{cov}(X_{10,3,N})\|_F$ (see caption of the respective figure), the parameters computed by the EM and SEM algorithm are similar. It is clearly visible that the differences between the parameters become smaller when the number of points N becomes larger. However, this does not necessarily result in different likelihood values.

Figure 15.3 shows our results regarding the EM-like and SEM-like algorithm for Laplacian mixture models (LMMs). In general, the differences between the parameters are much larger than those we obtained in our comparison between the EM and SEM algorithm for GMMs (i.e., roughly by a factor of 10). Nonetheless, the SEM-like algorithm for LMMs does not necessarily produce solutions that are less likely than those computed by the EM-like algorithm for LMMs. Again, the differences between the parameters decrease for a larger number of observations.

An Illustration of Our Theoretical Bounds

In this section, we illustrate a tighter variant of Theorem 15.2 that only bounds the proximity of the mean updates. To this end, we proceed as follows: First, we compute an initial solution

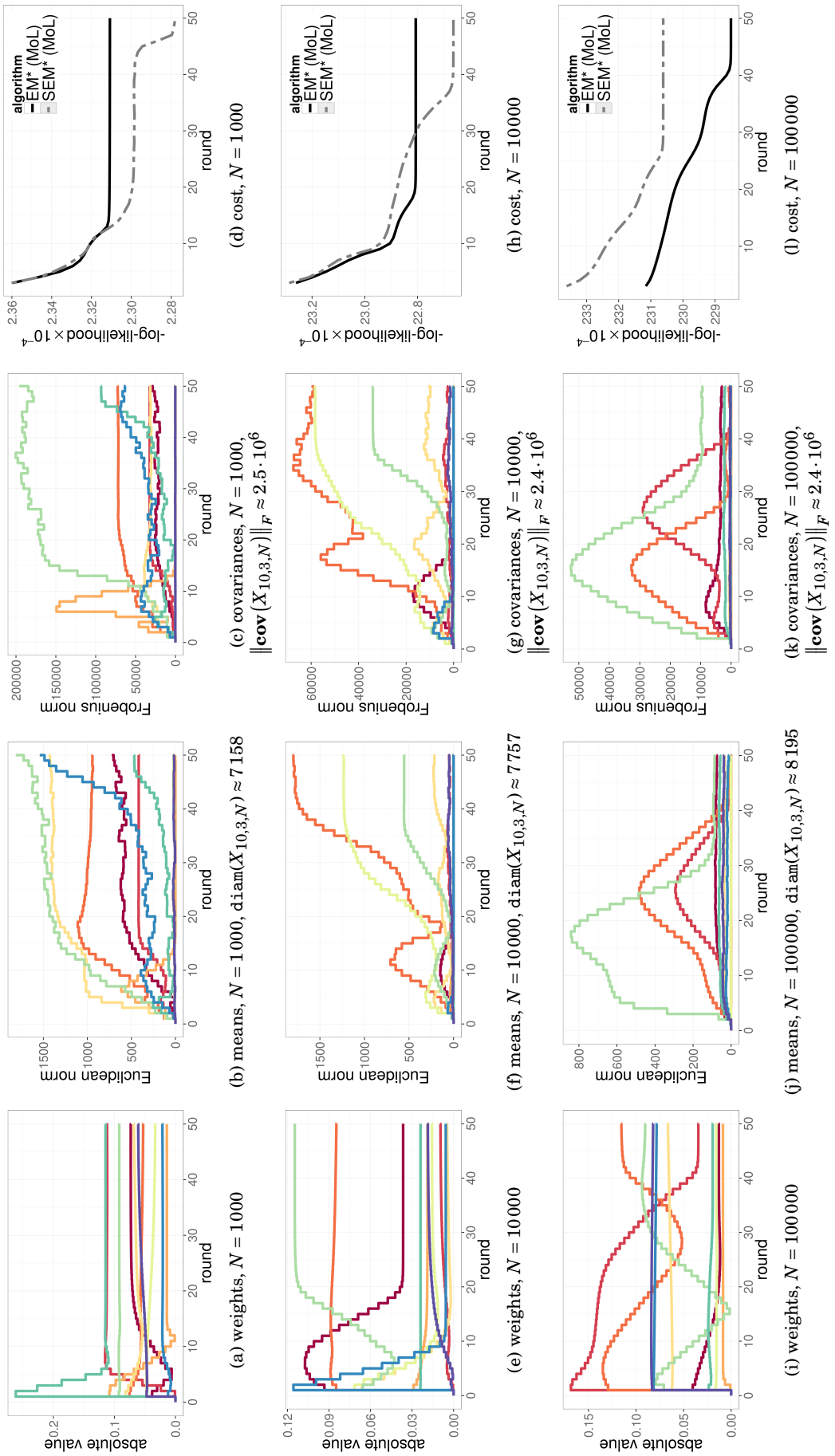


(a) weights, $N = 1000$ (b) means, $N = 1000$, $\mathtt{diam}(X_{10,3,N}) \approx 7158$ (c) covariances, $N = 1000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.5 \cdot 10^6$ (d) cost, $N = 1000$

(e) weights, $N = 10000$ (f) means, $N = 10000$, $\mathtt{diam}(X_{10,3,N}) \approx 7757$ (g) covariances, $N = 10000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.4 \cdot 10^6$ (h) cost, $N = 10000$

(i) weights, $N = 100000$ (j) means, $N = 100000$, $\mathtt{diam}(X_{10,3,N}) \approx 8195$ (k) covariances, $N = 100000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.4 \cdot 10^6$ (l) cost, $N = 100000$

Figure 15.2: Comparison of the EM and SEM algorithm for GMMs. Both algorithms were executed once on the artificial data set $X_{10,3,N}$, given the same initial solution. Each line corresponds to the difference between parameters with the same index k.

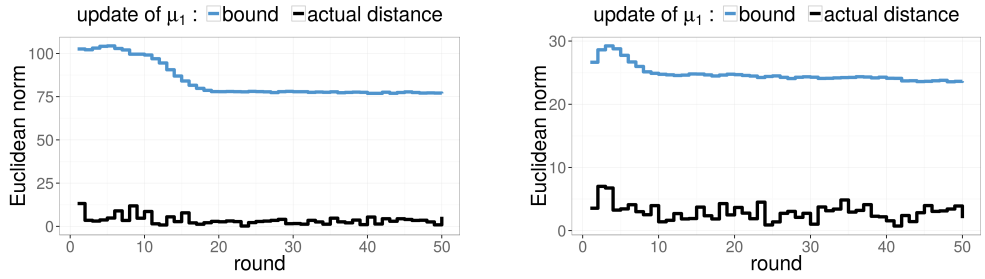


(a) weights, $N = 1000$ (b) means, $N = 1000$, $\texttt{diam}(X_{10,3,N}) \approx 7158$ (c) covariances, $N = 1000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.5 \cdot 10^6$ (d) cost, $N = 1000$

(e) weights, $N = 10000$ (f) means, $N = 10000$, $\texttt{diam}(X_{10,3,N}) \approx 7757$ (g) covariances, $N = 10000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.4 \cdot 10^6$ (h) cost, $N = 10000$

(i) weights, $N = 100000$ (j) means, $N = 100000$, $\texttt{diam}(X_{10,3,N}) \approx 8195$ (k) covariances, $N = 100000$, $\|\mathbf{cov}(X_{10,3,N})\|_F \approx 2.4 \cdot 10^6$ (l) cost, $N = 100000$

Figure 15.3: Comparison of the EM-like and SEM-like heuristic for LMMs. Both algorithms were executed once on the artificial data set $X_{10,3,N}$, given the same initial solution. Each line corresponds to the difference between the parameters of the component with the same index k.



(a) $\texttt{diam}(X_{10,3,N}) \approx 7\,757$, $N = 10\,000$ (b) $\texttt{diam}(X_{10,3,N}) \approx 8\,195$, $N = 100\,000$

Figure 15.4: Experimental difference and theoretical bound on the difference between the mean updates for Gaussian mixture models (GMMs).

θ^{old}. Then we compute the mean updates as described in Algorithm 20 and Algorithm 21 for the given θ^{old}, respectively. We want to compare the Euclidean distance between these mean updates with our theoretical bound. Hence, we additionally compute a bound on the difference of the mean updates (given θ^{old}) that holds with high probability. More precisely, we apply the bounds from the following Corollary 15.4 with $\epsilon = \frac{1}{100}$.

Corollary 15.4 (proximity of mean updates)**.** *Consider the same setting as in Theorem 15.2. Fix $\epsilon \in (0,1)$. Set $\delta := \epsilon/K(D+1)$. If $u_k = \sum_{n=1}^{N} p_{nk}\zeta_{nk} \geq (3\ln(2/\delta))\zeta_k^{max}$ for all $k \in [K]$, then with probability $1-\epsilon$ we have*

$$\forall k \in [K]: \ \left\|M_k - \mu_k\right\|_2^2 \leq \sum_{d=1}^{D} \left(\frac{\lambda_{(kd)}^{(\mu)}}{\sqrt{u_k} - \sqrt{3\ln(2/\delta)\zeta_k^{max}}} \right)^2 \cdot \frac{\sum_{n=1}^{N} p_{nk}(1-p_{nk})\zeta_{nk}^2 (x_n - \mu_k)_d^2}{\sum_{n=1}^{N} p_{nk}\zeta_{nk}}.$$

After that, we generate a new model θ^{old} because our analysis only covers the comparison of a single update step of both algorithms (both started with the same θ^{old}). Instead of computing a new initial solution θ^{old} from scratch, we compute a new model θ by applying a complete update step according to Algorithm 21 to the given θ^{old} and repeat our evaluation with θ^{old} set to θ.

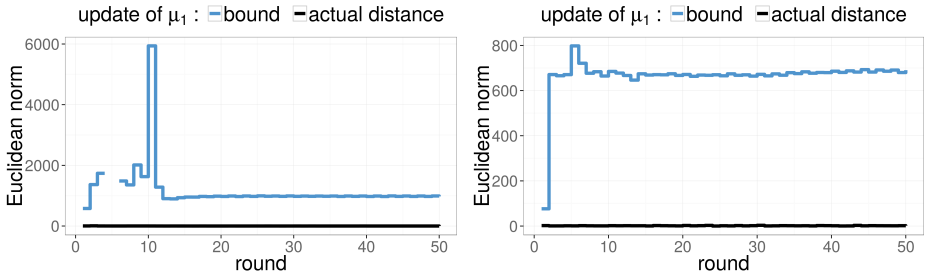
Figure 15.4 depicts our results with respect to GMMs. We observe that the actual difference is significantly smaller than our bound, as it is to be expected. However, our bounds on the differences become tighter when the number of points becomes larger. This matches the fact that a larger number of observations results in a larger u_k value which in turn yields tighter bounds. For the data set $X_{10,3,N}$ with N=1 000, our bounds were not applicable.

Figure 15.5 depicts results with respect to LMMs. Again, we do not depict results for the data set $X_{10,3,N}$ with $N = 1\,000$, since our bounds were not applicable at all. For $N = 10\,000$, there are missing values in Figure 15.5a since our bounds were not applicable in each round. For $N = 100\,000$, our bound was applicable in each round and, as to be expected, significantly tighter than the bound for $N = 10\,000$.

15.7 Discussion

Unsurprisingly, for Gaussian mixture modes, our probabilistic bounds confirm the intuition that an SEM* update step imitates the EM* update step well if the soft clusters (given by the current model θ^{old}) do not have too small a weight. Apart from that, our probabilistic bound does not guarantee much.

First, our bound from Theorem 15.2 is not particularly tight. One can also see this from the examples that we will provide in Section 15.6. There, we evaluate a bound on the



(a) $\texttt{diam}(X_{10,3,N}) \approx 7757$, $N = 10000$ (b) $\texttt{diam}(X_{10,3,N}) \approx 8195$, $N = 100000$

Figure 15.5: Experimental difference and theoretical bound on the difference between the mean updates for Laplacian mixture models (LMMs).

proximity of the mean updates (which is tighter than the bound in Theorem 15.2 because we simply ignore the proximity of the covariance updates).

Second, our bounds are only valid for a single update step. They do not guarantee that several consecutive update steps still perform similarly for certain initial solutions and data sets. Note that for consecutive update steps one cannot simply repeatedly apply Theorem 15.2 because it assumes that both algorithms are given the same model θ^{old}, while the very first update step of both algorithms will certainly already yield two different models.

Third, we do not expect that monitoring the conditions given in Theorem 15.2 during a run of the SEM* algorithm and switching to the update steps of the EM* algorithm (in case the SEM* update step might deviate too far from the EM* update step) is useful. This is simply due to the fact that the evaluation of the conditions from Theorem 15.2 introduces a large overhead.

"The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data."

John Tukey[1]

Chapter 16

Adaptive Seeding for Gaussian Mixture Models

The performance of the expectation-maximization (EM) algorithm for Gaussian mixture models (GMMs) crucially depends on its initial solution. Hence, numerous initialization methods have been proposed: On the one hand, there are simple random methods that, for instance, sample mean vectors uniformly from the given set of observations. On the other hand, there are rather complex methods that crucially depend on the right choice of hyperparameters. None of these methods comes with a performance guarantee.

Simply speaking, an initialization method is a fast algorithm that computes a good solution. For other clustering problems, there are such algorithms: For the K-means problem, we have the K-means++ algorithm by Arthur and Vassilvitskii (2007). For the discrete radius K-clustering problem, there is an algorithm by Gonzalez (1985). Both algorithms are fast and work provably well. Moreover, both have already been utilized to determine the mean vectors of an initial GMM. In this chapter, we aim to continue this work and modify the K-means++ algorithm and Gonzalez' algorithm further.

Overview. In Section 16.1, we give an overview of common initialization methods for the EM algorithm for GMMs. In Section 16.2, we briefly state our main contribution. In Section 16.3, we give a more detailed description of those algorithms to which we compare our methods. In Section 16.4, we introduce our adaptive seeding methods, which use ideas from the K-means++ algorithm and Gonzalez' algorithm. Finally, in Section 16.5, we compare our adaptive seeding methods with existing algorithms with respect to large sets of artificial data sets and some real-world data sets.

Publication. The following results have been published in (Blömer and Bujna, 2016).

16.1 Related Work

The problem of initializing the EM algorithm for GMMs properly is well-known. Only recently, Jin et al. (2016) published a theoretical analysis of local maxima in the likelihood of GMMs that also stresses the necessity of a careful initialization, even "in highly favorable settings". In the following, we give a brief overview of initialization methods. More information can be found in (Maitra, 2009; Thiesson, 1995; Fayyad et al., 1998; Biernacki, 2004), for instance.

A common way to initialize the EM algorithm is to first draw mean vectors uniformly at random from the input set and then to approximate covariances and weights (Biernacki,

[1] Source: Sunset salvo. The American Statistician 40 (1). Online at `http://www.jstor.org/pss/2683137`

2004; Melnykov and Melnykov, 2011; Maitra, 2009; Meilă and Heckerman, 1998). To compensate for the random choice of initial means, several candidate solutions are created and the one with the largest likelihood is chosen. Often, few steps of the EM, Classification EM, Stochastic EM algorithm, or Lloyd's K-means algorithm are applied to the candidates (Bishop, 2006, p. 427).

Other popular initializations are based on hierarchical agglomerative clustering (HAC). For instance, in (Melnykov and Melnykov, 2011; Maitra, 2009; Meilă and Heckerman, 1998), HAC (with different distance measures) is used to obtain mean vectors. Since HAC is generally very slow, it is usually only executed on a random sample. However, the size of any reasonable sample depends on the size of the smallest optimal component, which is unknown. Moreover, it is often outperformed by other methods (Melnykov and Melnykov, 2011; Meilă and Heckerman, 1998). An approach that tries to speed up HAC is presented in (Maitra, 2009). It aims at finding the best local modes of the data set in a reduced m^*-dimensional space and applies HAC only on these modes. However, this method is time-consuming and the choice of m^* is crucial (Maitra, 2009, p. 5,13). Moreover, in (Maitra and Melnykov, 2010) it is outperformed by simple random methods.

Melnykov and Melnykov (2011) present a density based approach that not only determines an initial solution but also tries to determine the correct number of components. It initializes the means by points which have a "high concentration" of neighbors. To this end, the size m of the neighborhood of a point (i.e., the minimum number of points in a cluster) has to be fixed in advance. In our experiments, we found that the performance crucially depends on the choice of m.

In Verbeek et al. (2003), a greedy algorithm is presented which constructs a sequence of mixture models with 1 through K components. Given a model θ_k with k components, it constructs several new candidates with $k+1$ components. Each candidate is constructed by adding a new component to θ_k and executing the EM algorithm. Hence, this method is rather expensive.

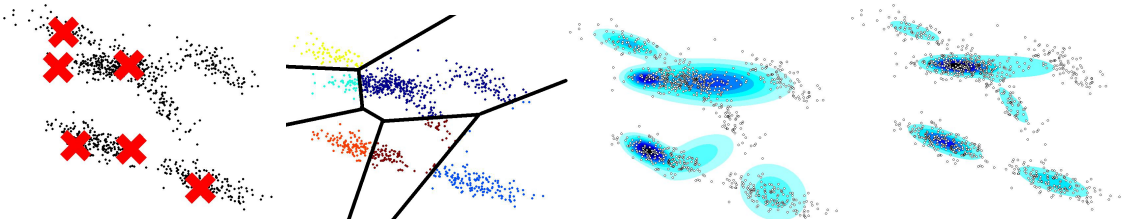
In Kwedlo (2015), a modification of the Gonzalez algorithm for GMMs is presented. First, it estimates covariances and weights randomly. Then it uses a variant of the algorithm from (Gonzalez, 1985) to determine initial mean vectors. Furthermore, there are some practical applications that use the K-means++ algorithm for the initialization of GMMs. For instance, in (Krüger et al., 2010), it is applied in the context of speech recognition.

16.2 Our Contribution

Clearly, there is no way to determine the best initialization algorithm that outperforms all other algorithms on all instances. The performance depends on our notion of whether it performs well, the given data, and the allowed computational cost. Nonetheless, the initialization methods presented so far (except the simple random initializations) face mainly two problems: First, they are rather complex and time consuming. Second, the choice of hyperparameters is crucial for the outcome. In this chapter, we present new methods that are fast and do not require choosing sensitive hyperparameters. These methods make use of the ideas behind the K-means++ algorithm (Arthur and Vassilvitskii, 2007) and the Gonzalez algorithm (Gonzalez, 1985). Thereby, we continue the work of Kwedlo (2015) and Krüger et al. (2010).

16.3 Baseline Algorithms

Due to the large number of initialization methods, our comparison only focuses on the most common ones. To the best of our knowledge, the most widely used initializations consist of the following steps, which we illustrate in Figure 16.1.



(a) Choose K points (b) K-means partition (c) Construct a GMM (d) EM algorithm

Figure 16.1: An illustration of the baseline algorithm `Gonz` (without additional preprocessing via Lloyd's algorithm) and a subsequent run of the EM algorithm for GMMs.

Choosing K Points. First, they choose K points from the given set of observations. We consider the following four methods:

`Unif` draws K points independently and uniformly at random from X.

`HAC` computes a uniform sample S of size $s \cdot |X|$ of the input set X and executes hierarchical K-clustering with average linkage cost on S.

`Gonz` executes the algorithm by Gonzalez (1985), which yields a 2-approximation for the discrete radius K-clustering problem. Iteratively, it chooses the point with the largest cost (i.e., the minimum Euclidean distance) with respect to the already chosen means and adds it to its set of means.

That is, given means $\mu_1, \dots, \mu_{k-1}$, it chooses

$$\mu_k \in \arg\max \left\{ \texttt{dist}\left(x, \{\mu_1, \dots, \mu_{k-1}\}\right) \mid x \in X \right\} , \tag{16.1}$$

where we re-use Definition 9.1 and write

$$\texttt{dist}\left(x, \{\mu_1, \dots, \mu_{k-1}\}\right) = \min \left\{ \left\| x - \mu_i \right\|_2 \mid i \in [k-1] \right\} . \tag{16.2}$$

`KM++` executes the K-Means++ algorithm by Arthur and Vassilvitskii (2007), which has been designed for the K-means problem. In each round, `KM++` samples a data point p from the given data set X with probability proportional to its K-means cost (with respect to the points chosen so far) and adds this point as the next mean.

That is, given means $\mu_1, \dots, \mu_{k-1}$, it chooses μ_k randomly according to

$$\Pr(\mu_k = x) = \frac{\texttt{dist}\left(x, \{\mu_1, \dots, \mu_{k-1}\}\right)^2}{\sum_{y \in X} \texttt{dist}\left(y, \{\mu_1, \dots, \mu_{k-1}\}\right)^2} , \tag{16.3}$$

where $\texttt{dist}(\cdot, \cdot)$ is defined in (16.2). In expectation, the resulting solution is a $\mathcal{O}(\log(K))$-approximation to the K-means problem. `KM++` is particularly interesting since the K-means algorithm is a special (limit) case of the EM algorithm (Bishop, 2006).

(Optional) Preprocessing. Some methods process the K chosen points further, before generating a GMM: They feed the K points to Lloyd's K-means algorithm as initial cluster means and then use the resulting set of K points hereafter (Bishop, 2006, p. 427, p. 438). As this preprocessing is quite popular, it seems to improve the initial solution and reduce the risk of getting stuck at a poor local minimum later on.

Notation 16.1. *We refer to the K-means algorithm as an* intermediate *algorithm and indicate its use by the postfix "*$_{\texttt{km}}$*". For instance, we write* $\texttt{Gonz}_{\texttt{km}}$.

Construction of a K-GMM. Finally, the K points are used to create a GMM. To this end, one determines the K-means hard clustering of X that is induced by the K points. Then for each of the resulting hard clusters, one computes a single Gaussian with maximum likelihood (cf. Lemma 14.12) and uses this Gaussian as a single component of the GMM. The weight of a component is estimated by the relative number of points in the respective hard cluster. Algorithm 22 describes this approach in detail (including our error handling in case the estimate of the covariance is not positive definite).

Algorithm 22 Construction of a GMM

Require: $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $\left(\mu_l\right)_{l\in[k]} \subset \mathbb{R}^D$

1: Determine the k-means hard clustering $A_1, \dots, A_k$ of X induced by $\left(\mu_l\right)_{l\in[k]}$.
2: **for** $l = 1, \dots, k$ **do**
3: Set $w_l := \frac{|A_l|}{|X|}$, $\mu_l := \mathbf{m}(A_l)$ and $\Sigma_l := \mathbf{cov}(A_l)$.
4: If Σ_l is not positive definite, then let $\Sigma_l := \frac{\mathbf{var}(A_l)}{D} \cdot I_D$.
5: If Σ_l is still not positive definite, set $\Sigma_l := I_D$.
6: **return** $\theta = \left((w_l, \mu_l, \Sigma_l)\right)_{l=1,\dots,k}$.

16.4 Adaptive Seeding for GMMs

We want to construct a sequence of GMMs with $k = 1$ through $k = K$ components adaptively. Given a GMM θ_{k-1} with $k-1$ components, we want to choose a point from the data set that is a good representative for those points that are not described well by θ_{k-1}. We hope that such a point is a good representative of a component of an optimal k-GMM that is not well described by θ_{k-1}. Then we want to use this point p and the old model θ_{k-1} to construct a GMM θ_k with k components.

Overview. In Section 16.4.1, we discuss several possible ways of choosing a data point p, given some current GMM θ_{k-1} with $k-1$ components. In Section 16.4.2, we describe how we use the chosen point p and θ_{k-1} to construct a GMM θ_k with k components. Additionally, in Section 16.4.3, we propose some post-processing of our initial GMM, which we execute before we execute the EM algorithm. In Section 16.4.4 we briefly sum up the resulting adaptive seeding methods and compare them with existing methods.

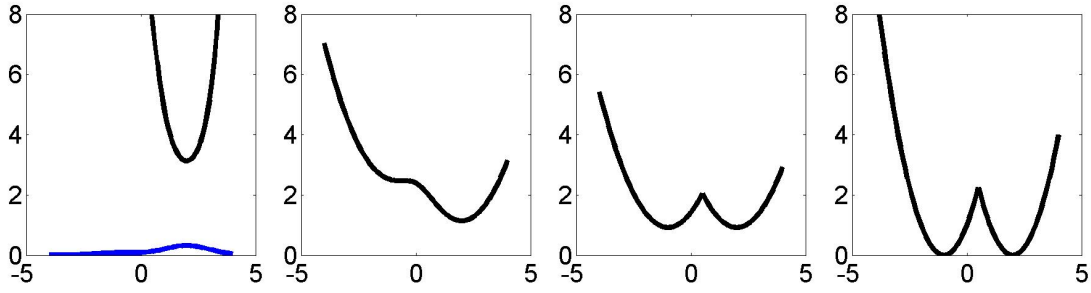
16.4.1 Choosing the Next Point

Assume we are given a GMM θ_{k-1} with $k-1$ components that is a likely solution, compared to other GMMs with $k-1$ components. In this section, we deal with the question of how to choose a point $x \in X$ that is not well described by θ_{k-1} and that is a good representative for other points that are also not well described.

The main idea is to proceed similarly to the K-means++ algorithm and the algorithm by Gonzalez (1985), which we described in Section 16.3. That is, we want to make use of a cost function that describes how *poorly* a point is described by the current model. Then we choose a point either proportional to its cost or choose the point with maximum cost.

Cost Function

We want to use a cost function $m(\cdot|\theta_{k-1}) : X \to [0,\infty)$ that describes how *poorly* a point is described by a GMM θ_{k-1}. Ideally, the cost $m(x|\theta_{k-1})$ of point x is larger than the cost $m(y|\theta_{k-1})$ of point y if the model θ_{k-1} describes y better than x. In other words, if $m(x|\theta_{k-1}) > m(y|\theta_{k-1})$, then we tend to prefer x over y as the next chosen point. Additionally,



(a) $\mathrm{p}(x|\theta_{k-1})^{-1}$ and $\mathrm{p}(x|\theta_{k-1})$ (b) $-\ln\left(\mathrm{p}(x|\theta_{k-1})\right)$ (c) $\min\{-\ln(\mathcal{N}_1(x|-1,1)), -\ln(\mathcal{N}_1(x|2,1))\}$ (d) $\min\{(x-(-1))^2, (x-(+2))^2\}$

Figure 16.2: Examples for different ways to measure how poor a point is described by a GMM. In each figure, we evaluate the respective measure for each $x \in [-4,4]$ with respect to the GMM $\theta_{k-1} = ((0.2,(-1),(1)),(0.8,(2),(1))$.

there is one more requirement to keep in mind: As we want to be able to sample a point with probability proportional to its cost, we require that the cost function m only takes non-negative values.

There are several possible functions that could be used as a cost function m. We illustrate these functions in Figure 16.2.

Inverse? The most obvious choice is the inverse density $\mathrm{p}(x|\theta_{k-1})^{-1}$. It clearly has an exponential behaviour. An example that illustrates this aspect is given in Figure 16.2a. Therefore, we consider it being too strict.

Negative Logarithm? Usually, to avoid the problem of the exponential behaviour (under- and over-flows) of Gaussian densities, one considers the logarithm of this density instead. Obviously, just as the inverse density, the negative logarithm $-\ln(\mathrm{p}(x|\theta_{k-1})$ is also a measure of how poorly x is described by θ_{k-1}. For an illustration, we refer to Figure 16.2b.

There are two problems left: First, this function may take negative values. We saw in Section 14.2.4 that the density function is unbounded from above and, hence, its negative logarithm is unbounded from below. Second, as we already explained in Section 14.2, the logarithm of the likelihood does not scale with the data set and the GMM. Recall that $\ln\left(\mathcal{N}_D(c\cdot x|c\cdot\mu,c^2\cdot\Sigma)\right) = \ln\left(\mathcal{N}_D(x|\mu,\Sigma)\right) - D\ln(c)$ for all $c \in \mathbb{R}_+$ (Section 14.2.3). Therefore, one should obviously not sample data points proportional to this measure.

Component-Wise Negative Logarithm? Next, consider the minimum negative logarithm of the density of a point in a single Gaussian component:

$$\min\left\{-\ln\left((2\pi)^{D/2}|\Sigma_l|^{1/2}\right)+\frac{1}{2}(x-\mu_l)^T\Sigma_l^{-1}(x-\mu_l)\,\middle|\,(w_l,\mu_l,\Sigma_l)\in\theta_{k-1}\right\}.$$

An example is depicted in Figure 16.2c. This measure suffers from the same problems as the negative logarithm of the density of x, given θ_{k-1}. Observe that these problems would vanish if we removed the terms $-\log\left((2\pi)^{D/2}|\Sigma_l|^{1/2}\right)$, which originate from the normalization terms of the Gaussian components. Therefore, we just remove these terms.

A Minimum Mahalanobis Distance! Our previous considerations lead us to the following minimum Mahalanobis distance:

$$m(x|\theta_{k-1}) := \min\left\{(x-\mu_l)^T\Sigma_l^{-1}(x-\mu_l)\,\middle|\,(w_l,\mu_l,\Sigma_l)\in\theta_{k-1}\right\}. \tag{16.4}$$

As the covariance matrices are positive definite[1], we know that $\mathfrak{m}(x|\theta_{k-1})$ is always positive for all x with $\forall l \in [k-1] : x \neq \mu_l$ and equals zero if $x = \mu_l$ for some $l \in [k-1]$ (Golub and Loan, 1996, p. 140). Moreover, it does not exhibit the strong exponential behavior as the (inverse) density. This is also apparent in our example in Figure 16.2b. A potential downside is that this measure does not take into account the mixture weights at all.

With Some Extra Softness

We know that the cost function m, which we defined in (16.4), does *not* exhibit a strong exponential behaviour like the inverse density. Still, we are interested in weakening the effect of our function m further. In other words, we want to reduce the probability to choose an outlier even further. Thus, we additionally consider the following approaches:

In the K-means++ like approach, we sample a point according to the distribution given by $m(x|\theta_{k-1}) / \left(\sum_{y \in X} m(y|\theta_{k-1})\right)$. To make this distribution weaker, we add a constant portion of uniform distribution. That is, we sample a point according to

$$m_\alpha(x|\theta_{k-1}) := \alpha \cdot \frac{m(x|\theta_{k-1})}{\sum_{y \in X} m(y|\theta_{k-1})} + (1-\alpha) \cdot \frac{1}{|X|} \ ,$$

with some $\alpha \in [0,1]$, instead of $m(x)$. Observe that α is some hyperparameter that we have to determine in advance. In our experiments, we just evaluate $\alpha = 0.5$ (and $\alpha = 1$). For our Gonzalez-like initialization, we obviously have to take a different approach: To weaken the effect of always choosing the point that maximizes $m(x)$, we sample a uniform subset of the whole data set X in advance. Here, the size $s \cdot |X|$ of the uniform sample is a hyperparameter that we have to choose in advance. In our experiments, we just test $s = 0.1$ and $s = 1$.

16.4.2 Construction of a k-GMM

Given some chosen point $x \in X$ and the old GMM θ_{k-1} with $(k-1)$ components, we want to construct a GMM with k components. In our very first experiments, we used Algorithm 22 to construct such a GMM. However, the estimation of spherical covariance matrices yielded a better performance than the estimation of full covariance matrices. That is, we replaced the covariance update in Step 3 of Algorithm 22 by a spherical covariance estimate, as described in Algorithm 23 (see also Lemma 14.12). Given the resulting GMM, our methods then again choose a new point from X as described in the previous section.

16.4.3 Post-Processing of the K-GMM

Recall from Section 16.3 that some popular initialization methods use the K-means algorithm as an intermediate algorithm, which is a hard clustering algorithm. Since we do *not* solely construct mean vectors but a (complete) GMM, we apply a hard-clustering variant of the EM algorithm instead, which is known as the Classification EM algorithm (CEM) (Celeux and Govaert, 1992). As our initial GMM is spherical, we choose a spherical variant of the CEM algorithm, which is described in Algorithm 24. We indicate its use by the postfix "`cem`".

16.4.4 Summary and Comparison

Algorithm 25 and Algorithm 26 summarize our methods. We point out that in our experiments we did not optimize the hyperparameters $\alpha \in (0,1]$ and $s \in (0,1]$. Instead, we evaluate the instantiations with $\alpha \in \{0.5, 1\}$ and $s \in \{0.1, 1\}$, respectively.

Let us briefly compare our algorithms with the algorithms by Arthur and Vassilvitskii (2007), Gonzalez (1985), and Kwedlo (2013, 2015). First of all, consider the K-means++

[1] This means that we have to make sure that they are. If not, we are in trouble anyway.

Algorithm 23 Construction of a Spherical GMM

Require: $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $(\mu_l)_{l\in[k]} \subset \mathbb{R}^D$

1: Determine the k-means hard clustering $A_1,\dots,A_k$ of X induced by $(\mu_l)_{l\in[k]}$.
2: **for** $l = 1,\dots,k$ **do**
3: Set $w_l := \frac{|A_l|}{|X|}$, $\mu_l := \mathbf{m}(A_l)$ and $\Sigma_l := \frac{\mathbf{var}(A_l)}{D} \cdot I_D$.
4: If Σ_l is still not positive definite, set $\Sigma_l := I_D$.
5: **return** $\theta = \big((w_l,\mu_l,\Sigma_l)\big)_{l=1,\dots,k}$.

Algorithm 24 Spherical Classification EM (CEM) Update Step (postfix "`cem`")

Require: $X = (x_n)_{n\in[N]} \subseteq \mathbb{R}^D$, $\big((w_k^{old},\mu_k^{old},\Sigma_k^{old})\big)_{k\in[K]}$

1: **for all** $n \in [N]$ and $k \in [K]$ **do**
2: $p_{nk} := \frac{w_k^{old}\mathcal{N}_D(x_n|\mu_k^{old},\Sigma_k^{old})}{\sum_{l=1}^K w_l^{old}\mathcal{N}_D(x_n|\mu_l^{old},\Sigma_l^{old})}$
3: Compute a partition $A_1,\dots,A_K$ of X such that for all $x_n \in X$ we have $p_{nk} \geq p_{nl}$ if $x_n \in X_k$.
4: **for all** $k \in [K]$ **do**
5: $w_k := \frac{|A_k|}{|X|}$, $\mu_k := \mathbf{m}(A_k)$, $\Sigma_k := \frac{\mathbf{var}(A_k)}{D} I_D$
6: **return** $\big((w_k,\mu_k,\Sigma_k)\big)_{k\in[K]}$

algorithm by Arthur and Vassilvitskii (2007) and the algorithm by Gonzalez (1985), which inspired our algorithms. We already described both algorithms in Section 16.3. In each step, the Gonzalez algorithm chooses a point x with the maximum distance $\texttt{dist}(x,C)$ to the already chosen points C, while the K-means++ initialization chooses a point with probability proportional to $\texttt{dist}(x,C)^2$. Hence, both algorithms can be seen as special cases of our initialization method where the covariances matrices are *fixed* to the identity matrix: Formally, for $\theta = \big((w_k,\mu_k,I_D)\big)_k$, we have $m(x|\theta_{k-1}) = \texttt{dist}\big(x,\{\mu_l \mid l \in [k-1]\}\big)^2$. Instead of keeping the covariances fixed to $\Sigma_k = I_D$, our methods estimate the covariance matrices adaptively, along with the mean vectors.

Another algorithm that is similar to ours is the algorithm by Kwedlo (2013, 2015), which is described in Algorithm 27. In our experiments, we denote this algorithm by `KG`. Unlike Algorithm 25, this algorithm fixes weights and covariance matrices *before* it determines the mean vectors. Then it *only* chooses the mean vectors adaptively. To ensure equal conditions in our evaluation, we also tested a version of Algorithm 27 that samples means only from a subset of the input set which has been chosen uniformly at random beforehand.

Algorithm 25 Our Adaption of Gonzalez' Algorithm (`SphericalGonzalez`, `SG`(s), `SG`$(s)_{\texttt{cem}}$)

Require: $X \subset \mathbb{R}^D, K \in \mathbb{N}, s \in \{0.1, 1\}$

1: $\theta_1 :=$ optimal 1-MLE wrt. X
2: If $s < 1$, let S be a uniform sample of X of size $\lceil s \cdot |X| \rceil$. Otherwise, set $S = X$.
3: **for** $k = 2,\dots,K$ **do**
4: Choose a point $p \in \arg\max_{x\in S} m(x|\theta_{k-1})$.
5: $M_k := \big\{\mu \mid (\cdot,\mu,\cdot) \in \theta_{k-1}\big\} \cup \{p\}$
6: Compute a GMM θ_k by applying Algorithm 23 to X and M_k.
7: (only `SG`$(s)_{\texttt{cem}}$) Apply a small number (25) rounds of Algorithm 24
8: **return** θ_K

Algorithm 26 Our Adaption of the K-means++ Algorithm (`Adaptive`, `Ad`(α), `Ad`(α)$_{\texttt{cem}}$)

Require: $X \subset \mathbb{R}^D, K \in \mathbb{N}, \alpha \in \{0.5, 1\}$

1: $\theta_1 :=$ optimal 1-MLE wrt. X
2: **for** $k = 2, \ldots, K$ **do**
3: Sample a point p from X with probability $m_\alpha(p|\theta_{k-1})$.
4: $M_k := \{\mu \mid (\cdot, \mu, \cdot) \in \theta_{k-1}\} \cup \{p\}$
5: Compute a k-GMM θ_k by applying Algorithm 23 to X and M_k
6: (only `Ad`(α)$_{\texttt{cem}}$) Apply a small number (25) rounds of Algorithm 24
7: **return** θ_K

Algorithm 27 Kwedlo's Gonzalez Adaption (`KG`) (Kwedlo, 2013)

Require: $X \subset \mathbb{R}^D, K \in \mathbb{N}, s \in (0, 1]$

1: Draw a uniform sample $\tilde{w}_1, \ldots, \tilde{w}_K$ from $[0, 1]$.
2: **for** $k = 1, \ldots, K$ **do**
3: Set $w_k = \tilde{w}_k / \sum_{l=1}^{k} \tilde{w}_l$.
4: Choose $\lambda_1, \ldots, \lambda_D \in \mathbb{R}$ randomly such that $\sum_{d=1}^{D} \lambda_k = \mathbf{d}(X)/(10 \cdot D \cdot K)$ and $\max\{\lambda_d \mid d = 1, \ldots, D\} / \min\{\lambda_d \mid d = 1, \ldots, D\} \leq 10$. Draw a orthonormal matrix $Q \in \mathbb{R}^{D \times D}$ at random. Set $\Sigma_k := Q^T \operatorname{diag}(\lambda_1, \ldots, \lambda_D) Q$.
5: Let S be a uniform sample of X of size $s \cdot |X|$.
6: Choose $\mu_1 \in S$ uniformly at random and set $\theta_1 = (w_1, \mu_1, \Sigma_1)$.
7: **for** $k = 2, \ldots, K$ **do**
8: Choose $\mu_k \in \arg\min\{m(x|\theta_{k-1}) \mid x \in S\}$ and set $\theta_k := \theta_{k-1} \cup \{(w_k, \mu_k, \Sigma_k)\}$.
9: **return** θ_K

16.5 Evaluation

We evaluated all methods with respect to artificial as well as real-world data sets. Our implementation and the complete results are available at (Bujna, 2016). In the following, we omit the results of those algorithms that are consistently outperformed by others.

16.5.1 Preliminaries

Recall that the EM algorithm has been designed to find a maximum likelihood estimator (MLE). Thus, the likelihood is an obvious measure of the performance. Other measures need to be treated with caution: Some authors consider their methods only with respect to some specific tasks where fitting a GMM to some data is part of some framework. Hence, any observed effects might be due to several reasons and, possibly, not due to the fact that some GMM explains the (presumed) generation of the given data better than another GMM. In particular, GMMs are often compared with respect to certain classifications. As also pointed out by Färber et al. (2010), the class labels of real-world data sets do not necessarily correspond to the structure of an MLE. The same holds for data sets and classifications generated according to some GMM. A cross-validation that examines whether methods over-fit models to training data is certainly reasonable. Nonetheless, we refrain from this evaluation method as our goal is *not* to find a model that does not fit too well to the given data set. This problem should become less important for a large number of observations, though. Moreover, we could generate data sets according to some "ground truth" GMM θ_{gt} and compare GMMs with θ_{gt} because in many cases, in particular for a "small" number of observations, one cannot expect θ_{gt} to be a good surrogate on a maximum likelihood estimate. Again, this problem should become less important for a large number of observations. There are several ways to compare a GMM θ with a ground-truth θ_{gt}: One can

consider the likelihood ratio, the difference in parameters, or some measure that compares the difference of the respective density functions. However, it is not clear how to summarize such comparisons for a large number of experiments. For these reasons, we also refrain from this evaluation method as well and stick with the evaluation of likelihood values. For more information on the evaluation of clustering methods, we refer to von Luxburg et al. (2012).

Besides that, recall that Algorithm 25 and Algorithm 26 have hyperparameters α and s. We did not optimize these parameters, but evaluated $\alpha \in \{0.5, 1\}$ and $s \in \{0.1, 1\}$. Another hyperparameter that is hidden in our formulation of the algorithm is the number of rounds that we execute for the intermediate algorithms. Here, we chose a fixed number of 25 rounds. Besides that, we chose a fixed number of rounds for the EM algorithm: If some intermediate algorithm is applied, then we execute the EM algorithm for 50 rounds. If only the EM algorithm is applied, but no intermediate algorithm, then we execute it for 75 rounds.

16.5.2 Artificial Data Sets

In the following, we first describe our generation of artificial data sets. Then we explain how we evaluated the methods with respect to this large number of data sets. Finally, we state our results with respect to certain sets of data sets.

Generation of Data Sets. We generate data sets by drawing points according to randomly generated GMMs. However, we control the following four properties of the GMMs: First, the Gaussian components of a GMM can either be spherical or elliptical. Formally, we describe the eccentricity of a covariance matrix Σ_k by

$$e_k = \frac{\max_d \lambda_{kd}}{\min_d \lambda_{kd}} ,$$

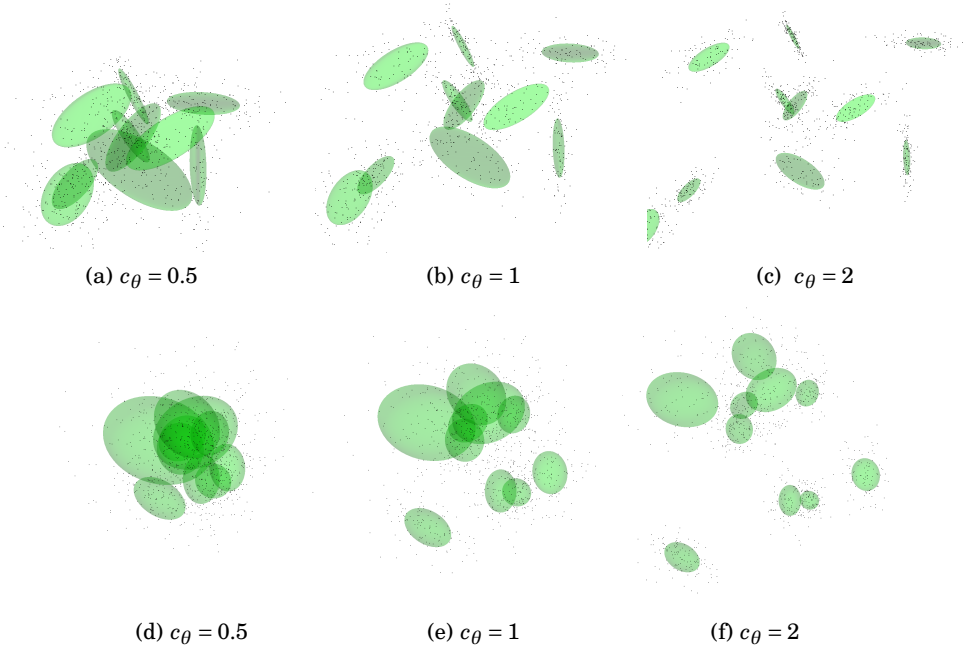
where λ_{kd}^2 denotes the d-th eigenvalue of Σ_k. Second, components can have different sizes, in terms of the smallest eigenvalue of the corresponding covariance matrices. Third, the components can have different or uniform mixture weights $w_1, \dots, w_k$. Fourth, the components can overlap more or less. Following Dasgupta (1999), we define the separation parameter c_θ of a GMM $\theta = \big((w_k, \mu_k, \Sigma_k)\big)_{k \in [K]}$ as

$$c_\theta = \min \left\{ \frac{\|\mu_l - \mu_k\|}{\sqrt{\max\{\operatorname{trace}(\Sigma_l), \operatorname{trace}(\Sigma_k)\}}} \,\middle|\, k, l \in [K],\ k \neq l \right\} .$$

In high dimension $D \gg 1$, $c_\theta = 2$ indicates almost completely separated clusters (i.e., points generated by the same component), while $c_\theta \in \{0.5, 1\}$ indicates a slight but still negligible overlap (Dasgupta and Schulman, 2000). However, in small dimension, $c_\theta \in \{0.5, 1\}$ indicates significant overlaps between clusters, while $c_\theta = 2$ implies rather separated clusters. Figure 16.3 illustrates the effect of the separation parameter.

With these properties of a GMMs in mind, we generate the parameters of a GMM as follows: Initially, we draw K mean vectors independently uniformly at random from a cube with a fixed side length. For the weights, we fix some weight constant $c_w \geq 0$, construct a set of weights $\{2^{c_w \cdot i} / \sum_{j=1}^K 2^{c_w \cdot j}\}_{i=1,\dots,K}$ and assign these weights randomly. To control the sizes and the eccentricity, we fix a minimum λ_{k1} and maximum eigenvalue λ_{kD} and draw the remaining values $\lambda_2, \dots, \lambda_{k(D-1)}$ uniformly at random from the interval. We draw a random orthonormal $(D \times D)$-matrix Q (cf. (Golub and Loan, 1996, pp. 69)) and set $\Sigma_k := Q^T \operatorname{diag}(\lambda_{k1}^2, \dots, \lambda_{kD}^2) Q$. Finally, the mean vectors are scaled as to fit the predefined separation parameter c_θ.

We generate 30 GMMs for each of the following combination of parameters: $K = 20$, $D \in \{3, 10\}$, separation parameter $c_\theta \in \{0.5,\ 1,\ 2\}$, weight parameter $c_w \in \{0.1, 1\}$, and different combinations of size and eccentricity (i.e., equal size and $e_k = 10$, equal size and $e_k \in [1, 10]$,



(a) $c_\theta = 0.5$ (b) $c_\theta = 1$ (c) $c_\theta = 2$

(d) $c_\theta = 0.5$ (e) $c_\theta = 1$ (f) $c_\theta = 2$

Figure 16.3: Examples for different separation parameters. Each figure shows an orthogonal projection of a data set to a randomly chosen plane. Data sets in (16.3a)–(16.3c) are 3-dimensional. (16.3d)–(16.3f) are 10-dimensional.

different size and $e_k = 1$, different size and $e_k \in [1, 10]$). Then, for each of the resulting GMMs θ, we generate data sets as follows: We draw $N \in \{1\,000,\ 5000\}$ points according to θ. If the data set shall contain noise points, then we draw only $90\% \cdot N$ points according to the GMM, construct a bounding box, elongate its side lengths by a factor 1.2, draw $10\% \cdot N$ noise points uniformly at random from the resized box, and add these points to the $90\% \cdot N$ that we have drawn in the beginning.

Evaluation Method. We consider the initial solutions produced by the initialization (including the optional intermediate algorithm) and the final solutions obtained by running the EM algorithm afterwards. For each data set, we compute the average log-likelihood of the initial and final solution, respectively. Based on these averages, we create rankings of the algorithms (per data set). Then we compute the average rank (and standard deviation of the rank) of each algorithm over all data sets matching certain properties.

We point out that averaging the (average) log-likelihood values over different data sets is not meaningful, since the optimal log-likelihoods may deviate significantly (per data set).

Results. In the following, we focus solely on those methods that are not constantly outperformed by the other methods in all experiments. Therefore, we omit the results of many methods that do not use an intermediate algorithm. Hence, we also recommend to use an intermediate algorithm before applying the EM algorithm.

With respect to the data sets that do not contain uniform noise points, there is no method that constantly outperforms all others. Nonetheless, it is always one of our adaptive methods or the $\texttt{G}_{km}$ initialization that performs best.

The results depicted in Table 16.1 and Table 16.2 suggest that, regardless of the weights, the performance is determined by the separation. Furthermore, they show that a good initial solution does not imply a good final solution. Given overlap ($c_\theta = 0.5$) or moderate separation ($c_\theta = 1$), $\texttt{SG}_{cem}(s = 1)$ and $\texttt{Ad}_{cem}(\alpha = 1)$ work best, even though their initial solutions have

low average ranks compared to $\texttt{KM++}_{km}$. Given higher separation ($c_\theta = 2$), we expect it to be easier to identify clusters and that skewed covariance matrices do not matter much if means are assigned properly in the first place. Indeed, the simple $\texttt{G}_{km}$ and KG do the trick.

Table 16.3 shows that $\texttt{Ad}_{cem}(\alpha = 1)$ works well for elliptical data, while $\texttt{G}_{km}$ should be chosen for spherical data. Recall that there are no noise points yet. We expect that the performance of $\texttt{G}_{km}$ degenerates in the presence of noise since it is prone to choose outliers. Overall, given data sets without noise, $\texttt{Ad}_{cem}(\alpha = 1)$ performs best.

When introducing noise, our adaptive methods are still among the best methods, while the performance of some other methods degenerates significantly. Table 16.4 and Table 16.5 show that $\texttt{SG}_{cem}$ and $\texttt{Ad}_{cem}$ still work well for $c_w \leq 1$ and, in contrast to data without noise, also for separated instances ($c_w = 2$). KG and $\texttt{G}_{km}$ are now among the methods with the lowest average rank. This is not a surprise, since our noisy data sets contain outliers. From the results depicted in Table 16.6 one can draw the same conclusion. That is, KG and $\texttt{G}_{km}$ cannot handle these noisy data sets. Here, $\texttt{Ad}_{cem}(\alpha = 1)$ outperforms the others.

We expect that if the dimension is low or the sample size is large enough, it is generally easier to identify clusters (cf. Section 14.1.3). Indeed the results differ significantly from our previous results. For data sets with $D = 3$ and $|X| = 1000$, Table 16.7 shows that the $\texttt{KM++}_{km}$ method works well even in the presence of noise. However, if we are given noise *and* small separation, the simple $\texttt{Unif}_{km}$ does well. We also increased the sample size to $|X| = 5000$ *and* the dimension to $D = 10$, expecting that the higher sample size can make up for the higher dimension (results available at (Bujna, 2016)). Indeed, for data sets without noise, where clusters can presumably be identified easier, $\texttt{KM++}_{km}$ suffices. However, given noise or too small a separation, our $\texttt{Ad}_{cem}$ methods and the simple $\texttt{Unif}_{km}$ work better.

16.5.3 Results: Real-World Data Sets

We use five publicly available data sets: The *Covertype* data set contains $|X| = 581012$ points, which consist of $D = 10$ real-valued features (Asuncion, 2007). We consider two *Aloi* data sets with $|X| = 110250$ points and dimension $D \in \{27, 64\}$. Both data sets are based on color histograms in HSV color space (Kriegel et al., 2011) from data provided by the ELKI project (Achtert et al., 2012) and the Amsterdam Library of Object Images (Geusebroek, Burghouts, and Smeulders, 2005). The *Cities* data set contains $|X| = 135082$ 2-dimensional points. It is a projection of the coordinates of cities with a population of at least 1000 from the GeoNames geographical database (`http://www.geonames.org/`). The *Spambase* data set contains $|X| = 4601$ points, which consist of $D = 10$ real-valued features (Asuncion, 2007).

The results are depicted in Figure 16.4: For *Aloi* ($D = 27$) and *Spambase* ($K = 3$), $\texttt{SG}_{cem}(s = 1)$ is considerably better than the other methods. For *Cities* and *Spambase* ($K = 10$), $\texttt{SG}(s = 1)$ does better (without running the CEM algorithm). For *Aloi* ($D = 64$) and the *Covertype*, $\texttt{Ad}_{cem}(\alpha = 1)$ works better than the others.

16.6 Conclusion and Future Work

If you need a fast and simple method, then we suggest using one of the following methods: Try the K-means++ initialization followed by Means2GMM and the K-means algorithm. Besides, we recommend testing our new methods Ad and SG followed by the spherical CEM algorithm, especially if your data is presumably noisy. Last but not least, whatever you prefer, we suggest applying intermediate steps of the spherical CEM or K-means algorithm.

For the K-means++ algorithm and the Gonzalez algorithm there are provable guarantees. We hope our results are a starting point for a theoretical analysis that will transfer these results to the MLE problem for GMMs.

Table 16.1: Average ranks (± std.dev.) for generated data with $K = 20$, $|X| = 1\,000$, $D = 10$, **different weights**, and **without noise.**

	separation $c_\theta = 0.5$		**separation** $c_\theta = 1$		**separation** $c_\theta = 2$	
	initial	final	initial	final	initial	final
$\texttt{SG}(s = \frac{1}{10})$	7.53±1.08	3.58±1.93	7.29±0.86	5.08±2.95	7.14±0.57	7.28±2.31
$\texttt{SG}(s = 1)$	8.00±1.52	3.26±2.27	8.77±0.68	5.53±3.42	8.72±0.53	7.44±2.60
$\texttt{KG}(s = 1)$	10.00±0.00	9.75±0.72	10.00±0.00	7.68±2.52	10.00±0.00	2.38±1.34
$\texttt{Unif}_{km}$	1.56±0.74	8.39±1.15	2.19±0.61	7.46±1.88	2.98±0.13	7.08±2.14
$\texttt{G}_{km}$	3.34±1.33	6.03±2.32	2.38±0.86	5.16±2.79	1.23±0.46	1.99±1.31
$\texttt{KM++}_{km}$	1.85±0.64	7.87±1.14	1.43±0.64	6.22±2.60	1.78±0.41	3.75±2.25
$\texttt{SG}(s = \frac{1}{10})_{cem}$	6.15±0.60	3.95±1.24	6.30±0.68	4.71±2.54	6.10±0.40	6.65±2.13
$\texttt{SG}(s = 1)_{cem}$	6.47±1.51	3.20±2.48	7.32±1.26	5.12±3.31	8.03±0.61	7.80±2.61
$\texttt{Ad}(\alpha = 1)_{cem}$	5.16±1.84	4.31±1.77	4.59±0.64	3.88±1.75	4.47±0.50	5.06±1.40
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	4.93±2.11	4.67±1.93	4.72±0.80	4.15±1.77	4.53±0.50	5.58±1.71

Table 16.2: Average ranks (± std.dev.) for generated data with $K = 20$, $|X| = 1\,000$, dimension $D = 10$, **equal weights**, and **without noise**.

	separation $c_\theta = 0.5$		**separation** $c_\theta = 1$		**separation** $c_\theta = 2$	
	initial	final	initial	final	initial	final
$\texttt{SG}(s = \frac{1}{10})$	7.58±0.98	3.98±1.87	7.36±0.73	5.02±2.97	7.08±0.41	7.35±2.24
$\texttt{SG}(s = 1)$	8.11±1.53	3.62±2.58	8.67±0.85	5.67±3.20	8.79±0.43	7.77±2.59
$\texttt{KG}(s = 1)$	10.00±0.00	9.54±0.96	10.00±0.00	7.97±2.40	10.00±0.00	2.38±1.23
$\texttt{Unif}_{km}$	1.44±0.70	8.36±1.25	2.14±0.61	7.28±1.83	2.98±0.16	7.00±1.98
$\texttt{G}_{km}$	3.39±1.34	6.12±2.17	2.53±0.83	6.04±2.76	1.27±0.50	1.82±1.08
$\texttt{KM++}_{km}$	1.91±0.55	7.82±1.30	1.35±0.56	5.89±2.77	1.75±0.43	3.77±2.50
$\texttt{SG}(s = \frac{1}{10})_{cem}$	6.18±0.62	3.58±1.31	6.44±0.87	4.26±2.43	6.02±0.13	6.55±1.95
$\texttt{SG}(s = 1)_{cem}$	6.49±1.44	3.17±2.73	7.35±1.13	5.30±3.23	8.12±0.45	8.04±2.37
$\texttt{Ad}(\alpha = 1)_{cem}$	5.03±1.75	4.12±1.79	4.49±0.64	3.48±1.61	4.42±0.50	4.90±1.35
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	4.87±1.99	4.69±1.91	4.67±0.65	4.09±1.73	4.58±0.50	5.42±1.38

Table 16.3: Average ranks (± std.dev.) for generated data with $K = 20$, $|X| = 1\,000$, dimension $D = 10$, and **without noise**. Only final solutions.

	equal weights			**different weights**		
	spherical	**elliptical**	**both**	**spherical**	**elliptical**	**both**
$\texttt{SG}(s = \frac{1}{10})$	6.13±2.63	5.22±2.80	5.45±2.78	6.03±2.65	5.07±2.90	5.31±2.87
$\texttt{SG}(s = 1)$	6.64±2.99	5.37±3.30	5.69±3.27	6.04±3.22	5.20±3.28	5.41±3.28
$\texttt{KG}(s = 1)$	6.78±3.19	6.58±3.59	6.63±3.49	6.81±3.20	6.53±3.65	6.60±3.54
$\texttt{Unif}_{km}$	7.67±1.48	7.50±1.91	7.54±1.81	7.64±1.64	7.64±1.92	7.64±1.85
$\texttt{G}_{km}$	3.03±2.26	5.20±2.92	4.66±2.92	2.83±2.33	4.91±2.78	4.39±2.82
$\texttt{KM++}_{km}$	5.44±2.60	5.96±2.87	5.83±2.81	5.57±2.66	6.07±2.69	5.95±2.69
$\texttt{SG}(s = \frac{1}{10})_{cem}$	4.77±2.61	4.81±2.23	4.80±2.33	5.68±2.19	4.91±2.35	5.10±2.33
$\texttt{SG}(s = 1)_{cem}$	6.53±3.16	5.16±3.46	5.51±3.43	6.07±3.30	5.14±3.40	5.37±3.39
$\texttt{Ad}(\alpha = 1)_{cem}$	3.62±1.61	4.34±1.69	4.16±1.69	4.02±1.82	4.55±1.66	4.42±1.71
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	4.38±1.78	4.86±1.75	4.74±1.77	4.30±1.86	4.97±1.88	4.80±1.89

Table 16.4: Average ranks (± std.dev.) for generated data with $K = 20$, $|X| = 1\,000$, dimension $D = 10$, **different weights**, and **10% noise**.

	separation $c_\theta = 0.5$		**separation** $c_\theta = 1$		**separation** $c_\theta = 2$	
	initial	final	initial	final	initial	final
$\texttt{SG}(s = \frac{1}{10})$	8.41±0.68	3.40±1.75	8.22±0.64	4.44±2.45	8.06±0.68	5.46±2.42
$\texttt{SG}(s = 1)$	8.25±0.98	3.46±2.60	8.67±0.47	4.13±3.02	8.74±0.44	5.93±2.87
$\texttt{KG}(s = 1)$	10.00±0.00	9.95±0.22	10.00±0.00	9.72±0.76	10.00±0.00	9.02±1.49
$\texttt{Unif}_{km}$	1.98±0.89	8.65±1.03	1.05±0.25	7.89±1.71	1.19±0.49	7.34±1.87
$\texttt{G}_{km}$	4.29±1.29	5.31±1.76	4.17±0.98	6.59±1.43	3.97±0.96	7.16±1.38
$\texttt{KM++}_{km}$	3.23±0.98	6.45±1.40	2.27±0.60	6.83±1.62	2.17±0.60	6.69±1.69
$\texttt{SG}(s = \frac{1}{10})_{cem}$	6.06±0.55	4.26±1.36	6.04±0.20	3.90±0.90	6.01±0.091	3.65±1.27
$\texttt{SG}(s = 1)_{cem}$	6.31±1.43	3.49±2.64	7.08±0.39	3.80±2.89	7.19±0.42	5.00±3.01
$\texttt{Ad}(\alpha = 1)_{cem}$	3.64±1.88	4.61±2.26	4.05±0.90	3.57±2.03	3.76±1.26	2.06±1.39
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	2.83±2.26	5.42±2.70	3.46±0.89	4.12±2.38	3.92±0.78	2.69±1.45

Table 16.5: Average ranks (± std.dev.) for generated data sets with $K = 20$, $|X| = 1\,000$, dimension $D = 10$, **equal weights**, and **10% noise**.

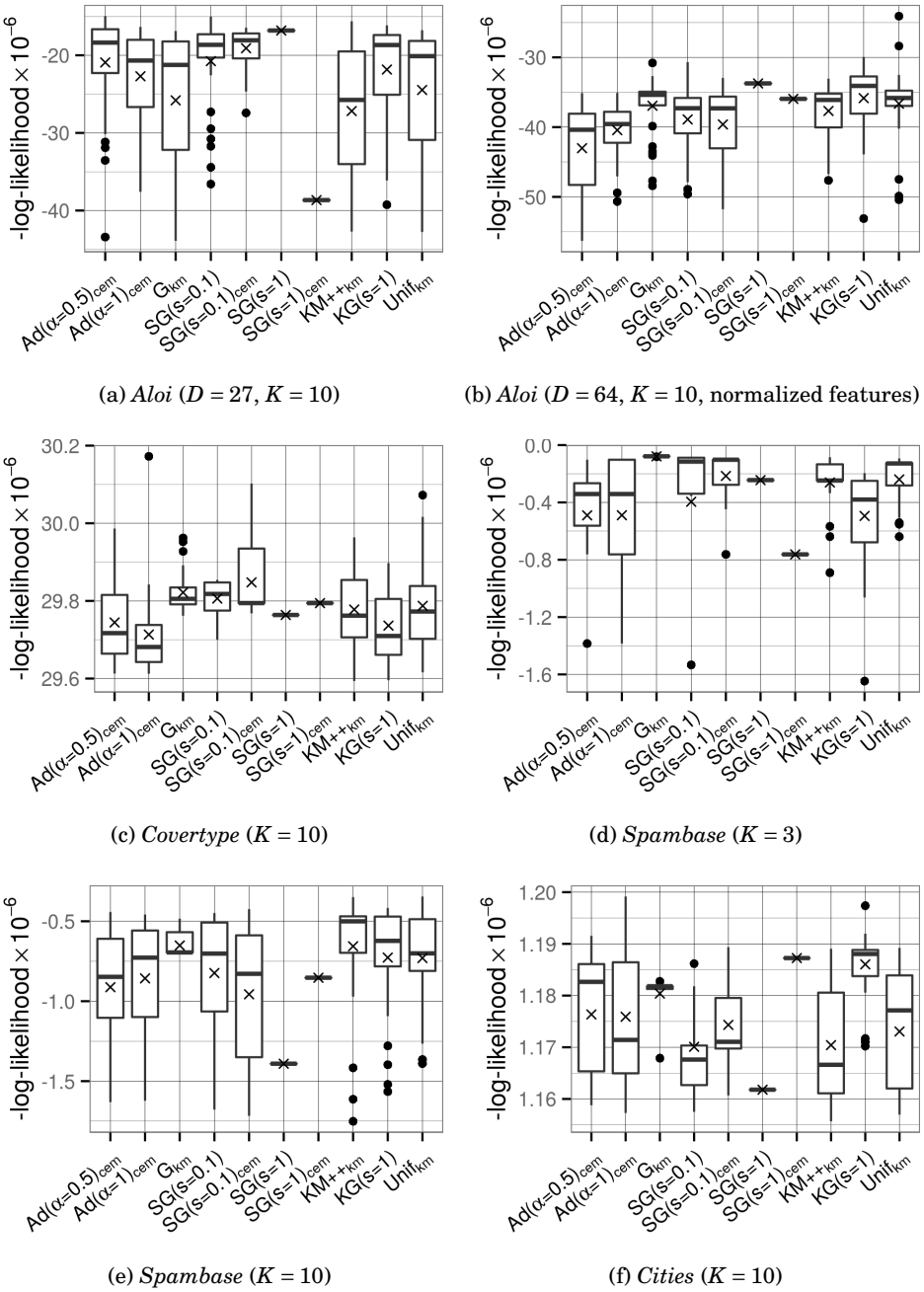
	separation $c_\theta = 0.5$		**separation** $c_\theta = 1$		**separation** $c_\theta = 2$	
	initial	final	initial	final	initial	final
$\texttt{SG}(s = \frac{1}{10})$	8.57±0.62	3.38±1.92	8.18±0.65	4.17±2.28	7.94±0.77	5.02±2.44
$\texttt{SG}(s = 1)$	8.05±1.08	3.19±2.23	8.68±0.47	3.77±2.68	8.73±0.44	5.47±2.96
$\texttt{KG}(s = 1)$	10.00±0.00	9.93±0.35	10.00±0.00	9.62±0.87	10.00±0.00	7.85±2.22
$\texttt{Unif}_{km}$	1.92±0.87	8.83±0.77	1.02±0.13	8.39±1.22	1.11±0.31	8.18±1.63
$\texttt{G}_{km}$	4.47±0.99	5.53±1.75	4.01±1.01	6.74±1.51	3.53±0.83	7.18±1.28
$\texttt{KM++}_{km}$	3.20±1.07	6.66±1.22	2.08±0.31	7.04±1.46	1.93±0.37	7.66±1.56
$\texttt{SG}(s = \frac{1}{10})_{cem}$	6.08±0.53	4.47±1.31	6.03±0.18	3.92±0.97	6.00±0.00	3.64±1.11
$\texttt{SG}(s = 1)_{cem}$	6.20±1.43	3.14±2.43	7.10±0.40	3.88±2.75	7.33±0.47	5.13±2.76
$\texttt{Ad}(\alpha = 1)_{cem}$	3.62±1.85	4.42±2.35	4.20±0.79	3.54±2.11	4.15±0.82	2.27±1.59
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	2.89±2.45	5.45±2.47	3.69±0.74	3.92±2.29	4.28±0.66	2.60±1.51

Table 16.6: Average ranks (± std.dev.) for generated data sets with $K = 20$, $|X| = 1\,000$, dimension $D = 10$, and **10% noise**. Only final solutions.

	equal weights			**different weights**		
	spherical	**elliptical**	**both**	**spherical**	**elliptical**	**both**
$\texttt{SG}(s = \frac{1}{10})$	5.38±2.26	3.79±2.20	4.19±2.32	5.53±2.30	4.07±2.29	4.43±2.38
$\texttt{SG}(s = 1)$	4.98±2.79	3.87±2.76	4.14±2.81	5.53±3.14	4.17±2.90	4.51±3.02
$\texttt{KG}(s = 1)$	8.61±2.03	9.31±1.49	9.13±1.66	9.32±1.28	9.64±0.95	9.56±1.05
$\texttt{Unif}_{km}$	8.36±1.36	8.50±1.26	8.47±1.28	7.63±1.72	8.07±1.63	7.96±1.66
$\texttt{G}_{km}$	6.70±1.69	6.41±1.67	6.49±1.68	6.42±1.87	6.33±1.66	6.35±1.71
$\texttt{KM++}_{km}$	7.09±1.71	7.13±1.39	7.12±1.48	6.46±1.71	6.72±1.53	6.66±1.58
$\texttt{SG}(s = \frac{1}{10})_{cem}$	4.00±1.45	4.01±1.09	4.01±1.18	4.03±1.34	3.90±1.17	3.94±1.22
$\texttt{SG}(s = 1)_{cem}$	5.04±2.79	3.72±2.69	4.05±2.77	4.88±2.98	3.84±2.86	4.10±2.92
$\texttt{Ad}(\alpha = 1)_{cem}$	2.01±1.29	3.88±2.27	3.41±2.22	2.31±1.57	3.78±2.25	3.41±2.19
$\texttt{Ad}(\alpha = \frac{1}{2})_{cem}$	2.83±1.72	4.37±2.50	3.99±2.42	2.88±1.78	4.48±2.57	4.08±2.50

Table 16.7: Average ranks (± std.dev.) for generated data ($K = 20$, $|X| = 1000$, $\mathbf{D} = \mathbf{3}$).

	without noise		**noisy**	
	initial	final	initial	final
$\texttt{SG}\left(s = \frac{1}{10}\right)$	7.31±0.63	7.94±1.39	8.05±0.70	7.93±1.31
$\texttt{SG}(s = 1)$	8.90±0.39	8.56±1.98	8.72±0.45	7.99±2.19
$\texttt{KG}(s = 1)$	9.94±0.42	3.28±1.98	10.00±0.00	8.09±1.46
$\texttt{Unif}_{km}$	2.82±0.58	4.63±1.40	3.38±0.94	2.76±1.38
$\texttt{G}_{km}$	1.93±1.04	2.80±1.97	4.43±1.59	6.01±1.59
$\texttt{KM++}_{km}$	1.51±0.60	1.99±1.21	2.96±1.12	2.35±1.57
$\texttt{SG}\left(s = \frac{1}{10}\right)_{cem}$	6.10±0.42	7.46±1.14	5.75±0.82	6.07±1.16
$\texttt{SG}(s = 1)_{cem}$	7.65±0.94	8.83±1.74	7.04±0.86	8.16±2.10
$\texttt{Ad}(\alpha = 1)_{cem}$	4.35±0.83	4.66±1.48	2.54±1.43	3.02±1.13
$\texttt{Ad}\left(\alpha = \frac{1}{2}\right)_{cem}$	4.49±0.74	4.85±1.51	2.12±1.36	2.62±1.28



(a) *Aloi* ($D = 27$, $K = 10$)

(b) *Aloi* ($D = 64$, $K = 10$, normalized features)

(c) *Covertype* ($K = 10$)

(d) *Spambase* ($K = 3$)

(e) *Spambase* ($K = 10$)

(f) *Cities* ($K = 10$)

Figure 16.4: Results for the real-world data sets depicted as boxplots (final solutions only).

"Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise."

John Tukey[1]

Chapter 17

On the Soft K-Means Problem

In this chapter, we deal with a special case of the maximum likelihood estimation (MLE) problem for Gaussian mixture models (GMMs): We predefine a number of components $K \in \mathbb{N}$, weights $(\omega_k)_{k\in[K]} \in \Delta_{K-1}$, and a spherical covariance matrix $\frac{2}{\beta} I_D$. Then, we consider the MLE problem with respect to the restricted class of GMMs $\Theta_{K,\omega,\beta}$ which contains all GMMs with K components where each component has the corresponding predefined weight ω_k and the predefined covariance $\frac{2}{\beta} I_D$. That is, given a set of observations X, we want to find a GMM θ from the restricted class of GMMs $\Theta_{\omega,\beta}$ with maximum likelihood. As the only degree of freedom left are the mean vectors, we refer to this problem as the soft K-means problem. To the best of our knowledge, there is no algorithm for this problem with an approximation guarantee. In this chapter, we derive a clustering-centric variant of the soft K-means problem and present an approach towards an approximation algorithm.

Overview. In Section 17.1, we give an overview of work related to the soft K-means problem and, in particular, related to a theoretical analysis thereof. In Section 17.2, we briefly state the main contributions of this chapter. In Section 17.3, we introduce the soft K-means problem formally. In Section 17.4, we derive a clustering-centric variant of the problem. Finally, in Section 17.5, we present first steps towards an analysis of this variant.

Publication. In this chapter, we present unpublished ongoing work.

17.1 Related Work

The work of Jin et al. (2016) deals with the a variant of the soft K-means problem: They consider the population (infinite-sample) likelihood function of Gaussian mixture models (with $K \geq 3$ components) instead of the likelihood with respect to a finite set of observations. First, they show that this function has poor local maxima even in the special case of equally-weighted mixtures of well-separated and spherical Gaussians. Second, they analyse the probability that an EM algorithm converges to poor critical points if it is initialized randomly.

Feldman et al. (2011) and Lucic et al. (2017) consider coreset constructions for the MLE problem with respect to GMMs. They consider an surrogate cost function which differs from the likelihood function on the normalization terms of the Gaussian distributions. Lucic et al. (2017) shows that, for each $\epsilon \in (0,1)$, their construction yields a set C_ϵ such that $\sup_{\theta\in\Theta_\lambda} \left|-\log(\mathcal{L}_X(\theta)) + \log(\mathcal{L}_{C_\epsilon}(\theta)\right| \to 0$ as $\epsilon \to 0$, where Θ_λ denotes the set of all GMMs with K components and where each covariance matrix takes eigenvalues in some fixed interval $[\lambda, 1/\lambda]$ for some fixed $\lambda \in [0,1]$.

[1] Source: The future of data analysis. Annals of Mathematical Statistics 33 (1), 1962 (p. 13)

Moreover, the line of work of Dasgupta and Schulman (2007), Kalai et al. (2012), and Hardt and Price (2015) is related to the topic of this chapter. They deal with the following problem: Based on samples that have been drawn according to an unknown GMM θ_{gt}, the goal is to estimate the density function of the underlying GMM θ_{gt}. They investigate the sample complexity and computational complexity of this problem. That is, they investigate how many samples (drawn according to θ_{gt}) are needed to estimate θ_{gt} up to a certain precision and how much runtime is necessary to achieve this. However, to the best of our knowledge, this line of work does not result in any practical algorithms. Moreover, it is questionable whether this analysis can help to explain the quality of some practical clustering algorithms based on GMMs. For instance, a combination of the K-means++ algorithm by Arthur and Vassilvitskii (2007) and the EM algorithm for GMMs, which we explain in detail in Chapter 16, works well in practice. Clearly, an estimate computed by the EM algorithm could be used as an estimate for the underlying GMM θ_{gt}. However, the EM algorithm has been designed for the MLE problem. Therefore, we follow the MLE approach.

Besides that, note that the EM algorithm for the MLE problem that we consider here is also known as the soft K-means algorithm (Mackay, 2003, p. 289). There is no guarantee on the quality of the solution produced by the soft K-means algorithm. For a more details regarding the soft K-means algorithm, we refer to (Mackay, 2003, p. 289).

17.2 Contribution

First, we propose a clustering-centric version of the soft K-means problem where an optimal solution is described by a soft clustering. On the one hand, we want to simplify the analysis of the resulting problem. On the other hand, we want to maintain the core of the soft K-means problem. We claim that our version of the problem explicitly models goals that are implicitly given in the classical soft K-means problem. To underpin this claim, we give a detailed derivation of our problem formulation.

Second, we discuss an approach towards an analysis of this problem: We show that our soft-to-hard-cluster technique is applicable. With the help of this result, we are able to show that a technique for constrained K-means clustering by Bhattacharya et al. (2016) can be used to determine a set of candidate mean vectors that contains means that describes an approximative solution well. Finally, we show that, under certain restrictions, we can determine a soft clustering that complements given approximate mean vectors via linear programming. Though these results do not yield a proper approximation algorithm, they constitute a first step towards an analysis.

17.3 The Weighted Soft K-Means Problem

Consider the MLE problem where the space of solutions is restricted to GMMs whose covariances take the form $\Sigma_k = \frac{2}{\beta} I_D$ for some constant $\beta \in \mathbb{R}_+$ and whose weights $w_1, \dots, w_K$ are equal to some constants $\omega_1, \dots, \omega_K$.

17.3.1 Preliminaries

In the following, we will make use of the notion of the (relative) entropy.

Definition 17.1 (relative entropy)**.** *Let $p, q \in \Delta_{M-1}$ be two categorical distributions over M classes. The entropy of q is given by*

$$\mathcal{H}(q) := -\sum_{z \in \Delta_{M-1}} q(z) \ln(q(z))$$

where we use the convention that $\ln(0) = 0$.

The relative entropy or Kullback-Leibler divergence between p and q is given by

$$\mathrm{KLD}(p\|q) := \sum_{z\in\Delta_{M-1}} p(z)\ln\left(\frac{p(z)}{q(z)}\right)$$

where we use the convention that $\ln(0)=0$, $0\cdot\ln(0/q)=0$, *and* $p\cdot\ln(p/0)=\infty$.

This definition is a special case of the definition that we stated in Lemma 14.18. In the remainder of this section, we only need the following properties of the (relative) entropy. For more information, we refer to (Cover and Thomas, 2006, pp.13).

Lemma 17.2. *For all categorical distributions q,p over the same number of classes, we have* $\mathrm{KLD}(p\|q)\geq 0$, $\mathrm{KLD}(p\|q)=0$ *if and only if* $p=q$, *and* $\mathcal{H}(q)\geq 0$.

Let $Z_1,\dots,Z_K$ be K mutually independent categorical random variables (over the same number of classes) that are distributed according to $q_1,\dots,q_K$, respectively. Then the entropy of their joint distribution $q(Z_1,\dots,Z_K)=\prod_{k=1}^K q_k(Z_k)$ computes to $\mathcal{H}(q)=\sum_{k=1}^K\mathcal{H}(q_k)$.

Proof. A proof can be found in (Cover and Thomas, 2006, pp.42), for instance □

With the help of Lemma 17.2 and Lemma 14.18, we can now rewrite the likelihood of a GMM with constant covariances $\Sigma_k=\frac{2}{\beta}I_D$ and constant weights $w_k=\omega_k$ as follows.

Observation 17.3 (reformulate the likelihood)**.** *Consider a GMM $\theta:=((\omega_k,\mu_k,\frac{2}{\beta}I_D)_{k\in[K]}$ and observations $X=(x_n)_{n\in[N]}$. Let $p_{nk}:=\mathrm{p}(Z_{nk}=1|X,\theta)$ for all latent variables Z_{nk} with $n\in[N]$ and $k\in[K]$. We write $Z_n=(Z_{nk})_{k\in[K]}$ for all $n\in[N]$. Then we can write*

$$\begin{aligned}
&-\ln(\mathcal{L}_X(\theta))\\
&=-\ln\mathrm{p}(X|\theta) && \text{(Definition 14.4)}\\
&=-\mathrm{E}_{Z\sim\mathrm{p}(Z|X,\theta)}[\ln(\mathrm{p}(X,Z|\theta))]-\mathcal{H}(\mathrm{p}(Z|X,\theta))+0 && \text{(Lemma 14.18)}\\
&=-\mathrm{E}_{Z\sim\mathrm{p}(Z|X,\theta)}\left[\ln\left(\prod_{n=1}^N\mathrm{p}(x_n,Z_n|\theta)\right)\right]-\mathcal{H}\left(\prod_{n=1}^N\mathrm{p}(Z_n|x_n,\theta)\right) && \text{(cf. Section 14.1.2)}\\
&=-\sum_{n=1}^N\mathrm{E}_{Z\sim\mathrm{p}(Z|X,\theta)}[\ln(\mathrm{p}(x_n,Z_n|\theta))]-\mathcal{H}\left(\prod_{n=1}^N\mathrm{p}(Z_n|x_n,\theta)\right) && \text{(linearity)}\\
&=-\sum_{n=1}^N\mathrm{E}_{Z\sim\mathrm{p}(Z|X,\theta)}[\ln(\mathrm{p}(x_n,Z_n|\theta))]-\sum_{n=1}^N\mathcal{H}(\mathrm{p}(Z_n|x_n,\theta)) && \text{(Lemma 17.2)}\\
&=-\sum_{n=1}^N\sum_{k=1}^K p_{nk}\ln(\mathrm{p}(x_n,Z_{nk}=1|\theta))+\sum_{n=1}^N\sum_{k=1}^K p_{nk}\ln(p_{nk}) && \text{(total probability)}\\
&=\sum_{n=1}^N\sum_{k=1}^K -p_{nk}\ln\left(w_k\mathcal{N}_D\left(x_n\middle|\mu_k,\frac{2}{\beta}I_D\right)\right)+p_{nk}\ln(p_{nk}) && \text{(cf. Section 14.1.2)}\\
&=\sum_{n=1}^N\sum_{k=1}^K -p_{nk}\ln\left(\mathcal{N}_D\left(x_n\middle|\mu_k,\frac{2}{\beta}I_D\right)\right)+p_{nk}\ln\left(\frac{p_{nk}}{w_k}\right)\\
&=\sum_{n=1}^N\sum_{k=1}^K p_{nk}\left(\frac{D}{2}\cdot\ln\left(\frac{1}{2\pi\beta}\right)+\beta\left\|x_n-\mu_k\right\|_2^2\right)+\sum_{n=1}^N\mathrm{KLD}(p_{nk})_k(w_k)_k && \text{(Definition 14.1)}\\
&=\underbrace{\frac{ND}{2}\cdot\ln\left(\frac{1}{2\pi\beta}\right)}_{const}+\sum_{n=1}^N\sum_{k=1}^K p_{nk}\beta\left\|x_n-\mu_k\right\|_2^2+\sum_{n=1}^N\mathrm{KLD}((p_{nk})_k\|(w_k)_k)\ .
\end{aligned}$$

and

$$p_{nk}=p(Z_{nk}=1|X,\theta)=\frac{p(x_n,Z_{nk}=1|\theta)}{p(x_n|\theta)}=\frac{w_k\exp(-\beta\left\|x_n-\mu_k\right\|_2^2)}{\sum_{l=1}^K w_l\exp(-\beta\left\|x_n-\mu_l\right\|_2^2)}\ .$$

17.3.2 Problem Statement

Our observation from the previous section directly leads us to the following formulation:

Problem 17.4 (weighted soft K-means)**.** *We are given observations $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$, $\omega \in \Delta_{K-1}$, and $\beta \in \mathbb{R}_+$. Find K mean vectors $(\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ minimizing*

$$\text{škm}_X^{(\beta,\omega)}((\mu_k)_k) := \beta \cdot \sum_{n=1}^{N}\sum_{k=1}^{K} p_{nk} \left\|x_n - \mu_k\right\|_2^2 + \sum_{n=1}^{N} \mathrm{KLD}\left((p_{nk})_{k\in[K]} \middle\| \omega\right)$$

where

$$p_{nk} = \frac{\omega_k \exp(-\beta \left\|x_n - \mu_k\right\|_2^2)}{\sum_{l=1}^{K} \omega_l \exp(-\beta \left\|x_n - \mu_l\right\|_2^2)} \tag{17.1}$$

for all $n \in [N]$ and $k \in [K]$.

Recall that, strictly speaking, the MLE problem for GMMs that we stated in Problem 14.9 is meaningless because the objective function is unbounded. However, this does not hold for Problem 17.4: Recall that $\mathrm{KLD}(p\|q) \geq 0$ for all categorical distributions q, p over the same number of classes (Lemma 17.2). Hence, the goal of Problem 17.4 is to minimize an objective function that bounded from below.

An EM algorithm can be used to tackle this problem. There is no guarantee on the quality of the computed solution, though.

Remark 17.5 (soft K-means algorithm)**.** *The instantiation of the EM algorithm that is referred to as the soft K-means algorithm (Mackay, 2003, p. 289) is an algorithm for the weighted soft K-means problem with $\omega_k = 1/K$ for all $k \in [K]$.*

17.3.3 Approximation

Optimal solutions are probably hard, if not impossible, to compute. Therefore, we aim to compute an approximate solution that is not much less likely than an optimal solution. This raises the following question: How do we compare the soft K-means cost of two different solutions and what differences can we expect?

Difference of Objective Values. Recall that in the fuzzy K-means problem we compared the *factor* between the fuzzy K-means costs of an optimal solution and an approximate solution. That is, we sought for a solution C with $\phi_X^{(r)}(C) \leq \alpha \cdot \phi_X^{(r)}(C^{opt})$ for some small constant $\alpha \in \mathbb{R}_+$.

However, in the soft K-means problem, we deal with a negative log-likelihood as a cost. As we already explained in Section 14.2, one compares likelihood values in terms of their ratio. This means that one compares *log*-likelihoods in terms of their difference. Hence, our goal is that the difference

$$\text{škm}_X^{(\beta,\omega)}((\mu_k)_k) - \text{škm}_X^{(\beta,\omega)}\left(\left(\mu_k^{opt}\right)_k\right) = \log\left(\Lambda_X\left(\left((\omega_k, \mu_k^{opt}, \frac{2}{\beta} I_D\right)_k, \left((\omega_k, \mu_k, \frac{2}{\beta} I_D\right)_k\right)\right)$$

between the cost of an approximate solution $(\mu_k)_k$ and an optimal solution $(\mu_k^{opt})_k$ is small.

Dependencies. Observe that, in general, the value of the objective function

$$\text{škm}_X^{(\beta,\omega)}((\tilde{\mu}_k)_k) = \sum_{n=1}^{N}\left(\beta \cdot \sum_{k=1}^{K} \tilde{p}_{nk} \left\|x_n - \tilde{\mu}_k\right\|_2^2 + \mathrm{KLD}\left((\tilde{p}_{nk})_{k\in[K]} \middle\| \omega\right)\right)$$

increases with every observation that we add to X, irrespective of the given set of means $(\tilde{\mu}_k)_k$ and the induced soft clustering $(\tilde{p}_{nk})_{n,k}$ (cf. Lemma 17.2). Similarly, the value of the objective function increases (or at least, does not decrease) with every coordinate that we add to each observation in X. Only the first summand depends on the dimension, though.

To sum up, we *cannot* expect an algorithm to compute an approximative solution such that the difference between the cost of the approximative solution and the optimal solution is at most a factor $\log(1+\epsilon) =: \epsilon'$, which is independent of the number of given observations $|X|$ and their dimension D, without adjusting the precision parameter ϵ that we feed to the algorithm. In other words, we *cannot* expect an algorithm to compute an approximative solution such that the (non-log) likelihood ratio between the approximative solution and the optimal solution is at most a factor $(1+\epsilon)$, which does not depend on $|X|$ and D.

Conclusion. We propose the following approximation problem.

Problem 17.6 (approximate weighted soft K-means)**.** *We are given $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$, $\omega \in \Delta_{K-1}$, $\beta \in \mathbb{R}_+$, and $\epsilon \in [0,1]$. Find K mean vectors $(\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ such that*

$$\check{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((\mu_k)_{k\in[K]}) - \check{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((\mu_k^{opt})_{k\in[K]}) \leq \epsilon \cdot |X| D \ ,$$

where $(\mu_k^{opt})_{k\in[K]} \in \operatorname{arg\,min}\left\{\check{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((\tilde{\mu}_k)_{k\in[K]}) \,\middle|\, (\tilde{\mu}_k)_{k\in[K]} \subseteq \mathbb{R}^D\right\}$ is an optimal solution.

17.4 A Clustering-Centric Variant

In this section, we derive a problem that is similar to the soft K-means problem and that is focused on a soft clustering of the given data points.

17.4.1 Motivation

First, we do not know how to tackle Problem 17.6 directly. In particular, we cannot apply the techniques that we used in Part II: There we worked with soft clusterings rather than with mean vectors and used the notion of *induced* means. In Problem 17.6, a solution is solely described via mean vectors, which *induce* some soft clustering.

Second, *every* data set can be described by a GMM with fixed covariances and weights, in the sense that, among all GMMs with fixed covariances and weights, there is *always* some GMM with fixed covariances and weights that describes a given data set best. We can find such a GMM even though the data set might not have been generated by such a GMM. In particular, the soft clusters that are induced by this GMM, might exhibit completely different weights and covariances than those weights and covariances that are given by the GMM itself.

For these two reasons, we want to derive a variant of Problem 17.6 where a soft clustering is used to describe a solution. Roughly speaking, we assume that there is a soft clustering of the given data set such the variances and weights of these clusters take the desired form, and we want to find this soft clustering. Table 17.1 gives an overview of the derivation of the new clustering problem, which we present in the following sections.

17.4.2 A First Clustering-Centric Variant

To derive a clustering-centric variant of Problem 17.4, we first relax the problem and then add constraints.

	solution	property	cost
Problem 17.4	C	GMM with means from C, covariances $2/\beta \cdot I_D$ and weights $(\omega_k)_k$ induces $P = (p_{nk})_{n,k}$ with $\mathbf{w}\left(A_k^{(X,P)}\right) = ?$, $\mathbf{cov}\left(A_k^{(X,P)}\right) = ?$	$\check{\mathrm{skm}}_X^{(\beta,\omega)}(C) = \mathrm{skm}_X^{(\beta,\omega)}(C,P)$
Problem 17.9	P	P describes clusters with $\mathbf{w}\left(A_k^{(X,P)}\right) = ?$, $\mathbf{cov}\left(A_k^{(X,P)}\right) = ?$	$\bar{\mathrm{skm}}_X^{(\beta,\omega)}(P)$
(17.2)	P	P describes describes clusters with $\mathbf{w}\left(A_k^{(X,P)}\right) = \omega_k \lvert X\rvert$, $\mathbf{cov}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} I_D$	$\bar{\mathrm{skm}}_X^{(\beta,\omega)}(P)$
(17.3)	P	P describes clusters with $\mathbf{w}\left(A_k^{(X,P)}\right) = \omega_k \lvert X\rvert$, $\mathbf{var}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} D$	$\bar{\mathrm{skm}}_X^{(\beta,\omega)}(P)$
(17.4)	$((\mu_k)_k, P)$	$((\mu_k)_k, P)$ satisfy $\mathbf{w}\left(A_k^{(X,P)}\right) \approx \omega_k \lvert X\rvert$, $\mathbf{var}\left(A_k^{(X,P)}, \mu_k\right) \approx \frac{2}{\beta} D$ $\left\lVert \mathbf{m}\left(A_k^{(X,P)}\right) - \mu_k \right\rVert_2^2 \leq \epsilon \cdot \frac{2}{\beta} D$	$\mathrm{skm}_X^{(\beta,\omega)}((\mu_k)_k, P)$

Table 17.1: Overview of our Derivation of Problem 17.7.

Straightforward Relaxation. An obvious way to relax Problem 17.4 is to extend the input of the objective function by a soft clustering in a straightforward manner and to ignore the form that this soft clustering should take:

Problem 17.7 (a relaxation)**.** *We are given observations $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$, $\omega \in \Delta_{K-1}$, and $\beta \in \mathbb{R}_+$. Find K mean vectors $(\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ and a soft K-clustering $(p_{nk})_{n\in[N],k\in[K]}$ minimizing*

$$\mathrm{skm}_X^{(\beta,\omega)}((\mu_k)_k, (p_{nk})_{n,k}) := \beta \cdot \sum_{n=1}^{N} \sum_{k=1}^{K} p_{nk} \left\lVert x_n - \mu_k \right\rVert_2^2 + \sum_{n=1}^{N} \mathrm{KLD}\left((p_{nk})_{k\in[K]} \middle\| \omega\right) .$$

This problem is reasonable as, for fixed means $(\mu_k)_{k\in[K]}$, the induced soft K-clustering $(p_{nk})_{n,k}$ takes the desired form (17.1):

Lemma 17.8 (induced soft clustering)**.** *Let $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$, $\omega \in \Delta_{K-1}$, and $\beta \in \mathbb{R}_+$. Fix K means $C = (\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$. Then, the soft clustering $P = (p_{nk})_{n,k}$ that minimizes the cost $\mathrm{skm}_X^{(\beta,\omega)}(C,P)$ satisfies*

$$p_{nk} = \frac{\omega_k \exp(-\beta \left\lVert x_n - \mu_k \right\rVert_2^2)}{\sum_{l=1}^{K} \omega_l \exp(-\beta \left\lVert x_n - \mu_l \right\rVert_2^2)}$$

for all $n \in [N]$ and $k \in [K]$.

Proof. Define $\psi((p_{nk})_{n,k}) := \sum_{k=1}^{K} \sum_{n=1}^{N} \beta p_{nk} \left\lVert x_n - \mu_k \right\rVert_2^2 + p_{nk} \ln\left(\frac{p_{nk}}{w_k}\right) + \sum_{n=1}^{N} \lambda_n \left(\sum_{k=1}^{K} p_{nk} - 1\right)$, where λ_n denote Lagrange multipliers (which ensure that $\sum_{k=1}^{K} p_{nk} = 1$ for each $n \in [N]$). Setting the first derivative of ψ in the direction of p_{nk} to zero, gives

$$\frac{\partial}{\partial p_{nk}} \psi = \left\lVert x_n - \mu_k \right\rVert_2^2 + \ln\left(\frac{p_{nk}}{w_k}\right) + 1 + \lambda_n = 0 .$$

This implies that

$$p_{nk} = w_k \exp\left(-\left\|x_n - \mu_k\right\|_2^2\right) \exp(-1-\lambda_n) \ .$$

With the constraint $\sum_{k=1}^K p_{nk} = 1$, we can conclude that

$$\exp(-1-\lambda_n) = \frac{1}{\sum_{k=1}^K w_k \exp\left(-\left\|x_n - \mu_k\right\|_2^2\right)} \ .$$

A combination of these equalities yields the claim. □

Hence, the soft K-means algorithm (i.e., the EM algorithm for Gaussian mixtures from $\Theta_{K,\omega,\beta}$) is also an alternating optimization algorithm for Problem 17.9. In other words, similarly to Lloyd's algorithm for the K-means problem, the soft K-means algorithm makes use of a relaxation of the objective function that it tries to minimize.

Add Constraints on the Mean Vectors. To obtain a clustering-centric variant of Problem 17.4, we could just fix the mean vectors to the mean vectors induced by the soft clustering (cf. Lemma 2.20):

Problem 17.9 (first draft)**.** *We are given observations $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$, $\omega \in \Delta_{K-1}$, and $\beta \in \mathbb{R}_+$. Find a soft K-clustering $(p_{nk})_{n\in[N],k\in[K]}$ minimizing*

$$\bar{\mathrm{skm}}_X^{(\beta,\omega)}((p_{nk})_{n,k}) := \beta \cdot \sum_{n=1}^N \sum_{k=1}^K p_{nk} \left\|x_n - \mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2 + \sum_{n=1}^N \mathrm{KLD}\left((p_{nk})_{k\in[K]} \| \omega\right) \ .$$

However, there there is no guarantee that the soft clusters that are given by an optimal solution $(p_{nk})_{n,k}$ to Problem 17.9 take a form that is similar to the soft clusters described by the GMM $((\omega_k, \mu_k^{opt}, \frac{2}{\beta} I_D))_{k\in[K]}$ that we actually search for. That is, the variances and weights of the soft clusters defined by $(p_{nk})_{n,k}$ do not necessarily coincide with $2/\beta$ and $\omega_1, \ldots, \omega_K$, respectively.

Add Constraints on the Weights and Covariances. To sum up, in order to obtain a reasonable clustering-centric version of Problem 17.4, we have to take into account the assumptions on the covariances and weights explicitly.

We propose to consider Problem 17.9 subject to:

$$\forall k \in [K] \ : \ \mathbf{cov}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} I_D \quad \text{and} \quad \mathbf{w}\left(A_k^{(X,P)}\right) = \omega_k \cdot |X| \ . \tag{17.2}$$

The motivation for these constraints is the following: Recall from Lemma 14.16 that the expected covariance of observations that have been sampled according to a Gaussian $\mathcal{N}(\mu_k, 2/\beta \cdot I_D)$ converges to $\frac{2}{\beta} I_D$ when the number of samples diverges to infinity. Besides that, from Section 14.1.2, it is obvious that the expected number of observations that have been drawn according to the k-th component of a GMM with weights $(\omega_k)_{k\in[K]}$ computes to ω_k. We want the covariance and the weight of each soft cluster described by P to compute to these expected (limit) values.

The Non-Existence of an (Optimal) Solution. In our constraints from (17.2) we demand that the covariances and weights compute to certain values *exactly*. Does such a solution always exist? Formally, we are given a non-linear optimization problem with several side constraints. A solution to such a problem is not always guaranteed to exist. Nevertheless, in principle, this is no problem: Assume that we have an algorithm that returns

the best solution matching certain constraints if a solution that matches the constraints exists. Given an arbitrary set of observations, we do not know if a solution that matches the constraints exists. However, we can easily figure that out by applying the algorithm (if such an algorithm exists). In the following, we focus on the problem under the assumption that an (optimal) solution satisfying (17.2) exists.

17.4.3 A Relaxation

Considering Problem 17.9 under the constraints from (17.2) seems to be reasonable. Yet, we have no idea how to deal with the very specific constraint on the covariances. It demands that each of the $D(D+1)/2$ different entries of the covariance matrix matches a certain value. We presume that this constraint is far to strict to make sense. Therefore, our next question is: How can we relax the constraint on the covariance?

Variance instead of Covariance. We relax the problem further by constraining the *variances* of the clusters instead of the *covariances*. Formally, instead of the additional constraint from (17.3), we use the additional constraint

$$\forall k \in [K] \,:\, \mathbf{var}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} \cdot D \quad \text{and} \quad \mathbf{w}\left(A_k^{(X,P)}\right) = \omega_k \cdot |X| \ . \tag{17.3}$$

Observe that we demand that the variances compute to $2/\beta \cdot D$ due to the relation between the covariance and variance that we already explained in Lemma 2.17.

Cost of a Solution with Matching Variance. The constraint from (17.3) not only relaxes the problem, it also simplifies our notation: The cost of a solution that matches the constraints from (17.3) takes the following simple form:

Observation 17.10 ("$2|X|D$"). *Assume that* $\mathbf{var}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} \cdot D$ *for all* $k \in [K]$. *Then,*

$$\sum_{k=1}^{K} \beta \cdot \mathbf{d}\left(A_k^{(X,P)}\right) = \sum_{k=1}^{K} \left(2D \frac{\mathbf{w}\left(A_k^{(X,P)}\right)}{\mathbf{d}\left(A_k^{(X,P)}\right)} \right) \cdot \mathbf{d}\left(A_k^{(X,P)}\right) = 2D \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right) = 2|X|D \ .$$

Hence,

$$\bar{\mathrm{skm}}_X^{(\beta,\omega)}((p_{nk})_{n,k}) = 2|X|D + \sum_{n=1}^{N} \mathrm{KLD}\left((p_{nk})_{k\in[K]} \| \omega\right) \ .$$

17.4.4 A Relaxed Clustering-Centric Approximation Problem

So far, we only derived a clustering-centric variant of Problem 17.4 and relaxed this variant. That is, we introduced a new optimization problem by defining a new notion of an optimal solution, which is given by a soft clustering. To derive a clustering-centric variant of the approximation problem from Problem 17.6, we still need to specify how we describe approximative solutions.

Description of an Approximative Solution. Clearly, one could describe an approximate solution by a soft clustering as well. Again, we do not know how to determine such a solution. Therefore, we take a different approach and allow more degrees of freedom: We

propose to search for a soft K-clustering P of X *and* mean vectors $(\mu_k)_{k\in[K]}$ satisfying

$$\begin{aligned}\forall k \in [K]: \quad & \mathbf{var}\left(A_k^{(X,P)}, \mu_k\right) \in [1, 1+\epsilon] \cdot \frac{2}{\beta} D \\ \wedge \ & \left\| \mu_k - \mathbf{m}\left(A_k^{(X,P)}\right) \right\|_2^2 \leq \epsilon \cdot \frac{2}{\beta} D \\ \wedge \ & \mathbf{w}\left(A_k^{(X,P)}\right) \in [1 \pm \epsilon] \cdot \omega_k \cdot |X| \ .\end{aligned} \tag{17.4}$$

Our additional constraint on the mean vectors is motivated by Lemma 2.20.

Resulting Problem Formulation. The result of all our considerations is the following problem statement.

Problem 17.11 (clustering-centric approximation problem)**.** *We are given observations $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, a number of clusters $K \in \mathbb{N}$, weights $\omega = (\omega_k)_{k\in[K]} \in \Delta_{K-1}$, a parameter $\beta \in \mathbb{R}_+$, and a precision $\epsilon \in [0,1]$.*

Find a soft K-clustering $(p_{nk})_{n,k}$ and means $(\mu_k)_{k\in[K]} \subseteq \mathbb{R}^D$ with the following property:

If there exists a soft K-clustering P_{opt} that minimizes the objective function $\bar{\mathrm{skm}}_X^{(\beta,\omega)}(P_{opt})$ subject to

$$\forall k \in [K] \ : \mathbf{var}\left(A_k^{(X,P_{opt})}\right) = \frac{2}{\beta} \cdot D \quad \textit{and} \quad \mathbf{w}\left(A_k^{(X,P_{opt})}\right) = \omega_k \cdot |X| \ ,$$

then

$$\mathrm{skm}_X^{(\beta,\omega)}\left((\mu_k)_k, (p_{nk})_{n,k}\right) - \bar{\mathrm{skm}}_X^{(\beta,\omega)}\left(P_{opt}\right) \leq \epsilon \cdot |X| D$$

and

$$\begin{aligned}\forall k \in [K]: \quad & \mathbf{var}\left(A_k^{(X,P)}, \mu_k\right) \in [1, 1+\epsilon] \cdot \frac{2}{\beta} D \quad , \\ \wedge \ & \left\| \mu_k - \mathbf{m}\left(A_k^{(X,P)}\right) \right\|_2^2 \leq \epsilon \cdot \frac{2}{\beta} D \ , \quad \textit{and} \\ \wedge \ & \mathbf{w}\left(A_k^{(X,P)}\right) \in [1 \pm \epsilon] \cdot \omega_k \cdot |X| \ .\end{aligned}$$

We describe an optimal solution via a soft clustering. We consider a soft clustering optimal if it minimizes the cost subject to the constraint that the variances and weights of the soft clusters match the desired values (17.3).

Our goal is to describe the data set via a soft clustering P *and* mean vectors $(\mu_k)_{k\in[K]}$. We want this description to be not much less likely than an optimal solution. Moreover, we want the variances of the soft clusters with respect to the mean vectors μ_k, the mean vectors μ_k, and the weights of the soft clusters to approximately match the desired values (17.4). As explained in Section 17.3.3, we want the difference between the cost of this solution and the cost of an optimal solution to be at most $\epsilon \cdot |X| D$ for some small ϵ.

Downsides. Despite our claim that Problem 17.11 is a reasonable way to formalize the goals behind Problem 17.6, there are some downsides that we cannot compensate: Obviously, we lose the notion of a generative model. We only have a constraint on the variances but not on covariances. Another disadvantage might be that, even if we had a constraint on the covariance, the soft clustering (of a good solution to our problem) might still take a significantly different shape than a soft clustering induced by a solution to Problem 17.4. In other words, there is not necessarily a GMM that induces this soft clustering.

17.5 Towards an Analysis

In the following, we present three results regarding Problem 17.11: In Section 17.5.3, we show how our soft-to hard-cluster technique from Chapter 3 can be applied. In Section 17.5.2, we show that an algorithm for constrained K-means clustering by Bhattacharya et al. (2016) can be used to determine mean vectors. To prove this result, we make use of our soft-to-hard-cluster technique. The quality of the computed mean vectors depends on the given weights, though. The best estimates are obtained for uniform weights. In Section 17.5.3, we show how, under various further restrictions, an appropriate soft clustering for a given set of means can be determined via linear optimization.

17.5.1 Applying Our Soft-to-Hard-Cluster Technique

In this section, we take advantage of our formulation of Problem 17.11: It enables us to apply our soft-to-hard-cluster technique from Section 3.6.1.

Consider a soft K-clustering P where each cluster has the same fixed variance $2/\beta$ and not too small a weight. In the following, we show that there exist hard clusters $A_1,\dots,A_K$ of X whose mean vectors $\mathbf{m}(A_1),\dots,\mathbf{m}(A_K)$ are good surrogates for the mean vectors induced by the soft clustering P. In particular, when we exchange these means with the means of the hard clusters, the solution becomes only a little less likely and the variances of the clusters with respect to these means still approximately compute to $2/\beta$.

Corollary 17.12. *Let $X=(x_n)_{n\in[N]}$, $K\in\mathbb{N}$, $\beta\in\mathbb{R}_+$, and $\epsilon\in(0,1]$. Assume there is a soft K-clustering $P=(p_{nk})_{n\in[N],k\in[K]}$ of X where*

$$\forall k\in[K]\colon \mathbf{w}\left(A_k^{(X,P)}\right)\geq\frac{16K}{\epsilon}\quad\text{and}\quad\mathbf{var}\left(A_k^{(X,P)}\right)=\frac{2}{\beta}\cdot D\ . \tag{17.5}$$

Then, there exists a hard clustering $A_1,\dots,A_K$ of X such that

$$\mathrm{skm}_X^{(\beta,\omega)}\left((\mathbf{m}(A_k))_{k\in[K]},(p_{nk})_{n,k}\right)-\bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((p_{nk})_{n,k})\leq\epsilon|X|D$$

and

$$\begin{aligned}\forall k\in[K]\colon \mathbf{var}\left(A_k^{(X,P)},\mathbf{m}(A_k)\right)&\in\left[1,1+\frac{\epsilon}{2}\right]\frac{2}{\beta}\cdot D\ ,\\ \left\|\mathbf{m}\left(A_k^{(X,P)}\right)-\mathbf{m}(A_k)\right\|_2^2&\leq\frac{\epsilon}{2}\cdot\frac{2}{\beta}D\quad\text{and}\\ \mathbf{w}(A_k)&\geq\frac{1}{2}\cdot\mathbf{w}\left(A_k^{(X,P)}\right)\ .\end{aligned}$$

Proof. Let $\dot{\bigcup}_{k=1}^K A_k=X$ be the partition whose existence is guaranteed by Theorem 3.21. That is, we have

$$\mathbf{w}(A_k)\geq\frac{1}{2}\mathbf{w}\left(A_k^{(X,P)}\right)\quad\text{and} \tag{17.6}$$

$$\left\|\mathbf{m}\left(A_k^{(X,P)}\right)-\mathbf{m}(A_k)\right\|_2^2\leq\frac{\epsilon}{2}\cdot\frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)}=\frac{\epsilon}{2}\frac{2}{\beta}D\ , \tag{17.7}$$

where the last equality is due to (17.5).

Now we prove the first part of the claim. Fix some $k \in [K]$ and observe that

$$\begin{aligned}
\mathbf{d}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right) &= \mathbf{d}\left(A_k^{(X,P)}\right) + \mathbf{w}\left(A_k^{(X,P)}\right) \left\|\mathbf{m}\left(A_k^{(X,P)}\right) - \mathbf{m}(A_k)\right\|_2^2 && \text{(Lemma 2.20)} \\
&\leq \mathbf{d}\left(A_k^{(X,P)}\right) + \mathbf{w}\left(A_k^{(X,P)}\right) \frac{\epsilon}{2} \cdot \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} && \text{(Equation (17.7))} \\
&= \left(1 + \frac{\epsilon}{2}\right) \cdot \mathbf{d}\left(A_k^{(X,P)}\right) . && (17.8)
\end{aligned}$$

Due to (17.5) and by Definition 2.16, we have

$$\beta = \frac{2 \cdot D}{\mathbf{var}\left(A_k^{(X,P)}\right)} = 2D \cdot \frac{\mathbf{w}\left(A_k^{(X,P)}\right)}{\mathbf{d}\left(A_k^{(X,P)}\right)} . \tag{17.9}$$

From (17.8) and (17.9) we can conclude that

$$\beta \cdot \mathbf{d}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right) \leq \left(1 + \frac{\epsilon}{2}\right) \cdot 2D \cdot \mathbf{w}\left(A_k^{(X,P)}\right) = (2+\epsilon) \cdot D \cdot \mathbf{w}\left(A_k^{(X,P)}\right) . \tag{17.10}$$

Hence,

$$\begin{aligned}
&\mathrm{skm}_X^{(\beta,\omega)}\left((\mathbf{m}(A_k))_{k\in[K]}, (p_{nk})_{n,k}\right) - \bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((p_{nk})_{n,k}) \\
&= \left(\beta \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right)\right) - 2|X|D && \text{(Observation 17.10)} \\
&\leq (2+\epsilon) \cdot D \cdot \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right) - 2|X|D && \text{(Equation (17.10))} \\
&= \epsilon |X| D . && (X \text{ is an unweighted data set})
\end{aligned}$$

This yields the first part of the claim.

To prove the second part of the claim, observe that

$$\begin{aligned}
\mathbf{var}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right) &= \frac{\mathbf{d}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} && \text{(Definition 2.16)} \\
&= \frac{\mathbf{d}\left(A_k^{(X,P)}\right) + \mathbf{w}\left(A_k^{(X,P)}\right) \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2}{\mathbf{w}\left(A_k^{(X,P)}\right)} && \text{(Lemma 2.20)} \\
&= \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} + \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2 \\
&= \mathbf{var}\left(A_k^{(X,P)}\right) + \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2 . && \text{(Definition 2.16)}
\end{aligned}$$

On the one hand, this shows that $\mathbf{var}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right) \geq \mathbf{var}\left(A_k^{(X,P)}\right)$. On the other hand, observe that

$$\begin{aligned}
\mathbf{var}\left(A_k^{(X,P)}, \mathbf{m}(A_k)\right) &= \mathbf{var}\left(A_k^{(X,P)}\right) + \left\|\mathbf{m}(A_k) - \mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2 \\
&\leq \mathbf{var}\left(A_k^{(X,P)}\right) + \frac{\epsilon}{2} \cdot \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} && \text{(Equation (17.7))} \\
&= \left(1 + \frac{\epsilon}{2}\right) \mathbf{var}\left(A_k^{(X,P)}\right) .
\end{aligned}$$

This yields the second part of the claim. □

17.5.2 Applying an Algorithm for the Constrained K-Means Problem

In the last section, we showed that our soft-to-hard-cluster technique from Section 3.6.1 guarantees that, for each soft clustering whose clusters have the desired variances and not too small a weight, there exist hard clusters whose means are good surrogates for the means induced by the soft clustering. Recall from Section 3.6.3 that these hard clusters exhibit no locality property (for instance, the convex hulls of the hard clusters might overlap). Despite these downsides, these hard clusters form a *hard clustering* of the given observations X. This is different from our application of the soft-to-hard-cluster technique in Chapter 8. There, we were not able to guarantee that there exist appropriate hard clusters that form a hard clustering because we were only given a probabilistic membership matrix, which does not necessarily describe a soft clustering. As a consequence, we had to apply the superset sampling technique, which solely relies on the condition that each hard cluster contains a certain minimum number of points. Now, given the fact that the hard clusters form a hard clustering, we can apply a slightly more elaborate technique.

Bhattacharya et al. (2016) presented an algorithm that aims to identify the means of the hard clusters of an unknown hard clustering:

Theorem 17.13. *Given a data set $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $K \in \mathbb{N}$ and $\epsilon \in [0,1]$, the algorithm from (Bhattacharya et al., 2016) computes a set of candidates $S \subset \left(\mathbb{R}^D\right)^K$ that satisfies the following property:*

Fix an arbitrary hard K-clustering $A_1, \dots, A_K$ of X. Then, with constant probability, there is a candidate $(\mu_k)_{k\in[K]} \in S$ with

$$\forall k \in [K]\colon \mathbf{d}\left(A_k, \mu_k\right) \leq \left(1 + \frac{\epsilon}{2}\right)\mathbf{d}(A_k) + \frac{\epsilon}{2K}\sum_{l=1}^{K}\mathbf{d}(A_l) \tag{17.11}$$

The algorithm's runtime is $|X|D \cdot 2^{\tilde{O}(K/\epsilon)}$ and the size of S is $2^{\tilde{O}(K/\epsilon)}$.

Proof. See (Bhattacharya et al., 2016, p. 16:8) (yes, the page number is "16:8"). □

Their algorithm is based on the same idea as the K-means++ algorithm of Arthur and Vassilvitskii (2007) but does not require the hard clusters to satisfy any locality property. Hence, it is just the right tool for our problem.

In the following, we consider a straightforward application of their result:

Algorithm 28 Applying (Bhattacharya et al., 2016)

Require: $X = (x_n)_{n\in[N]} \subset \mathbb{R}^D$, $\beta \in \mathbb{R}_+$, $K \in \mathbb{N}$, $\epsilon \in (0,1]$

1: Choose
$$\tilde{\epsilon} := \frac{\epsilon}{64 \cdot K}\,.$$
2: Choose the number of copies
$$c := \left\lceil \frac{16K}{\tilde{\epsilon}} \right\rceil\,.$$
3: Construct a data set X_c that, for each $n \in [N]$, contains c copies of the data point (x_n, w_n).
4: Apply the algorithm from Bhattacharya et al. (2016) to X_c, K, and $\tilde{\epsilon}$ to compute a set of candidate solutions $S \subset \left(\mathbb{R}^D\right)^K$
5: **return** S

Theorem 17.14. *Given $X = (x_n)_{n\in[N]}$, $K \in \mathbb{N}$, $\beta \in \mathbb{R}_+$, $(\omega_k)_{k\in[K]} \in \Delta_{K-1}$ with $\forall k \in [K]\colon \omega_k > 0$, and $\epsilon \in (0,1]$, Algorithm 28 computes a set $S \subset (\mathbb{R}^D)^K$ such that the following property is satisfied:*

Consider some fixed but arbitrary soft K-clustering P of X with

$$\forall k \in [K] \; : \mathbf{var}\left(A_k^{(X,P)}\right) = \frac{2}{\beta} \cdot D \quad \textit{and} \quad \mathbf{w}\left(A_k^{(X,P)}\right) = \omega_k \cdot |X| \tag{17.12}$$

(if such a clustering exists).

With constant probability, there exists a $(\mu_k)_k \in S$ satisfying

$$\begin{aligned} &\mathrm{skm}_X^{(\beta,\omega)}\left((\mu_k)_{k\in[K]}, P\right) - \bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}(P) \leq \epsilon \cdot |X| D \;, \\ &\qquad \forall k \in [K] \colon \mathbf{var}\left(A_k^{(X,P)}, \mu_k\right) \in \left[1,\; 1 + \epsilon \cdot \left(1 + \frac{1/K}{\omega_k}\right)\right] \cdot \frac{2}{\beta} D \;, \quad \textit{and} \\ &\forall k \in [K] \colon \left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mu_k \right\|_2^2 \leq \frac{\epsilon}{4} \cdot D \cdot \frac{2}{\beta}\left(1 + \frac{1/K}{\omega_k}\right) . \end{aligned}$$

The algorithms' runtime is bounded by $|X| D \cdot 2^{\tilde{\mathcal{O}}(K^2/\epsilon)}$ and the size of S is bounded by $2^{\tilde{\mathcal{O}}(K^2/\epsilon)}$.

Proof. Let $\dot{\cup}_{k=1}^K A_k = X_c$ be the hard clustering whose existence is guaranteed by Theorem 3.21 (with respect to X_c and $\tilde{\epsilon}$). Using Corollary 2.26, we can conclude that

$$\frac{1}{c}\mathbf{w}(A_k) \geq \frac{1}{2}\mathbf{w}\left(A_k^{(X,P)}\right) \;, \tag{17.13}$$

$$\left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mathbf{m}(A_k) \right\|_2^2 \leq \frac{\tilde{\epsilon}}{2} \cdot \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} \quad \text{and} \tag{17.14}$$

$$\frac{1}{c}\mathbf{d}(A_k) \leq 4K \cdot \mathbf{d}\left(A_k^{(X,P)}\right) . \tag{17.15}$$

From Theorem 17.13 we know that, with constant probability, there is a $(\mu_k)_{k\in[K]} \in S$ that satisfies

$$\forall k \in [K] \; \mathbf{d}\left(A_k, \mu_k\right) \leq \left(1 + \frac{\tilde{\epsilon}}{2}\right)\mathbf{d}(A_k) + \frac{\tilde{\epsilon}}{2K}\sum_{l=1}^{K} \mathbf{d}(A_l) \; . \tag{17.16}$$

In the first part of this proof, we upper bound the term $\left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mu_k \right\|_2^2$. Due to Lemma A.3, we have

$$\left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mu_k \right\|_2^2 \leq 2\left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mathbf{m}(A_k) \right\|_2^2 + 2\left\| \mathbf{m}(A_k) - \mu_k \right\|_2^2 .$$

We can bound the first summand by

$$\begin{aligned} 2\left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mathbf{m}(A_k) \right\|_2^2 &\leq \tilde{\epsilon} \cdot \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)} && \text{(Equation (17.14))} \\ &= \tilde{\epsilon} \cdot \mathbf{var}\left(A_k^{(X,P)}\right) \\ &= \tilde{\epsilon} \cdot \frac{2}{\beta} \cdot D \; . && \text{(Equation (17.12))} \end{aligned}$$

We can bound the second summand by

$$
\begin{aligned}
&2\left\|\mathbf{m}(A_k)-\mu_k\right\|_2^2 \\
&= 2\cdot\frac{1}{\mathbf{w}(A_k)}\cdot\left(\mathbf{d}\left(A_k,\mu_k\right)-\mathbf{d}(A_k)\right) && \text{(Lemma 2.20)}\\
&\leq 2\cdot\frac{1}{\mathbf{w}(A_k)}\left(\frac{\tilde{\epsilon}}{2}\mathbf{d}(A_k)+\frac{\tilde{\epsilon}}{K}\sum_{l=1}^{K}\mathbf{d}(A_l)\right) && \text{(Equation (17.16))}\\
&= \tilde{\epsilon}\cdot\frac{1}{\mathbf{w}(A_k)}\cdot\left(\mathbf{d}(A_k)+\frac{2}{K}\cdot\sum_{l=1}^{K}\mathbf{d}(A_l)\right)\\
&\leq \tilde{\epsilon}\cdot\frac{8K}{\mathbf{w}\left(A_k^{(X,P)}\right)}\cdot\left(\mathbf{d}\left(A_k^{(X,P)}\right)+\frac{2}{K}\cdot\sum_{l=1}^{K}\mathbf{d}\left(A_l^{(X,P)}\right)\right) && \text{(Equation (17.15) and (17.13))}\\
&= 8K\tilde{\epsilon}\cdot\left(\frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)}+\frac{2}{K}\cdot\frac{1}{\mathbf{w}\left(A_k^{(X,P)}\right)}\sum_{l=1}^{K}\mathbf{d}\left(A_l^{(X,P)}\right)\right)\\
&= 8K\tilde{\epsilon}\cdot\left(D\frac{2}{\beta}+\frac{2}{K}\cdot\frac{1}{|X|\cdot\omega_k}\sum_{l=1}^{K}\left(D\frac{2}{\beta}\cdot\mathbf{w}\left(A_l^{(X,P)}\right)\right)\right) && \text{(Equation (17.12))}\\
&= 8K\tilde{\epsilon}\cdot D\frac{2}{\beta}\cdot\left(1+\frac{2}{K}\cdot\frac{1}{|X|\cdot\omega_k}\sum_{l=1}^{K}\mathbf{w}\left(A_l^{(X,P)}\right)\right)\\
&= 8K\tilde{\epsilon}\cdot D\frac{2}{\beta}\cdot\left(1+2\cdot\frac{1/K}{\omega_k}\right). && \text{(X is unweighted)}
\end{aligned}
$$

A combination of the bounds on the first and second summand gives

$$
\begin{aligned}
\left\|\mathbf{m}\left(A_k^{(X,P)}\right)-\mu_k\right\|_2^2 &\leq \tilde{\epsilon}\cdot D\cdot\frac{2}{\beta}+8K\tilde{\epsilon}\cdot D\frac{2}{\beta}\cdot\left(1+2\cdot\frac{1/K}{\omega_k}\right)\\
&= \tilde{\epsilon}\cdot D\cdot\frac{2}{\beta}\left(1+8K\cdot\left(1+2\cdot\frac{1/K}{\omega_k}\right)\right)\\
&= \tilde{\epsilon}\cdot D\cdot\frac{2}{\beta}\left(1+8K+16K\cdot\frac{1/K}{\omega_k}\right)\\
&\leq 16K\tilde{\epsilon}\cdot D\cdot\frac{2}{\beta}\left(1+\frac{1/K}{\omega_k}\right). && (17.17)\\
&\leq \frac{\epsilon}{4}\cdot\cdot D\cdot\frac{2}{\beta}\left(1+\frac{1/K}{\omega_k}\right) && (\tilde{\epsilon}=\tfrac{\epsilon}{64\cdot K})
\end{aligned}
$$

Next, we will use this bound to prove the claims stated in the theorem.

In the first part of this proof, we prove the second claim from the theorem. Observe that

$$
\begin{aligned}
\mathbf{var}\left(A_k^{(X,P)},\mu_k\right) &= \frac{\mathbf{d}\left(A_k^{(X,P)},\mu_k\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)}\\
&= \frac{\mathbf{d}\left(A_k^{(X,P)}\right)}{\mathbf{w}\left(A_k^{(X,P)}\right)}+\left\|\mu_k-\mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2 && \text{(Lemma 2.20)}\\
&= \mathbf{var}\left(A_k^{(X,P)}\right)+\left\|\mu_k-\mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2\\
&= \frac{2}{\beta}\cdot D+\left\|\mu_k-\mathbf{m}\left(A_k^{(X,P)}\right)\right\|_2^2. && \text{(Equation (17.12))}
\end{aligned}
$$

On the one hand, this directly shows that $\mathbf{var}\left(A_k^{(X,P)},\mu_k\right)\geq 2/\beta\cdot D$. On the other hand, we

have

$$\begin{aligned}
\mathbf{var}\left(A_k^{(X,P)}, \mu_k\right) &= \frac{2}{\beta} \cdot D + \left\| \mu_k - \mathbf{m}\left(A_k^{(X,P)}\right) \right\|_2^2 \\
&\leq \frac{2}{\beta} \cdot D + 16K\tilde{\epsilon} \cdot D \cdot \frac{2}{\beta}\left(1 + \frac{1/K}{\omega_k}\right) && \text{(Equation (17.17))} \\
&\leq \frac{2}{\beta} \cdot D\left(1 + 16K\tilde{\epsilon} \cdot \left(1 + \frac{1/K}{\omega_k}\right)\right) \\
&\leq \frac{2}{\beta} \cdot D\left(1 + \epsilon \cdot \left(1 + \frac{1/K}{\omega_k}\right)\right) . && (\tilde{\epsilon} \leq \epsilon/(16K))
\end{aligned}$$

This yields the second claim.

In the last part of this proof, we prove the first claim from the theorem. Observe that

$$\begin{aligned}
&\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}, \mu_k\right) \\
&= \left(\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}\right)\right) + \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right) \left\| \mathbf{m}\left(A_k^{(X,P)}\right) - \mu_k \right\|_2^2 && \text{(Lemma 2.20)} \\
&\leq \left(\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}\right)\right) + 16K\tilde{\epsilon} \cdot D \cdot \frac{2}{\beta} \cdot \sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right)\left(1 + \frac{1/K}{\omega_k}\right) && \text{(Equation (17.17))} \\
&= \left(\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}\right)\right) + 16K\tilde{\epsilon} \cdot D \cdot \frac{2}{\beta} \cdot \left(\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right) + \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbf{w}\left(A_k^{(X,P)}\right)}{\omega_k}\right) .
\end{aligned}$$

Observe that $\frac{\mathbf{w}\left(A_k^{(X,P)}\right)}{\omega_k} = |X|$ due to (17.12) and $\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right) = |X|$ since X is unweighted. Hence,

$$\begin{aligned}
\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}, \mu_k\right) &\leq \left(\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}\right)\right) + 16K\tilde{\epsilon} \cdot D \cdot \frac{2}{\beta} \cdot 2|X| \\
&\leq \left(\sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}\right)\right) + \epsilon \cdot \frac{1}{\beta} \cdot D|X| , && (17.18)
\end{aligned}$$

where we use the fact that $\tilde{\epsilon} \leq \epsilon/(64K)$. Recall that, due to (17.5) and by Definition 2.16, we have

$$\beta = \frac{2 \cdot D}{\mathbf{var}\left(A_k^{(X,P)}\right)} = 2D \cdot \frac{\mathbf{w}\left(A_k^{(X,P)}\right)}{\mathbf{d}\left(A_k^{(X,P)}\right)} . \tag{17.19}$$

From (17.18) and (17.19) we can conclude that

$$\begin{aligned}
\beta \cdot \sum_{k=1}^{K} \mathbf{d}\left(A_k^{(X,P)}, \mu_k\right) &\leq \left(\sum_{k=1}^{K} \beta \mathbf{d}\left(A_k^{(X,P)}\right)\right) + \epsilon \cdot D|X| && \text{(Equation (17.18))} \\
&\leq 2D\left(\sum_{k=1}^{K} \mathbf{w}\left(A_k^{(X,P)}\right)\right) + \epsilon \cdot D|X| && \text{(Equation (17.19))} \\
&\leq (2+\epsilon) \cdot D|X| . && (X \text{ is unweighted})
\end{aligned}$$

With this equation and Observation 17.10, we conclude that

$$\mathrm{skm}_X^{(\beta,\omega)}\left((\mathbf{m}(A_k))_{k\in[K]}, (\tilde{p}_{nk})_{n,k}\right) - \bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}((\tilde{p}_{nk})_{n,k}) \leq (2+\epsilon) \cdot D|X| - 2D|X| = \epsilon D|X| .$$

This yields the claim. □

Unfortunately, our bounds depend on the given weights $(\omega_k)_{k\in[K]}$. If we are given *uniform* weights, i.e., $\forall k \in [K]: \omega_k = 1/K$, then Algorithm 28 delivers a set of candidate means that contains the means that we search for:

Corollary 17.15. *Given $X = (x_n)_{n\in[N]}$, $K \in \mathbb{N}$, $\beta \in \mathbb{R}_+$, and $\epsilon \in (0,1]$, Algorithm 28 computes a set $S \subset (\mathbb{R}^D)^K$ such that the following property is satisfied:*

Consider some fixed but arbitrary soft K-clustering $P = (\tilde{p}_{nk})_{n\in[N],k\in[K]}$ with

$$\forall k \in [K] : \mathbf{var}\left(A_k^{(X,P)}\right) = \frac{2}{\beta}\cdot D \quad \text{and} \quad \mathbf{w}\left(A_k^{(X,P)}\right) = \frac{|X|}{K}$$

(if such a clustering exists).

With constant probability, there exists $(\mu_k)_k \in S$ satisfying

$$\begin{aligned} \mathrm{skm}_X^{(\beta,\omega)}\left((\mu_k)_{k\in[K]}, P\right) - \bar{\mathrm{s}}\mathrm{km}_X^{(\beta,\omega)}(P) &\leq \epsilon\cdot|X|D\ , \\ \forall k\in[K]:\ \mathbf{var}\left(A_k^{(X,P)},\mu_k\right) &\in [1,\ 1+2\epsilon]\cdot\frac{2}{\beta}D\ , \quad \text{and} \\ \forall k\in[K]:\ \left\|\mathbf{m}\left(A_k^{(X,P)}\right)-\mu_k\right\|_2^2 &\leq \frac{\epsilon}{2}\cdot D\frac{2}{\beta}\ . \end{aligned}$$

The algorithms' runtime is bounded by $|X|D\cdot 2^{\tilde{\mathcal{O}}(K^2/\epsilon)}$ and the size of S is bounded by $2^{\tilde{\mathcal{O}}(K^2/\epsilon)}$.

This latter fact is unsurprising as it resembles a main difference between the soft K-means cost function and the K-means cost function, for which the algorithm of Bhattacharya et al. (2016) has been designed (see Theorem 17.13): In the K-means cost function, the clusters are (implicitly) weighted equally. That is, the cost of each cluster contributes equally to the overall K-means cost and is not more important (i.e., has a higher weight) than that of any other cluster. In contrast, in the soft K-means cost function, the given weighs $\omega_1,\dots,\omega_K$ might not be all equal to $1/K$. That is, the cost of the clusters might not be equally important. The cost of a cluster with weight ω_k contributes more to the overall soft K-means cost than the cost of a cluster with weight $\omega_l < \omega_k$.

Even though we can now determine a set that contains approximate mean vectors, we do not have a solution yet. We are still missing an appropriate approximate soft clustering.

17.5.3 Determining the Soft Clustering

In the last section, we showed that the algorithm from Bhattacharya et al. (2016) can be used to determine likely mean vectors. However, it remains to determine a suitable soft clustering. Formally, assume that we have found good approximate means $(\mu_k)_{k\in[K]}$. How can we determine a *likely* soft clustering $P = (p_{nk})_{n,k}$ where the soft clusters have the appropriate variances, mean vectors, and weights?

We do not know how to solve this problem yet. However, there is a way to solve a very simplified version of this problem:

First, let us drop the constraint on the weights and means and simplify the constraint on the variances. More precisely, replace the lower and upper bound $[1,1+\epsilon]2/\beta\cdot D$ on the variances by a single lower bound $2/\beta\cdot D$. Observe that this lower bound can also be described as follows:

$$\begin{aligned} \mathbf{var}\left(A_k^{(X,P)},\mu_k\right) = \frac{\sum_{n=1}^N p_{nk}\left\|x_n-\mu_k\right\|_2^2}{\sum_{n=1}^N p_{nk}} &\geq \frac{2}{\beta}\cdot D \\ \Leftrightarrow \quad \sum_{n=1}^N p_{nk}\left(\left\|x_n-\mu_k\right\|_2^2 - \frac{2}{\beta}\cdot D\right) &\geq 0\ . \end{aligned}$$

Second, let us assume that the observations x_n and the given means μ_k all take values in $\mathbb{Z}^D$. Moreover, assume that the means lie inside the convex hull of the point set $\{x_n \mid n \in [N]\}$. These simplifications and assumptions lead us to the following observation:

We consider observations $X = (x_n)_{n\in[N]} \subset \mathbb{Z}^D$, $K \in \mathbb{N}$, and a variance $2/\beta \in \mathbb{N}$. Moreover, we fix some mean vectors $(\mu_k)_{k\in[K]} \subseteq \mathbb{Z}^D$. For all $n \in [N]$ and $k \in [K]$, let $d_{nk} \coloneqq \|x_n - \mu_k\|_2^2$ and $f_{nk} \coloneqq \left(\|x_n - \mu_k\|_2^2 - \frac{2}{\beta} \cdot D\right)$. Given these constants, we can formulate the problem of determining a soft clustering, where the variance of each soft cluster with respect to the corresponding mean vector is at least $2/\beta$, as a linear program:

$$\begin{array}{lll} \text{minimize} & \sum_{k=1}^{K}\sum_{n=1}^{N} \tilde{p}_{nk} d_{nk} & \\ \text{subject to} & \sum_{n=1}^{N} \tilde{p}_{nk} f_{nk} \quad \geq 0, & k = 1,\ldots,K \\ & \sum_{k=1}^{K} \tilde{p}_{nk} \quad = 1 & n = 1,...,N \\ & \tilde{p}_{nk} \quad \geq 0 & n = 1,...,N; k = 1,\ldots,K \end{array} .$$

Given the restriction that we only consider observations and mean vectors in $\mathbb{Z}^D$, we can use the Ellipsoid method to solve this linear program in polynomial time (Megiddo, 1986). Observe that the number of bits in a binary representation of the coefficients in the linear program is polynomial in $|X|$, K, $\log(\mathrm{r}_d(X))$ with $d \in [D]$, and $\log(2/\beta)$, where we use the fact that we can shift the whole point set such that $(\min\{(x_n)_d \mid n \in [N]\})_{d\in[D]} = 0_D$ and the fact that the mean vectors lie inside the convex hull of the point set. Hence, the ellipsoid method needs time $\mathrm{poly}(|X|, K, \log(\mathrm{r}_1(X)),\ldots,\log(\mathrm{r}_D(X),\log(2/\beta))$.

17.6 Conclusions

Even though we tried to derive a variant of the soft K-means problem that is (hopefully) easier to analyse, there is still no proper approximation algorithm for it. However, so far, our analysis again focused on techniques known from (constrained) K-means clustering. Just as in the case of fuzzy K-means (see Section 13.4), one might obtain a (more specific) algorithm easier if one accepts a larger approximation factor. Last but not least, there might be other ways of relaxing the soft K-means problem that might lead us to a better understanding of the soft K-means problem.

Part IV

Appendix

"Location, location, location!"

Appendix A

Three Handy Lemmata

In this appendix, we state three fundamental lemmata that we use throughout this thesis. For an overview of our handy notation, we refer to the Cheat Sheet at the very beginning of the thesis.

Lemma A.1. *Let $\epsilon \in [0,1]$ and $m \in \mathbb{N}$. Then, for all $i \in [m]$ it holds*

$$\left(1+\frac{\epsilon}{2m}\right)^i \leq 1+i\cdot\frac{\epsilon}{m}\,.$$

Proof. The proof is by induction. For $i=1$, the claim is clearly true. Assume that the claim holds true for an arbitrary but fixed $i \in [m]$. Observe that

$$\begin{aligned}
\left(1+\frac{\epsilon}{2m}\right)^{i+1} &= \left(1+\frac{\epsilon}{2m}\right)^i \cdot \left(1+\frac{\epsilon}{2m}\right) \\
&\leq \left(1+i\cdot\frac{\epsilon}{m}\right)\cdot\left(1+\frac{\epsilon}{2m}\right) && \text{(by induction hypothesis)} \\
&= 1+i\cdot\frac{\epsilon}{m}+\frac{\epsilon}{2m}+i\cdot\frac{\epsilon^2}{2m^2} \\
&= 1+i\cdot\frac{\epsilon}{m}+\frac{\epsilon}{m}\cdot\left(\frac{1}{2}+\frac{\epsilon}{2}\frac{i}{m}\right) \\
&\leq 1+i\cdot\frac{\epsilon}{m}+\frac{\epsilon}{m} && \text{(since } i\leq m \text{ and } \epsilon\leq 1\text{)} \\
&= 1+(i+1)\cdot\frac{\epsilon}{m}\,.
\end{aligned}$$

□

Lemma A.2. *For all $a,b,c \in \mathbb{R}$ we have*

1. $2ab \leq a^2+b^2$,
2. $(a+b)^2 \leq 2(a^2+b^2)$ *and*
3. $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$.

Proof. Fix arbitrary $a,b,c \in \mathbb{R}$. Observe that $0 \leq (a-b)^2 = a^2-2ab+b^2$. Hence, $2ab \leq a^2+b^2$ (Item 1). Consequently, $(a+b)^2 = a^2+2ab+b^2 \leq 2(a^2+b^2)$ (Item 2) and $(a+b+c)^2 = a^2+b^2+c^2+2ab+2ac+2cb \leq 3(a^2+b^2+c^2)$ (Item 3). □

Lemma A.3. *For all $a,b,c,x,y \in \mathbb{R}^D$ and all finite $A \subseteq \mathbb{R}^D$, we have*

1. $\|a-b\|_2 \leq \|a-b\|_2 + \|b-c\|_2$,
2. $2\langle a,\, b\rangle \leq \|a\|_2^2 + \|b\|_2^2$,

3. $\|a+b\|_2^2 \leq 2(\|a\|_2^2 + \|b\|_2^2)$, *and*

4. $\min\{\|x-a\|_2 \mid a \in A\} \leq \min\left\{\|y-a\|_2 \,\middle|\, a \in A\right\} + \|x-y\|_2$.

Proof. Item 1 is the well-known triangle inequality for vectors (Stewart, 2009, p. 822). Write $a = (a_1 \dots a_D)^T$ and $b = (b_1 \dots b_D)^T$. With Lemma A.2, we can conclude $2\langle a, b\rangle = \sum_{d=1}^{D}(2a_d b_d) \leq \sum_{d=1}^{D}(a_d^2 + b_d^2) = \|a\|_2^2 + \|b\|_2^2$ (Item 2) and moreover, $\|a+b\|_2^2 = \sum_{d=1}^{D}(a_d + b_d)^2 \leq \sum_{d=1}^{D}(2a_d^2 + 2b_d^2) = \|a\|_2^2 + \|b\|_2^2$ (Item 3).

Next, consider some $a' \in A$ with $\min\left\{\|y-a\|_2 \,\middle|\, a \in A\right\} = \left\|y-a'\right\|_2$. With Item 1 we can conclude that $\min\{\|x-a\|_2 \mid a \in A\} \leq \left\|x-a'\right\|_2 \leq \|x-y\|_2 + \left\|y-a'\right\|_2 = \min\left\{\|y-a\|_2 \,\middle|\, a \in A\right\}$ (Item 4). □

Bibliography

Achtert, Goldhofer, Kriegel, Schubert, and Zimek (2012). Evaluation of Clusterings Metrics and Visual Support. *ICDE 2012*, 0:1285–1288.

Ackerman, M. and Ben-David, S. (2009). Clusterability: A Theoretical Study. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, pages 1–8.

Ackermann, M. R. (2009). *Algorithms for the Bregman k-Median problem*. PhD thesis, University of Paderborn.

Ackermann, M. R., Blömer, J., and Sohler, C. (2010). Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms*, 6(4):59:1–59:26.

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2005). Geometric approximation via coresets. In *Combinatorial and Computational Geometry*, pages 1–30. University Press.

Aggarwal, A., Deshpande, A., and Kannan, R. (2009). Adaptive Sampling for k-Means Clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 5687 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin Heidelberg.

Ailon, N. and Chazelle, B. (2006). Approximate Nearest Neighbors and the Fast Johnson-Lindenstrauss Transform. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 557–563, New York, NY, USA. ACM.

Ailon, N. and Liberty, E. (2013). An Almost Optimal Unrestricted Fast Johnson-Lindenstrauss Transform. *ACM Transactions on Algorithms*, 9(3):21:1–21:12.

Anderson, J., Belkin, M., Goyal, N., Rademacher, L., and Voss, J. R. (2014). The More, the Merrier: the Blessing of Dimensionality for Learning Large Gaussian Mixtures. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, pages 1135–1164.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035. Society for Industrial and Applied Mathematics.

Asuncion (2007). UCI Machine Learning Repository.

Awasthi, P., Blum, A., and Sheffet, O. (2010a). Clustering under natural stability assumptions. *Computer Science Department*, page 123.

Awasthi, P., Blum, A., and Sheffet, O. (2010b). Stability Yields a PTAS for k-Median and k-Means Clustering. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 309–318.

Bachem, O., Lucic, M., Hassani, H., and Krause, A. (2016). Fast and Provably Good Seedings for k-Means. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 55–63. Curran Associates, Inc.

Bādoiu, M., Har-Peled, S., and Indyk, P. (2002). Approximate Clustering via Core-sets. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 250–257, New York, NY, USA. ACM.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005). Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6:1705–1749.

Basu, S., Davidson, I., and Wagstaff, K. (2008). *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press.

Ben-David, S. (2015). Computational Feasibility of Clustering under Clusterability Assumptions. *CoRR*, abs/1501.00437.

Ben-David, S. and Reyzin, L. (2014). Data stability in clustering: A closer look . *Theoretical Computer Science*, 558:51 – 61. Algorithmic Learning Theory.

Bezdek, J., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203.

Bezdek, J., Hathaway, R., Sabin, M., and Tucker, W. (1987). Convergence theory for fuzzy c-means: Counterexamples and repairs. *Systems, Man and Cybernetics, IEEE Transactions on*, 17(5):873–877.

Bhattacharya, A., Jaiswal, R., and Kumar, A. (2016). Faster Algorithms for the Constrained k-Means Problem. In Ollinger, N. and Vollmer, H., editors, *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:13, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Biernacki (2004). Initializing EM using the properties of its trajectories in Gaussian mixtures. *Statistics and Computing*, 14(3):267–279.

Bilmes, J. (1998). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report TR-97-021, International Computer Science Institute.

Bishop, C. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blömer, J., Brauer, S., and Bujna, K. (2015). Complexity and Approximation of the Fuzzy K-Means Problem. *CoRR*, abs/1512.05947.

Blömer, J., Brauer, S., and Bujna, K. (2016). A Theoretical Analysis of the Fuzzy K-Means Problem. In *IEEE 16th International Conference on Data Mining (ICDM 2016)*, pages 805–810, Barcelona, Spain. IEEE.

Blömer, J., Brauer, S., and Bujna, K. (2017). On coreset constructions for the fuzzy k-means problem. *CoRR*, abs/1612.07516.

Blömer, J. and Bujna, K. (2016). Adaptive Seeding for Gaussian Mixture Models. In *Proceedings of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2016)*, volume 9652 of *Lecture Notes in Computer Science*, pages 296–308, Auckland, New Zealand. Springer.

Blömer, J., Bujna, K., and Kuntze, D. (2014). A Theoretical and Experimental Comparison of the EM and SEM Algorithm. In *22nd International Conference on Pattern Recognition (ICPR 2014)*, pages 1419–1424, Stockholm, Sweden. IEEE.

Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based Clustering of High-dimensional Data: A Review. *Computational Statistics & Data Analysis*, 71:52–78.

Bujna, K. (2016). Supplemental Material (Initialization of the EM Algorithm for GMMs). http://cs.uni-paderborn.de/cuk/forschung/clusteranalyse/adaptive-seeding-for-gaussian-mixture-models/.

Celeux and Govaert (1992). A Classification EM Algorithm for Clustering and Two Stochastic Versions. *Computational Statistics & Data Analysis*, 14(3):315332.

Celeux, G., Chauveau, D., and Diebolt, J. (1995). On Stochastic Versions of the EM Algorithm. Research Report RR-2514, INRIA.

Celeux, G., Chauveau, D., and Diebolt, J. (1996). Stochastic versions of the em algorithm: an experimental study in the mixture case. *Journal of Statistical Computation and Simulation*, 55(4):287–314.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

Chen, K. (2009). On Coresets for K-Median and K-Means Clustering in Metric and Euclidean Spaces and Their Applications. *SIAM Journal on Computing*, 39(3):923–947.

Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. (2001). *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Dang, U. J., Browne, R. P., and McNicholas, P. D. (2015). Mixtures of multivariate power exponential distributions. *Biometrics*, 71(4):1081–1089.

Dasgupta, S. (1999). Learning Mixtures of Gaussians. In *FOCS 1999*, pages 634–644.

Dasgupta, S. (2008). The hardness of k-means clustering. Technical report, Department of Computer Science and Engineering, University of California, San Diego.

Dasgupta, S. and Gupta, A. (2003). An Elementary Proof of a Theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65.

Dasgupta, S. and Schulman, L. (2007). A Probabilistic Analysis of EM for Mixtures of Separated, Spherical Gaussians. *Journal of Machine Learning Research*, 8:203–226.

Dasgupta, S. and Schulman, L. J. (2000). A Two-Round Variant of EM for Gaussian Mixtures. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, UAI'00, pages 152–159, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463.

Dempster, Laird, and Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 39(1):138.

Dias, J. G. and Wedel, M. (2004). An Empirical Comparison of EM, SEM and MCMC Performance for Problematic Gaussian Mixture Likelihoods. *Statistics and Computing*, 14(4):323–332.

Ding, C. and He, X. (2004). K-means Clustering via Principal Component Analysis. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 29–, New York, NY, USA. ACM.

Ding, H. and Xu, J. (2015). A Unified Framework for Clustering Constrained Data Without Locality Property. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1471–1490, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., and Vinay, V. (2004). Clustering Large Graphs via the Singular Value Decomposition. *Machine Learning*, 56(1-3):9–33.

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57.

Effros, M. and Schulman, L. J. (2004). Deterministic clustering with data nets. *Electronic Colloquium on Computational Complexity (ECCC)*, (050).

Färber, I., Günnemann, S., Kriegel, H., Kröger, P., Müller, E., Schubert, E., S., T., and Z., A. (2010). On Using Class-Labels in Evaluation of Clusterings. In *Proc. 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings (MultiClust 2010) in conjunction with 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), Washington, DC, USA*.

Fayyad, Reina, and Bradley (1998). Initialization of Iterative Refinement Clustering Algorithms. In *KDD 1998*, page 194198.

Feldman, D., Faulkner, M., and Krause, A. (2011). Scalable Training of Mixture Models via Coresets. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2142–2150. Curran Associates, Inc.

Feldman, D., Monemizadeh, M., and Sohler, C. (2007). A PTAS for K-means Clustering Based on Weak Coresets. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, SCG '07, pages 11–18, New York, NY, USA. ACM.

Feldman, D., Schmidt, M., and Sohler, C. (2013). Turning Big Data into Tiny Data: Constant-size Coresets for K-means, PCA and Projective Clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1434–1453. SIAM.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2015). Avoiding spurious local maximizers in mixture modeling. *Statistics and Computing*, 25(3):619–633.

Gath, I. and Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):773–780.

Geusebroek, Burghouts, and Smeulders (2005). The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 6(1):103112.

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*. John Hopkins University Press.

Gómez, E., Gomez-Viilegas, M., and Marín, J. (1998). A multivariate generalization of the power exponential family of distributions. *Communications in Statistics - Theory and Methods*, 27(3):589–600.

Gonzalez (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293306.

Gustafson, D. and Kessel, W. (1978). Fuzzy clustering with a fuzzy covariance matrix. In *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, pages 761–766.

Har-Peled, S. and Kushal, A. (2005). Smaller Coresets for K-median and K-means Clustering. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, SCG '05, pages 126–134, New York, NY, USA. ACM.

Har-Peled, S. and Mazumdar, S. (2004). On Coresets for K-means and K-median Clustering. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 291–300, New York, NY, USA. ACM.

Hardt, M. and Price, E. (2015). Tight Bounds for Learning a Mixture of Two Gaussians. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 753–760.

Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952). *Inequalities*. Cambridge University Press.

Hasegawa, S., Imai, H., Inaba, M., and Katoh, N. (1993). Efficient algorithms for variance-based k-clustering. In *Proceedings of the 1st Pacific Conference on Computer Graphics and Applications*, pages 75–89.

Hathaway, R. and Bezdek, J. (1986). Local convergence of the fuzzy c-Means algorithms . *Pattern Recognition*, 19(6):477 – 480.

Hathaway, R. and Bezdek, J. (2001). Fuzzy c-means clustering of incomplete data. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(5):735–744.

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13:795–800.

Hathaway, R. J. and Bezdek, J. C. (1988). Recent convergence results for the fuzzy c-means clustering algorithms. *Journal of Classification*, 5(2):237–247.

Haussler, D. (1992). Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78 – 150.

Hopcroft, J. and Kannan, R. (2017). Foundations of Data Science.

Höppner, F. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.

Höppner, F. and Klawonn, F. (2003). A contribution to convergence theory of fuzzy c-means and derivatives. *IEEE Transactions Fuzzy Systems*, 11(5):682–694.

Hu, Y. and Hathaway, R. J. (2002). On efficiency of optimization in fuzzy c-means. *Neural Parallel & Scientific Computing*, 10(2):141–156.

Huang, M., Xia, Z., Wang, H., Zeng, Q., and Wang, Q. (2012). The range of the value for the fuzzifier of the fuzzy c-means algorithm. *Pattern Recognition Letters*, 33(16):2280 – 2284.

Inaba, M., Katoh, N., and Imai, H. (1994). Applications of Weighted Voronoi Diagrams and Randomization to Variance-based K-clustering. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SoCG '94, pages 332–339, New York, NY, USA. ACM.

Ip, E. H.-s. (1994). *A stochastic EM estimator in the presence of missing data – theory and applications*. PhD thesis, Stanford University.

Jin, C., Zhang, Y., Balakrishnan, S., Wainwright, M. J., and Jordan, M. I. (2016). Local Maxima in the Likelihood of Gaussian Mixture Models: Structural Results and Algorithmic Consequences. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4116–4124.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206.

Kalai, A. T., Moitra, A., and Valiant, G. (2012). Disentangling Gaussians. *Communications of the ACM*, 55(2):113–120.

Kane, D. M. and Nelson, J. (2014). Sparser Johnson-Lindenstrauss Transforms. *Journal of the ACM*, 61(1):4:1–4:23.

Kannan, R. and Vempala, S. (2009). Spectral Algorithms. *Foundations and Trends in Theoretical Computer Science*, 4(3-4):157–288.

Kearns, M., Mansour, Y., and Ng, A. Y. (1997). An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence (UAI'97)*, pages 282–293. Morgan Kaufmann.

Kiefer, J. and Wolfowitz, J. (1956). Consistency of the Maximum Likelihood Estimator in the Presence of Infinitely Many Incidental Parameters. *The Annals of Mathematical Statistics*, 27(4):887–906.

Kim, T., Bezdek, J. C., and Hathaway, R. J. (1988). Optimality tests for fixed points of the fuzzy c-means algorithm. *Pattern Recognition*, 21(6):651–663.

Klawonn, F. (2004). Fuzzy clustering: insights and a new approach. *Mathware & Soft Computing*, 11(3).

Klawonn, F. and Höppner, F. (2003). *What Is Fuzzy about Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier*, pages 254–264. Springer Berlin Heidelberg.

Knuth, D. E. (1997). *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Kriegel, H., Schubert, E., and Zimek, A. (2011). Evaluation of multiple clustering solutions. In *Proceedings of the 2nd MultiClust Workshop: Discovering, Summarizing and Using Multiple Clusterings, Athens, Greece, September 5, 2011, in conjunction with ECML/PKDD 2011*, pages 55–66.

Krüger, Leutnant, Haeb-Umbach, Ackermann, and Blömer (2010). On the initialization of dynamic models for speech features. *Sprachkommunikation 2010*.

Kumar, A., Sabharwal, Y., and Sen, S. (2004). A simple linear time (1+ ε)-approximation algorithm for geometric k-means clustering in any dimensions. In *Proceedings-Annual Symposium on Foundations of Computer Science*, pages 454–462. IEEE.

Kumar, A., Sabharwal, Y., and Sen, S. (2010). Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM*, 57(2):1–32.

Kwedlo (2013). A New Method for Random Initialization of the EM Algorithm for Multivariate Gaussian Mixture Learning. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, page 8190. Springer International Publishing.

Kwedlo (2015). A new random approach for initialization of the multiple restart EM algorithm for Gaussian model-based clustering. *Pattern Analysis and Applications*, 18(4):757–770.

Larsen, K. G. and Nelson, J. (2016). The Johnson-Lindenstrauss Lemma Is Optimal for Linear Dimensionality Reduction. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:11, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Lee, E., Schmidt, M., and Wright, J. (2017). Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43.

Levchenko, K. (2013). Chernoff Bound. (cseweb.ucsd.edu/ klevchen/techniques/chernoff.pdf).

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

Lucic, M., Bachem, O., and Krause, A. (2016). Strong Coresets for Hard and Soft Bregman Clustering with Applications to Exponential Family Mixtures. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 1–9.

Lucic, M., Faulkner, M., Krause, A., and Feldman, D. (2017). Training Mixture Models at Scale via Coresets. *Computing Research Repository*.

Mackay, D. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Magnus, J. R. and Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley, second edition.

Mahajan, M., Nimbhorkar, P., and Varadarajan, K. (2012). The planar k-means problem is NP-hard. *Theoretical Computer Science*, 442:13 – 21.

Maitra (2009). Initializing Partition-Optimization Algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):144–157.

Maitra and Melnykov (2010). Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354376.

Matoušek, J. (2000). On Approximate Geometric k -Clustering. *Discrete & Computational Geometry*, 24(1):61–84.

McDiarmid, C. (1998). *Concentration*, pages 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg.

McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2 edition.

Megiddo, N. (1986). *On the complexity of linear programming*. IBM Thomas J. Watson Research Division.

Meilă and Heckerman (1998). An Experimental Comparison of Several Clustering and Initialization Methods. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, page 386395. Morgan Kaufmann, Inc., San Francisco, CA.

Melnykov and Melnykov (2011). Initializing the EM algorithm in Gaussian mixture models with an unknown number of components. *Computational Statistics & Data Analysis*.

Mitzenmacher, M. and Upfal, E. (2005). *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA.

Nielsen, S. F. (2000a). On simulated EM algorithms . *Journal of Econometrics*, 96(2):267 – 292.

Nielsen, S. F. (2000b). The Stochastic EM Algorithm: Estimation and Asymptotic Results. *Bernoulli*, 6(3):457–489.

Oliveira, J. V. d. and Pedrycz, W. (2007). *Advances in Fuzzy Clustering and Its Applications*. John Wiley & Sons, Inc., New York, NY, USA.

Ostrovsky, R., Rabani, Y., Schulman, L. J., and Swamy, C. (2006). The effectiveness of Lloyd-type methods for the k-means problem. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 165–176. IEEE.

Pal, N., Pal, K., Keller, J., and Bezdek, J. (2005). A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems*, 13(4):517–530.

Schmidt, M. (2014). *Coresets and streaming algorithms for the k-means problem and related clustering objectives*. PhD thesis, Universität Dortmund.

Shlens, J. (2003). A Tutuorial on Principal Component Analysis – Derivation, Discussion and Singular Value Decomposition.

Stewart, J. (2009). *Calculus*. Cengage Learning.

Tang, C. and Monteleoni, C. (2016). On Lloyd's Algorithm: New Theoretical Insights for Clustering in Practice. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 1280–1289.

Thiesson (1995). *Accelerated quantification of Bayesian networks with incomplete data*. University of Aalborg, Institute for Electronic Systems, Department of Mathematics and Computer Science.

Tibshirani, R., Guenther, W., and Hastie., T. (2001). Estimating the Number of Clusters in a Data Set via the Gap Statistic. *Journal of the Royal Statistical Society Series B*.

Timm, H., Borgelt, C., Döring, C., and Kruse, R. (2004). An extension to possibilistic fuzzy cluster analysis. *Fuzzy Sets and systems*, 147(1):3–16.

Vempala, S. and Wang, G. (2004). A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860.

Verbeek, Vlassis, and Kröse (2003). Efficient greedy learning of Gaussian mixture models. *Neural computation*, 15(2):469485.

von Luxburg, U., Williamson, R. C., and Guyon, I. (2012). Clustering: Science or art? In Guyon, I., Dror, G., Lemaire, V., Taylor, G., and Silver, D., editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 65–79, Bellevue, Washington, USA. PMLR.

Vose, M. D. (1991). A Linear Algorithm for Generating Random Numbers with a Given Distribution. *IEEE Transactions on Software Engineering*, 17(9):972–975.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 577–584.

Wald, A. (1949). Note on the Consistency of the Maximum Likelihood Estimate. *The Annals of Mathematical Statistics*, 20(4):595–601.

Watt, J., Borhani, R., and Katsaggelos, A. K. (2016). *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge University Press.

Wei, D. (2016). A Constant-Factor Bi-Criteria Approximation Guarantee for k-means++. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 604–612.

Wu, C. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95103.

Xu, L. and Jordan, M. I. (1996). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computing*, 8(1):129–151.

Yang, M.-S. (1993). A survey of fuzzy clustering. *Mathematical and Computer Modelling*, 18(11):1–16.

Zhang, J. and Liang, F. (2010). Robust Clustering Using Exponential Power Mixtures. *Biometrics*, 66(4):1078–1086.