Pontifícia Universidade Católica
do Rio de Janeiro

Raphael Araujo Sampaio

A Study on Ellipsoidal Clustering

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática .

Advisor : Prof. Marcus Vinicius Soledade Poggi de Aragao
Co-advisor: Prof. Thibaut Victor Gaston Vidal

Rio de Janeiro
September 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Raphael Araujo Sampaio

A Study on Ellipsoidal Clustering

Dissertation presented to the Programa de Pós–graduação em Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática . Approved by the undersigned Examination Committee.

Prof. Marcus Vinicius Soledade Poggi de Aragao
Advisor
Departamento de Informática – PUC-Rio

Prof. Thibaut Victor Gaston Vidal
Co-advisor
Departamento de Informática – PUC-Rio

Prof. Marco Serpa Molinaro
Departamento de Informática – PUC-Rio

Prof. Álvaro de Lima Veiga Filho
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Marcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, September 24th, 2018

Raphael Araujo Sampaio

The author graduated in Computer Engineering from Pontifical Catholic University of Rio de Janeiro — PUC-Rio in 2015. He did his Master of Science in Computer Science at PUC-Rio with a full CAPES scholarship.

Bibliographic data

Sampaio, Raphael Araujo

A Study on Ellipsoidal Clustering / Raphael Araujo Sampaio; advisor: Marcus Vinicius Soledade Poggi de Aragao; co-advisor: Thibaut Victor Gaston Vidal. – 2018.

104 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2018.

Inclui bibliografia

1. Informática – Teses. 2. Reconhecimento de padrão;. 3. Agrupamentos Elipsoidais;. 4. Regularizações;. 5. Random Swap;. 6. Algoritmos Genéticos;. I. Poggi de Aragao, Marcus Vinicius Soledade. II. Vidal, Thibaut Victor Gaston. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Acknowledgments

First and foremost, to my advisors, Marcus Poggi and Thibaut Vidal, for giving me the opportunity, motivation and the assurance that my work was held to high standards at all moments.

To my friend, Joaquim Dias Garcia, for providing me with unfailing support and immense knowledge throughout this research since day one.

To my girlfriend, Julia, for all the partnership and devotion by my side during all this journey.

To PUC-Rio, PSR and Tecgraf, for all the collaboration, structure, teamwork and support to face such challenges and learning with them.

Finally, my deepest gratitude goes to my family, especially my lovely mother, Regina, and my cousins, Samara and Carol, for all love and continuous encouragement they have given throughout my life.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Abstract

Sampaio, Raphael Araujo; Poggi de Aragao, Marcus Vinicius Soledade (Advisor); Vidal, Thibaut Victor Gaston (Co-Advisor). **A Study on Ellipsoidal Clustering**. Rio de Janeiro, 2018. 104p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Unsupervised cluster analysis, the process of grouping sets of points according to one or more similarity criteria, plays an essential role in various fields. The two most popular algorithms for this process are the k-means and the Gaussian Mixture Models (GMM). The former assigns each point to a single cluster and uses Euclidean distance as similarity. The latter determines a probability matrix of points to belong to clusters, and the Mahalanobis distance is the underlying similarity. Apart from the difference in the assignment method – the so-called hard assignment for the former and soft assignment for the latter – the algorithms also differ concerning the cluster structure, or shape: the k-means considers spherical structures in the data; while the GMM considers ellipsoidal ones through the estimation of covariance matrices. In this work, a mathematical optimization problem that combines the hard assignment with the ellipsoidal cluster structure is detailed and formulated. Since the estimation of the covariance plays a major role in the behavior of ellipsoidal cluster structures, regularization techniques are explored. In this context, two meta-heuristic methods, a Random Swap perturbation and a hybrid genetic algorithm, are adapted, and their impact on the improvement of the performance of the methods is studied. The central objective is three-fold: to gain an understanding of the conditions in which ellipsoidal clustering structures are more beneficial than spherical ones; to determine the impact of covariance estimation with regularization methods; and to analyze the effect of global optimization meta-heuristics on unsupervised cluster analysis. Finally, in order to provide grounds for comparison of the present findings to future related works, a database was generated together with an extensive benchmark containing an analysis of the variations of different sizes, shapes, number of clusters, and separability and their impact on the results of different clustering algorithms. Furthermore, packages written in the Julia language have been made available with the algorithms studied throughout this work.

Keywords

Pattern Recognition; Ellipsoidal Clustering; Regularization; Random Swap; Genetic Algorithm;

Resumo

Sampaio, Raphael Araujo; Poggi de Aragao, Marcus Vinicius Soledade; Vidal, Thibaut Victor Gaston. **Um Estudo Sobre Agrupamento Baseado em Distribuições Elípticas**. Rio de Janeiro, 2018. 104p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A análise de agrupamento não supervisionado, o processo que consistem em agrupar conjuntos de pontos de acordo com um ou mais critérios de similaridade, tem desempenhado um papel essencial em vários campos. O dois algoritmos mais populares para esse processão são o k-means e o Gaussian Mixture Models (GMM). O primeiro atribui cada ponto a um único cluster e usa a distância Euclidiana como similaridade. O último determina uma matriz de probabilidade de pontos pertencentes a clusters, e usa distância de Mahalanobis como similaridade. Além da diferença no método de atribuição – a chamada atribuição *hard* para o primeiro e a atribuição *soft* para o último – os algoritmos também diferem em relação à estrutura do cluster, ou forma: o k-means considera estruturas esféricas no dados; enquanto o GMM considera elipsoidais através da estimação de matrizes de covariância. Neste trabalho, um problema de otimização matemática que combina a atribuição *hard* com a estrutura do cluster elipsoidal é detalhado e formulado. Uma vez que a estimativa da covariância desempenha um papel importante no comportamento de estruturas agrupamentos elipsoidais, técnicas de regularizações são exploradas. Neste contexto, dois métodos de meta-heurística, uma perturbação *Random Swap* e um algoritmo híbrido genético, são adaptados, e seu impacto na melhoria do desempenho dos métodos é estudado. O objetivo central dividido em três: compreender as condições em que as estruturas de agrupamento elipsoidais são mais benéficas que as esféricas; determinar o impacto da estimativa de covariância com os métodos de regularização; e analisar o efeito das meta-heurísticas de otimização global na análise de agrupamento não supervisionado. Finalmente, a fim de fornecer bases para a comparação das presentes descobertas com futuros trabalhos relacionados, foi gerada uma base de dados com um extenso benchmark contendo análise das variações de diferentes tamanhos, formas, número de grupos e separabilidade, e seu impacto nos resultados de diferentes algoritmos de agrupamento. Além disso, pacotes escritos na linguagem Julia foram disponibilizados com os algoritmos estudados ao longo deste trabalho.

Palavras-chave

Reconhecimento de padrão; Agrupamentos Elipsoidais; Regularizações; Random Swap; Algoritmos Genéticos;

Table of contents

List of figures

List of tables

List of Abreviations

EM – Expectation Maximization

ILS – Iterated Local Search

RS – Random Swap

HG – Hybrid Genetic Algorithm

RS-means – Random swap with K-means as local search

HG-means – Hybrid genetic algorithm with K-means as local search

KE – K-ellipses

RS-KE – Random swap with K-ellipses as local search

HG-KE – Hybrid genetic algorithm with K-ellipses as local search

RKE (Shrunk) – Regularized K-ellipses (Shrunk method)

RS-RKE (Shrunk) – Random swap with Regularized K-ellipses (Shrunk method) as local search

HG-RKE (Shrunk) – Hybrid genetic algorithm with Regularized K-ellipses (Shrunk method) as local search

RKE (LW) – Regularized K-ellipses (Ledoit-Wolf method)

RS-RKE (LW) – Random swap with Regularized K-ellipses (Ledoit-Wolf method) as local search

HG-RKE (LW) – Hybrid genetic algorithm with Regularized K-ellipses (Ledoit-Wolf method) as local search

RKE (OAS) – Regularized K-ellipses (OAS method)

RS-RKE (OAS) – Random swap with Regularized K-ellipses (OAS method) as local search

HG-RKE (OAS) – Hybrid genetic algorithm with Regularized K-ellipses (OAS method) as local search

RS C-means – Random swap with C-means as local search

HG C-means – Hybrid genetic algorithm with as local search

GMM – Gaussian Mixture Models

RS-GMM – Random swap with Gaussian Mixture Models as local search

HG-GMM – Hybrid genetic algorithm with Gaussian Mixture Models as local search

RGMM (Shrunk) – Regularized Gaussian Mixture Models (Shrunk method)

RS-RGMM (Shrunk) – Random swap with Regularized Gaussian Mixture Models (Shrunk method) as local search

HG-RGMM (Shrunk) – Hybrid genetic algorithm with Regularized Gaussian Mixture Models (Shrunk method) as local search

RGMM (LW) – Regularized Gaussian Mixture Models (Ledoit-Wolf method)

RS-RGMM (LW) – Random swap with Regularized Gaussian Mixture Models (Ledoit-Wolf method) as local search

HG-RGMM (LW) – Hybrid genetic algorithm with Regularized Gaussian Mixture Models (Ledoit-Wolf method) as local search

RGMM (OAS) – Regularized Gaussian Mixture Models (OAS method)

RS-RGMM (OAS) – Random swap with Regularized Gaussian Mixture Models (OAS method) as local search

HG-RGMM (OAS) – Hybrid genetic algorithm with Regularized Gaussian Mixture Models (OAS method) as local search

ARI – Adjusted Rand Index

CI – Centroid Index

1
Introduction

The information through the internet is increasing more and more [3]. In 2018, just a minute on the internet can generate an overwhelming amount of information, as presented in Figure 1.1, taken from Lewis and Callahan of Cumulus Media. In a matter of a decade, the internet will completely change the way that we do business and communicate, as forecast by an International Data Corporation (IDC) study [4]: by 2025, the information generated will grow up to 163 zettabytes, ten times the amount generated in 2016.

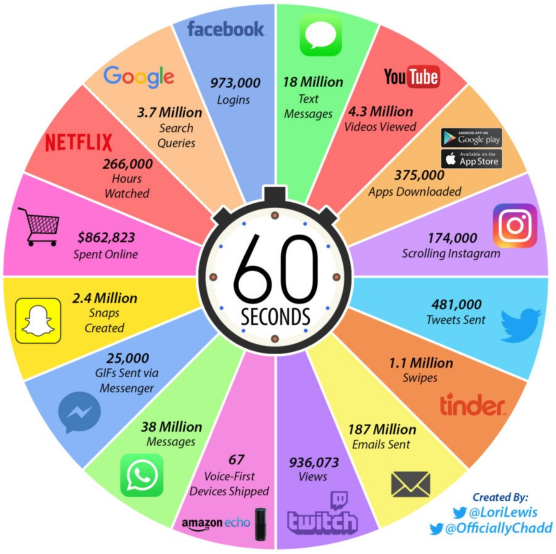


Figure 1.1: *What Happens in an Internet Minute in 2018?* [1]

New sources of information, aligned with the new technological capacities, have massively increased the number of the stored data [5], most of which is in its raw format [6]. Therefore, one of the most significant challenges today is to learn how to extract knowledge hidden in these complex raw datasets, a field known as Big Data [7].

An article in the magazine *The Economist*, in 2017, claimed that data is, in this century, what oil was in the last [8]. The statistics captured through the usage of Big Data approaches are creating new infrastructures, businesses, politics, and economics. On the other hand, it is almost impossible for humans to analyze such massive data flows in practical times. The techniques to provide computers with the ability to capture useful statistics based on raw data is known as Machine Learning [9].

A more formal definition, proposed by the regarded researcher Tom Mitchell, states that: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [10]. In simple terms, a program learned when it could improve the performance of a task by using previous experience.

Sometimes, to understand the large data sets is common to split the data into homogeneous groups that have high similarity between its samples. In this context, classification, or clustering, is a handy statistical tool for information extraction. As a supervised learning technique, the goal of classification is to construct a classification rule based on a training set where both characteristics and class labels are given. Once obtained, the classification rule can then be used for class prediction of new objects whose characteristics are available. Unfortunately, the similarities based on which the classification rules are constructed are subjective, leading to several different definitions. At the same time, there are different methodologies in the vast literature of the field [11, 12, 13].

Depending on how the data is presented, two main concepts can be derived: the supervised; and the unsupervised clustering [14]. Supervised clustering refers to the case where data is labeled, and the expected outcome of the clustering method is previously known. Hence, supervised data is usually used to train the clustering method, detecting the patterns, and then labeling a newly encountered, yet unlabeled, pattern [15]. The Support Vector Machines (SVM) [16] and Random decision forests [17] are two examples of popular algorithms in supervised clustering.

When the data is not labeled, the classification is known as unsupervised data learning. The input of the clustering method is the unlabeled data, and the methodology behind it needs to find a similarity measure to split the data into groups. Each cluster found must be meaningful and have similar characteristics based on a given metric. The k-means [18] and Affinity Propagation [19] are two examples of popular algorithms in unsupervised clustering.

The subject of a clustering method is a dataset, a collection of samples where each has features. This features, in turn, can be encoded into different data types. In order to apply most methods, these features are converted to numbers following a consistent procedure.

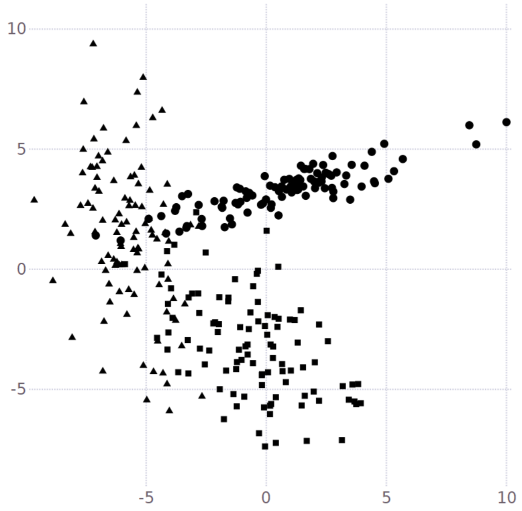
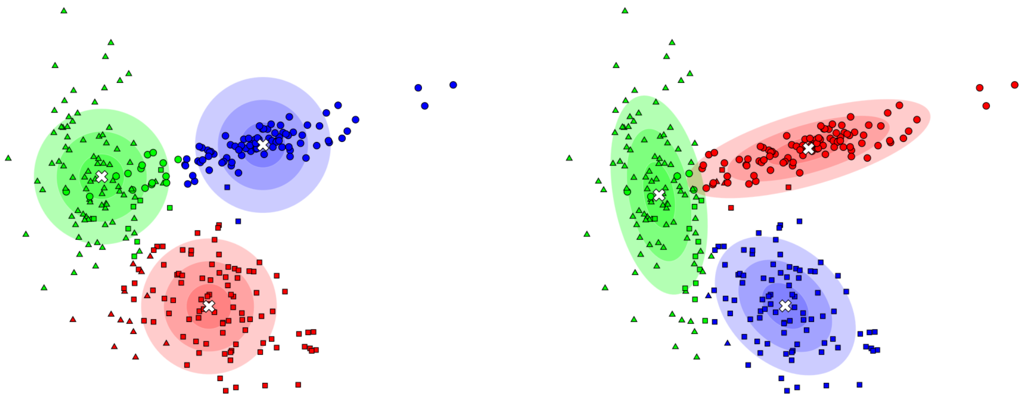


Figure 1.2: Visualization of a two-dimensional dataset. The shapes (square, triangle and circle) represent the real outcomes.

In case the dataset has two features, it can be represented as a two-dimensional figure, as can be seen in Figure 1.2, where each point represents a sample, and its position on the chart is given by the numerical values of its features, the shapes of the points indicating which class they belong. While unsupervised algorithms are used when this information is not available, datasets containing the real outcomes can be used to validate such algorithms.

The present work studies the class of clustering known as center-based algorithms. It requires a fixed number of clusters specified *a priori*. For instance, taking the dataset of Figure 1.2 as an example, three clusters could be devised. The intention is to identify such clusters by its middle point or *centroid*. Even though the widely used method in literature is the k-means [18], a spherical metric algorithm, real data sets with a perfectly spherical shape are not usual. Spheres, however, are particular instances of more general shape, the ellipsis. Therefore, these shapes would be capable of better-approximating clusters with more general shapes, resulting in an improved classification. The contrast between shapes is a central issue of this study and Figure 1.3 depicts the results of implemented algorithms using either metrics: spherical (Figure 1.3(a)) and ellipsoidal (Figure 1.3(b)).



1.3(a): k-means.

1.3(b): Ellipsoidal method proposed.

Figure 1.3: Result of the the actual implementations of the methods, analyzing the two possible shapes: spherical and ellipsoidal.

Since these problems are complex, to make them more accessible, the algorithms usually perform local searches leading to a local (often non-global) optimum. A better search in the solution space regarding results can be done with meta-heuristics [20], which will henceforth be referred to as global optimization efforts. This work adapts two meta-heuristics already applied in spherical methods to improve the results of ellipsoidal methods: the Random Swap [21]; and the Hybrid Genetic Algorithm [22].

In addition to the shapes and the search effort, a third definition of the analyzed methods is of concern: soft or hard assignment. The hard assignment involves the assignment of a sample to a single cluster. Soft assignment, on the other hand, assigns probabilities of each sample belonging to either cluster.

The goal of this work is to review the concepts and techniques used in the clustering analysis in order to propose ellipsoidal clustering methodologies. We present a mathematical formulation of the hard assignments considering the ellipsoidal shapes problem, from now on referred to as ellipsoidal minimum sum-of-squares clustering (EMSSC). Furthermore, ellipsoidal heuristics, with both hard and soft assignment, are proposed and compared. The proposed algorithms adapted the two described meta-heuristics.

Since the identification of the clusters shapes uses covariance matrices, when these are poorly estimated, the algorithms perform likewise. For this reason, a concept called regularization, which can improve covariance matrices estimation, is also tested and compared so that better results in ellipsoidal algorithms can be obtained. Finally, an analysis of the circumstances where ellipsoidal clustering algorithms supersede spherical ones is performed considering all the fundamental concepts of classification. Several circumstances involving these two shapes are also analyzed, such as the type of assignment and the effort of a global optimization search.

All the algorithms were implemented purely on Julia [23] and are available in packages.

This document is structured as follows: Chapter 2 presents the general problem of clustering and three mathematical formulations; in Chapter 3 we discuss the contributions in the literature and some fundamentals concepts necessary to understand the proposed methodologies; Chapter 4 begins by reviewing two ellipsoidal heuristics. Along with this heuristics, the proposed methodologies gather some regularization concepts of the covariance matrices and two meta-heuristics; Chapter 5 describes the benchmarks tests with artificially generated and some datasets available public. There is a comparison between the methods, their advantages, and disadvantages; Chapter 6 holds the conclusions and some future works.

2
Problem Statement

In this work, we define a dataset X as a collection of points $X = \{\vec{x}_1, ..., \vec{x}_n\}$. Each point $\vec{x}_i = \{x_i^{(1)}, ..., x_i^{(d)}\}$ is a sample and the dimension d denote the number of features of this dataset. The representation of the entire dataset is a matrix, $n \times d$, of the number of points and the number of features. The points of the dataset do not have its group label given a priori; the only known information is the number of groups, k. The method has to learn a way to infer a function that describes the structure of unlabeled data, and this is known as unsupervised learning. The general problem of grouping data, into k sets according to one or more similarity criteria is named clustering, and each group is a cluster. Partitioning data into clusters is a powerful tool in various fields of study [24], including pattern recognition, computer graphics, and bioinformatics. It is important to notice that clustering analysis is not a specific algorithm, but a general problem.

There are two ways to classify the points: hard and soft assignments. The former estimates the probability of each point belonging to each cluster. The latter assigns a classification to one and only one group without producing the probability.

Due to subjectivity about what a cluster is, there are several formulations in the literature, and each one can lead to a different solution [25]. However, the main task of the algorithms remains the same: gather several samples into clusters. The elements belonging to some cluster must have high similarity, and those who belong to different clusters should have low similarity. The very first challenge in this problem is to define a similarity measure to compare the objects. Among the premises of similarity, some stands out.

- *connectivity-based*: seeks to build a hierarchy of clusters, using some distance notion of relating points that are close to each other. The divisive strategy starts with all the points belonging to a single cluster, and new clusters are created by diving the existing ones. By contrast, the agglomerative strategy starts with all the points belonging to its cluster, and pairs of clusters merged into a new one.
- *density-based clustering*: works by assuming clusters in areas of high density of points. The most known density-based algorithm is the Density-

Based Spatial Clustering of Applications with Noise, or simply *DBSCAN* [26]. Introduced by *Ester et al.* in 1996, the paper was awarded the 2014 SIGKDD Test of Time for having a significant impact in the data mining research community beyond the last decade [27].

- *center-based*: has a point (not necessarily a member of the dataset) representing each cluster and assigns each point to the closest cluster center based on distance criteria. This premise is very popular due to the widespread k-means algorithm [28].
- *distribution-based*: instead of using a distance function, it fits a distribution model to define a probability to a point belonging to a cluster. These methods usually take advantage of the correlation between the features. The most popular one assumes Gaussian distributions and is known as Gaussian Mixture Models clustering.

All these premises and principles reveal the multiple definitions which a clustering problem can have. In this chapter, there is the revision of two classical mathematical formulations of the problem, which are the minimum sum of squares clustering and maximum likelihood estimation. In the end, we describe a mathematical optimization formulation of a known problem: ellipsoidal clustering with hard assignments, described in this work as the ellipsoidal minimum sum of squares clustering.

2.1 Minimum Sum of Squares Clustering Problem

There are several ways to define a clustering problem, one of the most common is the minimum sum of squares clustering (MSSC) problem [29]. The problem is to assign a single group to each point, and it is also known as the hard assignment problem. Since the problem is center-based, it defines k centers $\mu_1, ..., \mu_k$, and the objective is to minimize the sum of squared distances between the samples to their respective cluster center. The MSSC is a widely studied problem, and several algorithms to solve it are proposed in the literature [30, 31, 32].

Let S_i be a subset i of of X and $S = \{S_1, ..., S_k\}$ denote all the k partitions of X. The subsets represent the clusters, and so, they are disjoint (2-1) and every sample belongs to a subset (2-2).

$$S_i \cap S_j = \emptyset \quad \forall i, j \in \{1, ..., k\} \tag{2-1}$$

$$\bigcup_{i=1}^{k} S_i = X \tag{2-2}$$

Now, we detail an optimization problem that seeks to cluster a dataset using a hard assignment: Equation (2-3) describes the binary variable $w_{i,j}$ which denote that each point can only belong to exactly one cluster.

$$w_{i,j} = \begin{cases} 1 & \text{if the point } x_i \text{ belongs to the cluster } S_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2-3}$$

Given $M = \{\vec{\mu}_1, ..., \vec{\mu}_k\}$, Equation (2-4) describes the objective function of MSSC problem, which is the cost of the k subsets that minimizes the sum of squares of the i-th point to the center $\vec{\mu}_j$ of his respective cluster j in a subset S_j.

$$\underset{w,M}{\text{minimize}} \quad \sum_{i=1}^{n}\sum_{j=1}^{k} w_{i,j} \|\vec{x}_i - \vec{\mu}_j\|^2 \tag{2-4}$$

subject to

$$\sum_{j=1}^{k} w_{i,j} = 1, \qquad i \in \{1, ..., n\} \tag{2-5}$$

$$w_{i,j} \in \{0,1\}, \qquad i \in \{1, ..., n\}, j \in \{1, ..., k\} \tag{2-6}$$

Constraint (2-5) implies that each point belongs to one and only one cluster. The objective function (2-4) has a product of two decision variables, and therefore it is nonlinear and not convex. Due to the non-linearity of the problem, classical techniques of optimization have difficulty trying to solve it, taking a very long computation time. There is an alternative formulation of the problem using a big M constant: the distance between a sample $\vec{x_i}$ and a cluster center $\vec{\mu_j}$, defined in the objective function (2-4), is now stored in a variable d_i. So, a big M constant makes Constraint (2-9) always feasible for the points that do not belong to that cluster. Let $\mathcal{I} = \{1, ..., n\}$ and $\mathcal{J} = \{1, ..., k\}$:

$$\underset{w,M,\vec{(d)}}{\text{minimize}} \quad \sum_{i=1}^{n} d_i \tag{2-7}$$

subject to

$$\sum_{j=1}^{k} w_{i,j} = 1, \qquad i \in \mathcal{I} \tag{2-8}$$

$$d_i \geq \|\vec{x}_i - \vec{\mu}_j\|^2 - M\left(1 - w_{i,j}\right), \qquad i \in \mathcal{I}, j \in \mathcal{J} \tag{2-9}$$

$$d_i \geq 0, \qquad i \in \mathcal{I} \tag{2-10}$$

$$w_{i,j} \in \{0,1\}, \qquad i \in \mathcal{I}, j \in \mathcal{J} \tag{2-11}$$

Several studies about this problem were made, producing an extensive literature [33, 34, 35]. It is important to mention that the MSSC is an NP-Hard problem [36]. However, if both k and d are fixed, the problem can be solved exactly in $\mathcal{O}(n^{dk+1})$ time [37]. Among all the heuristics already developed for this problem, the most popular is the k-means [38].

2.2 Maximum likelihood Estimation

Besides minimizing the sum of squares between the points and their respective cluster, there is another traditional methodology that uses a different objective function, called maximum likelihood estimator (MLE) [39]. This section details the formulation of the clustering problem using MLE.

The first fundamental concept for defining the likelihood function is the probability distribution function: a probability function of a continuous random variable, in statistics, is a function that describes the odds of a random outcome, given a model parameter. On the other hand, a likelihood function describes the plausibility of the model parameters, given a specific observed point.

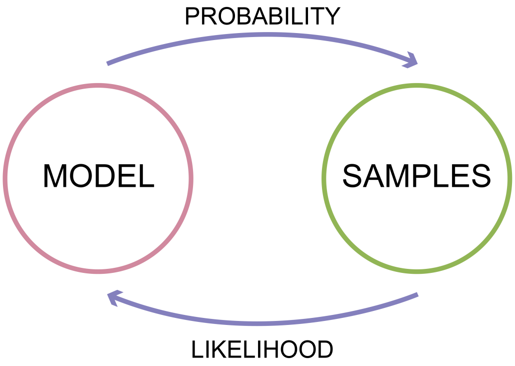


Figure 2.1: Probability and likelihood relationship.

Figure 2.1 shows the relationship between the probability and the likelihood functions: if we have a statistical model and a probability distribution that describes that model, then that probability function can be used to generate the samples of the model. On the other hand, instead of generating random samples from an existing model, the required is the opposite: find a model that explains the empirical data, in this case, we use a likelihood function. The use of these two functions depends how is the desired approach, after all, probability and likelihood use the distribution function fo a random variable.

Assuming that the sample set $X = \{x_1, ..., x_n\}$ of n points in $\mathbb{R}^d$ are independent and identically distributed random variables (IID) where each sample x_i is extracted from a probability distribution function $f(x_i|\theta_i)$, parameterized by θ, let the probability be defined:

$$P(X|\theta) = f(x_1, ..., x_n|\theta) = f(x_1|\theta) \cdot \dots \cdot f(x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta) \tag{2-12}$$

Since the samples are independent, Equation (2-12) describes the definition of the joint probability distribution function, which is the product of indexed terms. The likelihood function $\mathcal{L}$ of the θ parameters given the data is the same as the probability of the data given the *theta* parameters, shown in Equation (2-13).

$$\mathcal{L}(\theta|X) = P(X|\theta) \tag{2-13}$$

The estimation of the parameters in a statistical model usually uses likelihood functions. Equation (2-14) describes that the best parameter values that fit the given data are found such that they maximize the likelihood estimation.

$$\underset{\theta}{\text{maximize}} \quad \prod_{i=1}^{n} f(x_i|\theta) \tag{2-14}$$

Since Equation (2-14) is hard to differentiate, for computational convenience, it is common to rely on the log of the likelihood function. The log is a monotonic increasing, and the desired point is the maximum value. So the maximum of the likelihood and log-likelihood occurs at the same parameters, seen in Equation (2-15).

$$\underset{\theta}{\text{maximize}} \quad \prod_{i=1}^{n} f(x_i|\theta) \quad \rightarrow \quad \underset{\theta}{\text{maximize}} \quad \sum_{i=1}^{n} \log f(x_i|\theta) \tag{2-15}$$

However, if there is more than one subpopulation within an overall population, the data can have more than one statistical model, and then the Mixture of Models concept can be applied. This concept uses a weighted sum over the distributions to mixture these subpopulations. It can be, for example, several densities or statistical models. Let $M = \{P_1, ..., P_k\}$ be a set of k distributions and π_i be the mixing weights, where $\pi_j \geq 0$ and $\sum_{j=1}^{k} \pi_j = 1$. Equation (2-16) is a convex combination of the components and Figure 2.2 shows an example of a mixture of 3 Gaussian distributions.

$$\sum_{j=1}^{k} \pi_j P(X|\theta_j) \tag{2-16}$$

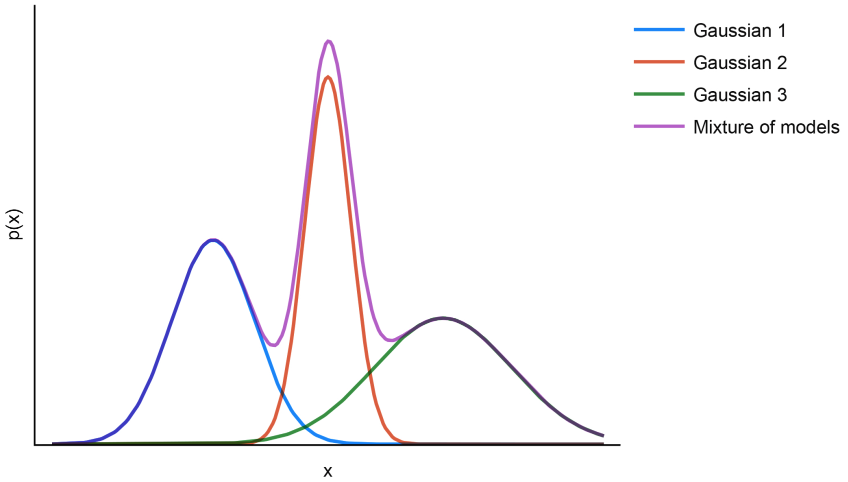


Figure 2.2: Example of a Mixture of Models of Gaussian distributions 1, 2 and 3.

Let $\Theta = \{\theta_1, ..., \theta_k\}$ and $\Pi = \{\pi_1, ..., \pi_k\}$, the formulation below combines the mixture of models and the maximum likelihood estimator, resulting in k parameters θ and π.

$$\underset{\Theta,\Pi}{\text{maximize}} \quad \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \pi_j f_j(x_i|\theta_j) \right) \tag{2-17}$$

subject to

$$\sum_{j=1}^{k} \pi_j = 1 \tag{2-18}$$

$$\pi_j \geq 0, \qquad j \in \{1, ..., k\} \tag{2-19}$$

When there is just one distribution, analytical expressions found the parameters of MLE directly. However, when there is more than one distribution, the problem becomes not trivial to solve analytically. In 1977, Dempster *et al.* [40] proposed the Expectation Maximization (EM), an iterative method to find the MLE with several distributions. The methodology modifies the problem by adding a new random variable: the classes Y, that is which distribution each sample comes from, analogously, the clusters. The addition of a new random variable can make the problem more difficult to solve directly. However, the EM takes advantage of the new variable to split the algorithm into two steps, and make an iterative procedure to reach local optima of the MLE [41]. Since the value of Y is unknown, we use the EM to maximize the expected value of the likelihood function, describe in Equation (2-20).

$$\underset{\theta}{\text{maximize}} \quad \underset{Y|X,\theta}{\mathbb{E}}[\mathcal{L}(\theta|X,Y)] \tag{2-20}$$

The first step is known as Expectation: it fixes the parameters θ and finds the expected value of the likelihood function. Equation (2-21) details Q, which is the function of likelihood, where $\theta^{(t-1)}$ is the current parameters and θ is the new parameters that we want to optimize.

$$Q(\theta,\theta^{(t-1)}) = \mathbb{E}\left[\log P(X,Y|\theta) \mid X,\theta^{(t-1)}\right] \tag{2-21}$$

Equation (2-22) describes the Maximization step, which maximizes the expectation evaluated in the previous step.

$$\theta^{(t)} = \underset{\theta}{\text{argmax}} Q(\theta,\theta^{(t-1)}) \tag{2-22}$$

The method iteratively attempts to estimate the distributions parameters, θ, and the cluster indexes, Y. Both steps improve the likelihood and converge to local optimum [42]. The complete explanation of the EM with all the proofs and details are in Bilmes *et al.* work [43].

A particular case of EM is when the distributions are Multivariate Gaussians distributions, the problem is known as Gaussian Mixture Models (GMM). Each Gaussian distributions have the mean vector μ and the covariance matrix Σ as parameters. The interpretation of μ and Σ in clustering context is, respectively, the cluster center and how the points spread in the space regarding an ellipsoidal shape.

The whole concept applies to the clustering problem, where the observations are the dataset, and the k models are the clusters [44]. Unlike MSSC, the MLE does not use hard assignments, it uses soft, which describes probabilities of each point belonging to each cluster.

2.3
Ellipsoidal Minimum Sum of Squares Clustering Problem

Both heuristics, k-means and GMM, are very popular in the literature [45, 46]. The former produces hard assignments to the samples considering a spherical metric. The latter treats the clusters as ellipsis while doing soft assignments. However, the mixture of these two, ellipsoidal shapes with hard assignments problem, are rarely used in practice.

The idea is using the same objective function and the hard assignments of the MSSC problem, but remodeling the restrictions to distort the distances using the ellipse's eccentricity. Positive definite matrices can be used to define these shapes. There are some descriptions of this problem in the literature [47, 48, 49], but in this work, we formulate as a mathematical optimization problem. We named this formulation as the ellipsoidal minimum sum of squares clustering (EMSSC).

Throughout this work, we reference the positive-definite matrix, which defines the ellipsoidal shape, as the shape matrix or Σ. Positive definite matrices have interesting characteristics: it has only positive eigenvalues and a unique Cholesky decomposition [50].

The objective function is the same as the MSSC. The difference is that the Euclidean norm is replaced by a metric called Mahalanobis distance norm, which distorts the distance using a positive definite matrix. Each cluster has an associated matrix and distorts the distance from each sample to its center. Equation (2-23) describes the Mahalanobis distance.

$$d_M(\vec{a}, \vec{b}, \Sigma) = \sqrt{\left(\vec{a} - \vec{b}\right)^\top \Sigma^{-1} \left(\vec{a} - \vec{b}\right)} \tag{2-23}$$

Let $\vec{\lambda} = \{\lambda_1, ..., \lambda_d\}$ be the eigenvalues of the matrix Σ, Equation (2-24) describes the determinant of it, which is the product of the eigenvalues. This determinant defines, in a certain sense, the volume of the ellipsoid

$$\det \Sigma = \prod_{i=1}^{d} \lambda_i \tag{2-24}$$

The shape matrices can not assume any value. If this happens, the eigenvalues of one matrix will tend to infinity, the distances to that cluster would be zero, and then all samples would be assigned to that cluster. To solve this issue, the determinant of the covariance matrix needs to have a fixed value.

Let P represent the inverse of the positive definite matrix Σ^{-1} and $\vec{P} = \{P_1, ..., P_k\}$. To precisely define the hard ellipsoidal problem, let $X = \{\vec{x}_1, ..., \vec{x}_n\}$ denote the set of n points in $\mathrm{I\!R}^d$, $M = \{\vec{\mu}_1, ..., \vec{\mu}_k\}$, $\mathcal{I} = \{1, ..., n\}$ and $\mathcal{J} = \{1, ..., k\}$. The problem below is the EMSSC mathematical formulation.

$$\underset{w,M,\vec{d},\vec{P}}{\text{minimize}} \quad \sum_{i=1}^{n} d_i \tag{2-25}$$

subject to

$$\sum_{j=1}^{k} w_{i,j} = 1, \qquad i \in \mathcal{I} \tag{2-26}$$

$$d_i \geq d_M(\vec{x_i}, \vec{\mu_j}, P_j)^2 - M\left(1 - w_{i,j}\right), \qquad i \in \mathcal{I}, j \in \mathcal{J} \tag{2-27}$$

$$d_i \geq 0, \qquad i \in \mathcal{I} \tag{2-28}$$

$$P_j \succeq 0, \qquad j \in \mathcal{J} \tag{2-29}$$

$$\det P_j = 1, \qquad j \in \mathcal{J} \tag{2-30}$$

$$w_{i,j} \in \{0, 1\}, \qquad i \in \mathcal{I}, j \in \mathcal{J} \tag{2-31}$$

Constraint (2-26) is analogous to Constraint (2-8) from MSSC; the variable $\vec{\mu}_j$ remains the center of its respective cluster j; the variable d_i is the cost of the sample i, that is the distance from the sample to the nearest cluster. If a point x_i does not belongs to the cluster j the Constraint (2-27) becomes slack. Otherwise, d_i is the Mahalanobis distance of the point i to the cluster j modified by the matrix P_j. Constraint (2-29) describes that the P must be positive definite to keep the metric properties [51]. Finally, Constraint (2-30) guarantees that the determinant of the covariance matrix is different from 0, which prevents any cluster from disappearing.

2.4
Research Questions

Chapter 2 describes three mathematical formulations for the clustering problem: MSSC, MLE, and EMSSC. The scalability and simplicity of the k-means algorithm made the MSSC one of the most used formulations to solve the unsupervised clustering problem. However, the ease of implementing this heuristic comes with the limitation of using spherical shapes. The GMM clustering, which has the MLE as the objective function, also has an essential role in literature. It is the state of the art when it is desired to use ellipsoidal shapes with soft assignments. The third and last formulation is the EMSSC, which have a few works in the literature, but in this work, we formulate as a mathematical optimization problem. This formulation makes hard assignments and generalizes the MSSC problem to consider ellipsoidal shapes.

Among the three formulations, two aspects stand out: the format of the clusters and the type of assignment. From these aspects, we raise two questions.

The first is under what conditions an algorithm considering ellipsoidal shapes is more advantageous than one that only considers spherical. The other question is about the type of assignment. A careful comparison between the formulations of these problems still lacks in the literature. This work highlights the qualities and defects of each one, with several and extensive benchmark tests with both artificial and real datasets.

3
Fundamentals and Literature Review

The three formulations discussed in the last chapter deal with two concepts of the clustering problem: type of classification (hard and soft assignment) and the shape of clusters (spherical and ellipsoidal). However, ellipsoidal problems have more degrees of freedom because of the estimation of shape matrices. Having more parameters can make the problem more difficult to solve.

A single local search to find a local optimum, which is a solution where no improving neighbors are available, may not be good enough, especially when the solution space is huge. We want to design an algorithm that explores the search space in a way to provide a sufficiently good solution for the problem. A meta-heuristic is a high-level algorithm aimed at implementing a solution space search strategy, possibly mixing different heuristics, in order to find better quality solution hopefully quasi-optimal. Two meta-heuristics are tailored to improve the local search: a random swap and a hybrid genetic algorithm. Therefore, we discuss the global optimization effort along with the shapes and assignment analysis throughout this chapter. In the last section of this chapter, we revise and summarize the methods.

3.1
Hard x Soft Assignment

There are two possible classification types: hard and soft assignments. This choice affects the assignment type of samples. When a classifier is soft, it estimates the probability $\gamma_{i,j}$ an observation i has to belong to cluster j. The probabilistic model gives the power to express uncertainty about the assignment of each point. Equation (3-1) describes the probabilities $\vec{\gamma}_i$ of the point i.

$$\vec{\gamma}_i = \{\gamma_{i,1}, ..., \gamma_{i,k}\}, \qquad \gamma_{i,j} \geq 0 \quad \forall\ j \in \{1, ..., k\} \quad \text{and} \quad \sum_{j=1}^{k} \gamma_{i,j} = 1 \qquad (3\text{-}1)$$

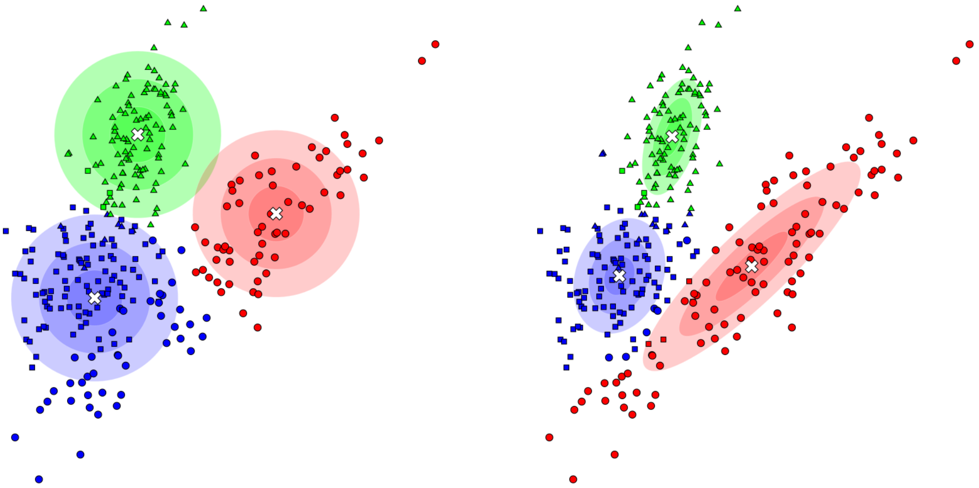
The conventional hard classifier targets the classification without producing the probability estimation, i.e., each object belongs to one and only one cluster. According to Liu *et al.* [52], this classifier tends to work better on datasets that are well separable. On the other hand, the soft classification tends to have better results when the underlying conditional class probability function is relatively smooth. There are some hard versus soft comparisons in literature, but when it comes to ellipsoidal clusters, there are almost none. This lack in the literature is one of the analyzed issues in this work.

Depending on the application or method, one may be more appropriate than the other. During this work, we observe the situations that one is worth more than another and evaluate the accuracy of each assignment type with different methods.

3.2 Spherical x Ellipsoidal Clustering

The use of spherical shapes, as in the MSSC problem, is widespread in clustering algorithms. The k-means heuristic has a straightforward implementation and is relatively cheap to compute. However, the formulation assumes that the points are spread as spherical distributions, even though not all datasets have the dispersion of points that way.

Since real-world datasets with a perfectly spherical shape are rare, generalizing this concept is essential. This generalization leads to another center-based possible shape, the ellipsoidal. Figure 3.1 illustrates the actual implementation of two algorithms: a spherical and an ellipsoidal.



3.1(a): k-means.

3.1(b): Ellipsoidal method proposed.

Figure 3.1: Result of the the actual implementations of the methods, k-means has the limitation of only considering spherical shapes, while ellipsoidal methods can be more adaptive.

The shape matrix Σ defines the ellipsoidal shape of the cluster: while its eigenvalues explain the spread of data in directions parallel to axes of the feature space, its eigenvectors represent a rotation matrix of the ellipsoid [53]. Whereas spheres are particular cases of ellipsoids, this flexibility can increase the space of solutions and perhaps improve the results.

The most common algorithm that has ellipsoidal shapes is the GMM clustering method [54], which uses Multivariate Gaussian distributions and MLE to identify the components of a probability density function that generated the data. The parameters that need to be estimated are the average vector $\vec{\mu}$ and the shape matrix Σ of each cluster. The $\vec{\mu}$ represents the centroid of each cluster, and the matrix Σ correlates the samples, that contributes to defining the distorted distances. Since the GMM assigns the probability of each point belonging to each cluster, then it is a soft assignment method.

Algorithms that consider ellipsoidal shapes and have a hard assignment are few in literature [47, 48] and practice, and this is the main difference between the GMM and the EMSSC problem. The use of ellipsoidal shape structure to the clusters gives more flexibility to the classification, and perhaps better results can be obtained. This flexibility leads to a fundamental research question in this work: to understand what circumstances ellipsoidal shapes are more beneficial than spherical shapes.

As previously seen, there are two basic requirements to define the structure of an ellipsoidal clustering: a metric that distorts the notion of distance and a shape matrix, which captures the shape of a distribution. In the next subsection, we revise the concept of Metric.

3.2.1 Metrics

There are several possible similarity criteria in clustering, which is a real-valued function that quantifies the similarity between the two observations. These similarity criteria can be a distance function. A distance function that satisfies the metric properties is known as a metric.

Equation (3-2) describes that a metric, on a set X, is a non-negative function between two points $d : X \times X \to [0, \infty)$ while satisfies the triangle inequality (3-3) and the property of symmetry (3-4).

$$d(\vec{x},\vec{y}) \geq 0 \tag{3-2}$$

$$d(\vec{x},\vec{z}) \leq d(\vec{x},\vec{y}) + d(\vec{y},\vec{z}) \tag{3-3}$$

$$d(\vec{x},\vec{y}) = d(\vec{y},\vec{x}) \tag{3-4}$$

$$d(\vec{x},\vec{y}) = 0 \Leftrightarrow \vec{x} = \vec{y} \tag{3-5}$$

The last property, Equation (3-5), means that if the points are the same, the distance between the points is zero. This property is the identity of indiscernibles, and its presence is what distinguishes a metric from a pseudometric.

The commonly-used Euclidean distance has an intuitive appeal and assumes that each point is equally important and disassociated from the others. Equation (3-6) gives the Euclidean distance between two points in a d-dimensional space. Its computational complexity is $\mathcal{O}(n)$.

$$d_E(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{3-6}$$

Since the Euclidean distance does not correlate the points, it would be essential to use a distance that considers the variance and the correlation between the variables. In 1936, Mahalanobis [55] introduced the Mahalanobis distance, which is a generalization of the Euclidean distance that considers a covariance matrix Σ that distorts the space and considers different weights to the points according to its distortion.

This distortion considers the covariance between the variables and is direction-sensitive, in other words, allows that each variances direction can have different sizes. Equation (3-7) defines the Mahalanobis distances, which weights dimensions according to their covariances.

$$d_M(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^\top \Sigma^{-1} (\vec{x}-\vec{y})} \tag{3-7}$$

Notice that when the Σ is the identity matrix the Mahalanobis distance reduces to the Euclidean distance. If Σ^{-1} is already given, the computational complexity of the Mahalanobis distance is $\mathcal{O}(n^2)$, otherwise the cost of the inversion of a matrix using the naive method, Gaussian elimination, is $\mathcal{O}(n^3)$.

3.2.2 Covariance Matrices

The covariance matrices are essential to estimate the shape matrices used in the Mahalanobis distance, and along the algorithms explained in this work, they approximate the description of the distribution of points in space. This subsection details how the estimation works.

Equation (3-8) reveals the matrices, which apply the concept of covariance of two random variables X and Y is the expected product of their deviations from their individual expected values. The variance is the expected value of the squared deviation from the expected value of a random variable X (3-9). Equation (3-10) shows that the variance is the covariance of a random variable with itself.

$$\mathrm{Cov}(\vec{X},\vec{Y}) = \mathbf{E}\left[\left(\vec{X} - \mathbf{E}\left[\vec{X}\right]\right)\left(\vec{Y} - \mathbf{E}\left[\vec{Y}\right]\right)\right] \tag{3-8}$$

$$\mathrm{Var}(\vec{X}) = \mathbf{E}\left[\left(\vec{X} - \mathbf{E}\left[\vec{X}\right]\right)^2\right] \tag{3-9}$$

$$\mathrm{Var}(\vec{X}) = \mathrm{Cov}(\vec{X},\vec{X}) \tag{3-10}$$

Equation (3-12) express the covariance matrix, which is a measure of how much the data spread across the feature space. The element in the i-th row and j-th column, of that matrix shows the covariance between the i-th and j-th elements of a random vector. When $i = j$, that is a diagonal entry, the element represents the variance of that variable. The expected value of an element can be seen simply as the mean (3-11).

$$\vec{\mu} = \mathbf{E}\left[\vec{X}\right] \tag{3-11}$$

$$\Sigma = \begin{bmatrix} \mathbf{E}\left[\left(X^{(1)} - \mu^{(1)}\right)\left(X^{(1)} - \mu^{(1)}\right)\right] & \cdots & \mathbf{E}\left[\left(X^{(1)} - \mu^{(1)}\right)\left(X^{(d)} - \mu^{(d)}\right)\right] \\ \vdots & \ddots & \vdots \\ \mathbf{E}\left[\left(X^{(d)} - \mu^{(d)}\right)\left(X^{(1)} - \mu^{(1)}\right)\right] & \cdots & \mathbf{E}\left[\left(X^{(d)} - \mu^{(d)}\right)\left(X^{(d)} - \mu^{(d)}\right)\right] \end{bmatrix} \tag{3-12}$$

Equation (3-13) and Equation (3-14) are the empirical formulas of variance and covariance, respectively. Consider n points, each point is d-dimensional, and μ is the mean of those points.

$$\vec{\mu} = \frac{1}{n} \cdot \sum_{i=1}^{n} \vec{x_i} \tag{3-13}$$

$$\Sigma = \frac{1}{n} \cdot \begin{bmatrix} \sum_{i=1}^{n}\left(x_i^{(1)} - \mu^{(1)}\right)\left(x_i^{(1)} - \mu^{(1)}\right) & \cdots & \sum_{i=1}^{n}\left(x_i^{(1)} - \mu^{(1)}\right)\left(x_i^{(d)} - \mu^{(d)}\right) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n}\left(x_i^{(d)} - \mu^{(d)}\right)\left(x_i^{(1)} - \mu^{(1)}\right) & \cdots & \sum_{i=1}^{n}\left(x_i^{(d)} - \mu^{(d)}\right)\left(x_i^{(d)} - \mu^{(d)}\right) \end{bmatrix} \tag{3-14}$$

The covariance matrix is positive semi-definite, meaning that Σ is symmetric and $z^{\top}\Sigma z \geq 0$ for every non-zero column vector $z \in \mathbb{R}^{d}$. The inverse of a covariance matrix Σ^{-1} is usually known as the precision matrix. The complexity of the covariance matrix estimation is $\mathcal{O}(nd^2)$.

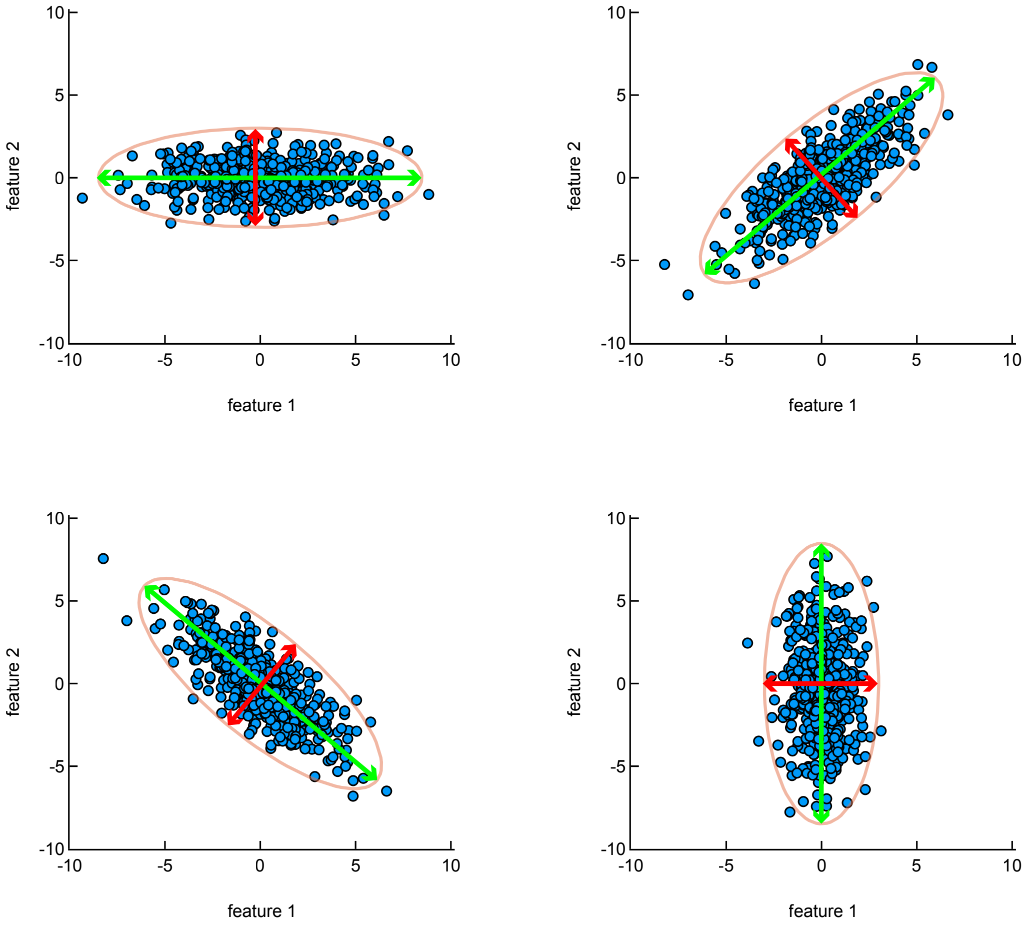


Figure 3.2: Different covariance matrices generated with the same eigenvalues $\lambda_1 = 8$ and $\lambda_2 = 1$.

Figure 3.2 shows different dataset generated by four different covariance matrices. Although the datasets are different, the eigenvalues of the matrices on this example are the same, $\lambda_1 = 8$ and $\lambda_2 = 1$. The dimensions of the ellipse are the same, the eigenvectors that rotate them.

Covariance matrices generated with the empirical estimator and with insufficient observations are usually ill-conditioned, and when it happens, the matrix can be singular and not invertible. The matrices can become flat with an empty interior and degrade the methods that use it.

Due to the increasing number of parameters in the ellipsoidal methods, the number of observations to estimate them need to be sufficiently large. If they are not, the model can be over-fitted [56]. There is a method for this specific problem called regularization, which tries to avoid these imprecise and unstable problems with the empirical covariance estimator.

3.2.2.1 Regularized Covariance Matrices

A technique known as regularization (or shrinkage) is designed to prevent over-fitting of statistical models [57]. In the covariance matrices context, the regularization is used to improve the estimator by reducing the condition number, which is the ratio between the largest and the smallest eigenvalue. The regularization, in turn, reduces the size of the hypothesis space and may allow for better generalization.

The method computes the convex linear combination between the matrix Σ and a scaled identity matrix [58]. When the number of samples is sufficiently larger than the number of features, no regularization is necessary. However, we can not guarantee that the number of observations is sufficient. The estimation can produce ill-conditioned matrices and spoil the final results of the clustering algorithm. Equation (3-15) describes the regularization of the covariance matrix.

$$\Sigma_{\text{regularized}} = (1-\delta)\Sigma + \delta\frac{\operatorname{Tr}\Sigma}{d}I \tag{3-15}$$

The regularization constant is defined to keep the magnitude of the original matrix. This constant is the average of the diagonal members of Σ, given by $\frac{\operatorname{Tr}\Sigma}{d}$. The convex linear combination of the original matrix and the identity matrix with the regularization constant on the diagonal maintains the order of magnitude of the original matrix and regularizes it.

The empirical covariance matrix can be seen as a random variable, having a variance and an average. If the samples change and the estimated matrix is entirely different, the variance is large. So it is not robust to small changes in the samples. The original matrix has a small bias but a large variance for the observations. While the matrix $\frac{\operatorname{Tr}\Sigma}{d}I$ has a large bias but a small variance. The purpose of the regularization is to reduce the variance, allowing a larger bias. The parameter δ controls the trade-off between the bias and the variance.

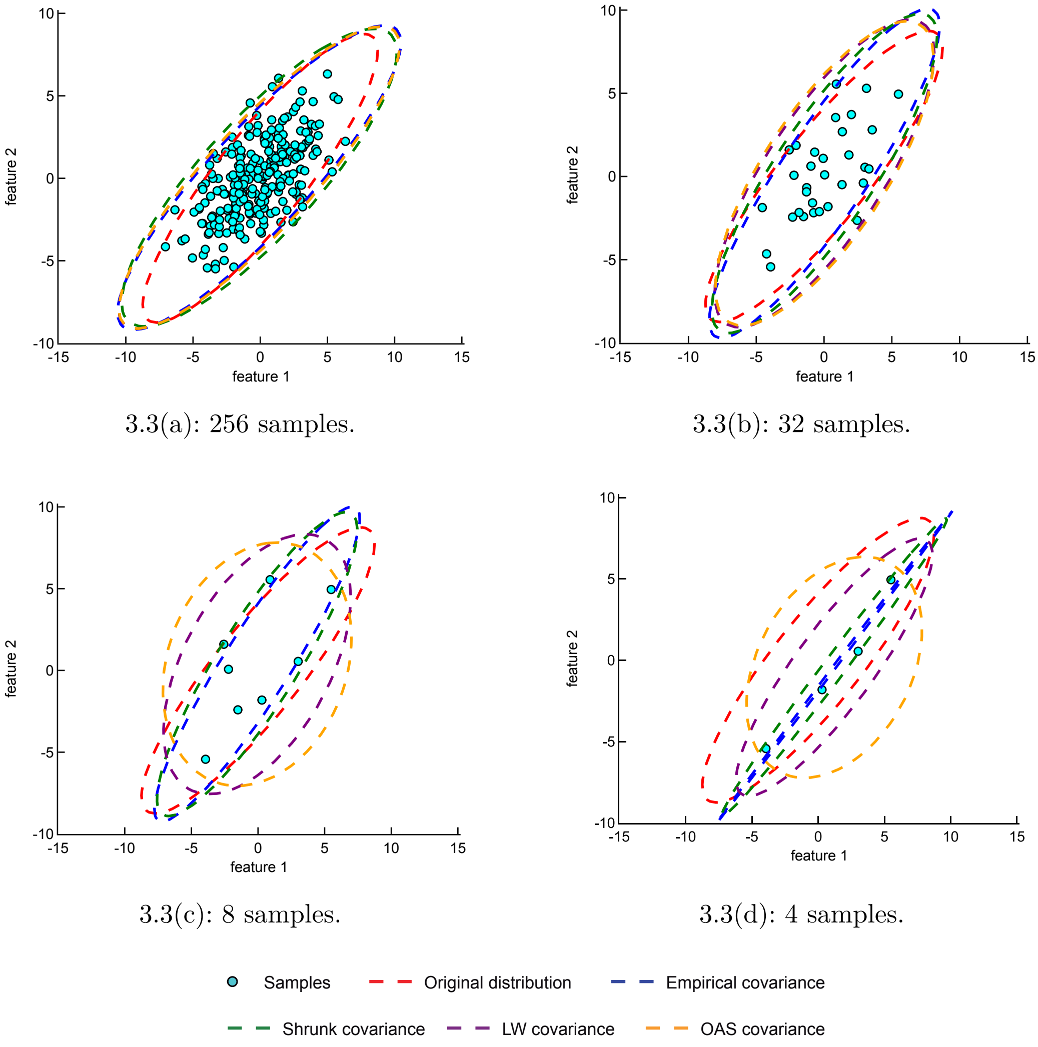
The default parameter δ value of the Shrunk methodology is 0.1, but the choice of a good value may depend on an analysis of the observations. There already exist methods in literature for choosing the δ parameters, and we choose two of them in this work.

The first method is the Ledoit-Wolf Shrinkage (LW) [58], which tries to minimize the mean squared error between the estimated and the real covariance matrix. Due to the extensive formulations, we do not place the equations in the text; they are accessible in the paper.

The second is the Oracle Approximating Shrinkage (OAS) [59], which assumes that the data came from a Gaussian distribution. The work of Chen *et al.* says that the shrinkage coefficient δ reduces, even more, the mean squared error compared to the LW regularization. Equation (3-16) is the formula that computes the δ in OAS methodology.

$$\delta_{\mathrm{OAS}} = \min\left(1, \frac{(1-2/d)\,\mathrm{Tr}\left(\Sigma^2\right) + \mathrm{Tr}^2\left(\Sigma\right)}{(n+1-2/d)\left[\mathrm{Tr}\left(\Sigma^2\right) + \mathrm{Tr}^2\left(\Sigma\right)/d\right]}\right) \qquad (3\text{-}16)$$

Figure 3.3 reveals the behavior of the covariance estimation of each method: the empirical, the Shrunk, the LW and the OAS regularizations. It is clear that when there are enough points, these regularizations are not necessary. However, when the number of points is not enough, regularization make the matrices more stable and well conditioned.



3.3(a): 256 samples.

3.3(b): 32 samples.

3.3(c): 8 samples.

3.3(d): 4 samples.

Figure 3.3: Behavior of regularized covariance estimations.

The regularization techniques present in this section are all variants of computing a convex combination between an empirically obtained covariance matrix and a scalar multiple of the identity matrix. These methods have an interpretation regarding the eigenvalues: the eigenvalues of the regularized matrices are eigenvalues of the empirically obtained matrices summed of a positive constant. Therefore, we can interpret the regularization procedure as setting lower bound on the eigenvalues of the covariance matrix.

To explain the algebraic behavior of eigenvalues in regularization we define and review some concepts: let A be a $d \times d$ covariance matrix, I be a identity matrix and $\lambda = \{\lambda_1, ..., \lambda_d\}$ be a set of eigenvalues of A, present in Equation (3-18). The λ_i is an eigenvalue of A if and only if there is an eigenvector $v \neq 0$ that satisfies $Av = \lambda v$. Equation (3-17) presents the decomposition of matrix A where D is a matrix with the eigenvalues of A in its diagonal and V is a matrix with the eigenvectors of A in its columns.

$$A = VDV^{\top} \tag{3-17}$$

$$\text{eigenvalues}(A) = \{\lambda_1, ..., \lambda_d\} \tag{3-18}$$

Let α be a scalar which multiplies the matrix A, express in Equation (3-19). Equation (3-20) shows that if α multiplies the matrix A, their eigenvalues are scaled by α too.

$$\alpha A = \alpha VDV^{\top} = V\alpha DV^{\top} \tag{3-19}$$

$$\text{eigenvalues}(\alpha A) = \{\alpha\lambda_1, ..., \alpha\lambda_d\} \tag{3-20}$$

The scaling of the eigenvalues can be shown with the characteristic polynomial too. Let c be a non-negative scalar, Equation (3-21) and Equation (3-22) define the characteristic polynomial of A and αA, respectively.

$$p_A(t) = \det\left(tI - A\right) = c\prod_{i=1}^{d}\left(t - \lambda_i\right) \tag{3-21}$$

$$p_{\alpha A}(t) = \det\left(tI - \alpha A\right) = c\prod_{i=1}^{d}\left(t - \alpha\lambda_i\right) \tag{3-22}$$

To show the translation of the eigenvevalues of A, let β be another scalar. Equation (3-23) describes the characteristic polynomial of $A + \beta I$. Equation (3-24) presents the relation between Equation (3-21) and $p_A(t - \beta)$.

$$p_{A+\beta I}(t) = \det\left(tI - A - \beta I\right) = \det\left(\left(t - \beta\right)I - A\right) = p_A(t - \beta) \tag{3-23}$$

$$p_A(t - \beta) = c\prod_{i=1}^{d}\left(t - \beta - \lambda_i\right) = c\prod_{i=1}^{d}\left(t - \left(\lambda_i + \beta\right)\right) \tag{3-24}$$

Finally, the eigenvalues of the translation of A by β are also translated by β, as shown in Equation (3-25).

$$\text{eigenvalues}(A + \beta I) = \{\lambda_1 + \beta, ..., \lambda_d + \beta\} \tag{3-25}$$

The last step is mixing both scaling and translation. The scalar α scales and β translates A. Equation (3-26) describes the characteristic polynomial and Equation (3-27) describes the relation between Equation (3-21) and $p_{\alpha A}(t-\beta)$.

$$p_{\alpha A+\beta I}(t) = \det\left(tI - \alpha A - \beta I\right) = \det\left(\left(t-\beta\right)I - \alpha A\right) = p_{\alpha A}(t-\beta) \quad (3\text{-}26)$$

$$p_{\alpha A}(t-\beta) = c\prod_{i=1}^{d}\left(t - \beta - \alpha\lambda_i\right) = c\prod_{i=1}^{d}\left(t - \left(\alpha\lambda_i + \beta\right)\right) \quad (3\text{-}27)$$

Equation (3-28) presents the eigenvalues of A scaled by α and translated by β.

$$\text{eigenvalues}(\alpha A + \beta I) = \{\alpha\lambda_1 + \beta, ..., \alpha\lambda_d + \beta\} \quad (3\text{-}28)$$

3.3 Local x Global Optimization

A continuous function f, defined on a domain X, has a local minimum at p if there is an open set I containing p that $f(p) \leq f(x)\ \forall x \in I$. The local maximum is analogous, however the point is where $f(p) \geq f(x)\ \forall x \in I$. These points are known as local optimum. Another definition to the local optimum is a solution with the best value within its neighboring solutions. On the other hand, a global optimum is a solution with the best value among all domain, that is all the feasible solutions.

The algorithms that reach a local optimum are known as local searches. Blum *et al.* [60] describes local search as *algorithms that start from some initial solution and iteratively try to replace the current solution by a better solution in an appropriately defined neighborhood of the current solution.* The popular algorithms k-means and GMM are local searches. They are sensitive to starting points, and they can get stuck easily in a local optimum. These algorithms are very dependent on the seeds, and bad initial solutions can lead to sub-optimal solutions or a poor convergence rate. To improve the results, the algorithms repeat the local search from different initial points and keep the best solution. However, there are smarter ways of improving a solution using meta-heuristics.

Meta-heuristics are methodologies that orchestrate the local searches and higher level strategies to escape from local optimum and reach better solutions [20].

A simple method which improves solution space search is the Iterated Local Search (ILS) [61]. The idea is to apply a perturbation on the current local optimum, trying to move to a different solution. This perturbation needs to be strong enough to lead the next local search to a different local optimum, but not so much to preserve the information gained in the past search. Figure

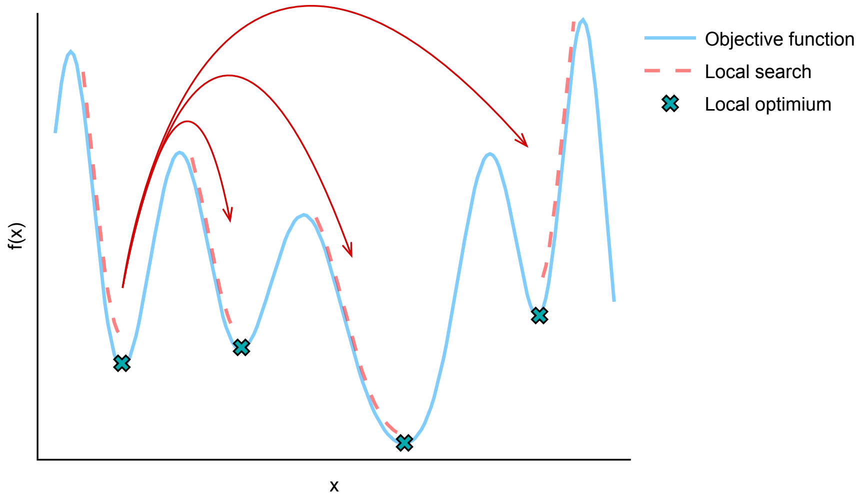


Figure 3.4: Iterated Local Search: the red arrows means the perturbations.

3.4 is an example of a global search using ILS. When the search reaches a local optimum, there is a perturbation that tries to escape from this solution.

The application of the ILS concept to cluster problems, using the k-means algorithm, is known as Random Swap [62]. The design of this algorithm follows the simple idea of ILS: when the local search finds a solution apply the swap, which changes the position of one random exiting cluster to a random point in the dataset. These cluster centers are the new starting points of a new local search. The algorithm rejects the solution if its worse than the best one. Algorithm 1 illustrates and shows how simple is the random swap.

Algorithm 1 Random swap clustering algorithm

```
Initialize the solution $c$
for $T$ times do
    Generate a solution $c'$ by perturbing $c$
    Apply k-means on $c'$
    if the solution $c'$ is better than $c$ then
        Set $c'$ as best solution $c$
    end if
end for
return best solution $c$
```

Another way to search for the global optimum is with genetic algorithms (GA), which is a meta-heuristic that aims to generate high-quality solutions based on the process of natural evolution. The process starts with an initial population of solutions, the algorithm selects two parents and apply a crossover to generate the offspring. After that, the algorithm modifies the offspring using a mutation operator. When the population exceeds, the algorithm applies a survivors operator to keep the population bounded. The process keeps evolving until reaches a maximum number of generations.

A hybrid genetic algorithm for the k-means problem, called HG-means, was proposed by Gribel *et al.* [22]. The proposed algorithm seeks to combine the k-means local improvement procedure with the problem-tailored crossover, mutation and diversification operators. The general structure of the algorithm follows the simple workflow:

- Create an initial population of solutions.
- Select the parents using a binary tournament, which selects two random individuals and keeps the one with the best fitness.
- Apply the crossover operator to create a new individual, by mixing two individuals.
- Apply the mutation operator by randomly perturbing the solution of the created individual.
- Enhance the produced individual with the local search algorithm (k-means).
- Select the survivors for the propagation of the population.

Both proposed algorithms, k-means with Random Swap and HG-means, can efficiently escape from local optimum and find better solutions.

3.4 Summary

The last sections describe some important attributes, summarize and make a clear analysis of the algorithms. Table 3.1 organizes in each row the fundamental attributes. Table 3.2 shows in its columns the attributes of the methods and in its rows, some algorithms of the literature.

Cluster Shape	1	Spherical
	2	Ellipsoidal with empirical covariance matrix.
	3	Ellipsoidal with Shrunk covariance matrix.
	4	Ellipsoidal with Ledoit Wolf covariance matrix.
	5	Ellipsoidal with OAS covariance matrix.
Assignment	6	Hard assignment of the observations classes.
	7	Soft assignment of the observations classes.
Optimization Effort	8	Single local search.
	9	Random Swap meta-heuristics.
	10	Hybrid Genetic Algorithm meta-heuristics.

Table 3.1: Attributes of the analysis.

The cluster shape is at the top of Table 3.1: it can be the standard spherical (Attribute 3.1.1) or the ellipsoidal. In this case, there are four possible methods for estimating the covariance matrix: the empirical (Attribute 3.1.2), the shrunk (Attribute 3.1.3), the Ledoit Wolf (Attribute 3.1.4) and the OAS (Attribute 3.1.5). There are only two assignment options: the hard (Attribute 3.1.6) and the soft (Attribute 3.1.7). The optimization effort is also analyzed: a single local search (Attribute 3.1.8) and two procedures for improving the solution with the application of the local search: the Random Swap meta-heuristics (Attribute 3.1.9) and a hybrid genetic algorithm (Attribute 3.1.10). Table 3.2 has in each row a different method, and in each column, there is a feature that is potentially present.

		1	2	3	4	5	6	7	8	9	10
		Shape					**Asg.**		**Opt.**		
			Ellipsoidal								
	Algorithm	Spherical	Empirical	Shrunk	Ledoit Wolf	OAS	Hard	Soft	Local Search	Random Swap	Hybrid Genetic
a	k-means [18]	×					×		×		
b	k-means++ [63]	×					×		×		
c	k-means ellipsoidal [47, 48]		×				×		×		
d	k-means with RS [62]	×					×			×	
e	HG-means [22]	×					×				×
f	c-means [64]	×						×	×		
g	GMM [54]		×					×	×		
h	RGMM [65]			×				×	×		
i	RGMM with LW [66]				×			×	×		
j	RSEM [67]		×					×		×	
k	GAEM [68]		×					×			×

Table 3.2: Summary of the methods and its attributes.

All the following methods are an application of the EM algorithm, which is an efficient iterative procedure to compute the MLE [69]. The algorithm has two main procedures: the Expectation and the Maximization step. The former fed the observations into the distribution model, and the latter maximizes the likelihood function under the assumption of the expectation step. After some iterations, the EM algorithm is guaranteed to converge to a local optimum monotonically [41].

The most popular statistical model of EM is the GMM (Method 3.2.g), where the models are multivariate Gaussian distributions [54]. So it uses ellipsoidal shape structure to the clusters, giving it more flexibility, and Gaussian distributions representing the probabilities of the soft assignments for each different group. Let the points of the dataset $X = \{x_1, ..., x_n\} \in \mathbb{R}^d$ and the k initial clusters centers $\in \mathbb{R}^d$ be the initialization parameters. The Expectation step updates the ownership weights (soft) of each point $\in X$, while the Maximization step recomputes the weighted clusters centers, the weighted covariances matrices, and the mixing weights variables. The improvements made by the iterative process leads to the convergence of a local optimum.

The standard k-means (Method 3.2.a) proposed by Lloyd [18] is one of the most popular clustering tool used in scientific and industrial applications [70]. K-means is the simplest heuristic that tackles the MSSC problem. It starts with random initials clusters centers, and the algorithm improves the objective function by reassigning each point to a cluster and re-computing each cluster center. The Euclidean distance is used as the similarity measure [37]. That is why the shapes of these clusters are spherical. It is given the same importance to all the points, and there is no distortion of the distance. The k-means has a hard assignment, i.e., assigns each point to exactly one cluster. The algorithm is a two-step procedure: the assignment step makes a hard classification of each point in X to a group with the nearest (Euclidean distance) clusters centers, and the update step recomputes the clusters centers, assuming that the assignment found in the previous step is correct. The assignment step fixes the clusters' centers μ and seeks to optimize the subsets S, while the update step is the opposite: fix the subsets S and optimize the clusters' centers μ. These steps iteratively refine the solution until it finds a local minimum. The complexity k-means' Lloyd's algorithm is $\mathcal{O}(iknd)$, where i is the number of iterations until convergence.

The solution convergence of the k-means is sensitive to their initial clusters centers. The k-means++ (Method 3.2.b), proposed by David Arthur [63], smartly select the initial cluster centers to speed up convergence and the accuracy of the results.

The k-means ellipsoidal (Method 3.2.c) is the classical k-means algorithm, but the Mahalanobis is used instead of Euclidean distance. In 2005, Cerioli [47] described the simple algorithm at a high level without benchmark tests. The author points out that the ellipsoidal clustering field can generate good results and encourages further research in this field. In 2014, Esteban [48] built an adaptive Mahalanobis k-means algorithm with a geosciences application. Although some authors published this method, the ellipsoidal hard assignment version is not public implemented and is not in the main packages of machine learning, such as *sklearn* [71]. The general heuristic is a k-means heuristic generalization with the Mahalanobis distance distortion with Σ covariance matrix associated with each cluster. The algorithm has two steps: assignment and update. The former makes hard classifications of each point $\in X$ to a group with the nearest (Mahalanobis distance) clusters centers, and the latter recomputes the centers and the empirical covariance matrices of each cluster.

The k-means with random swap (Method 3.2.d) is the standard k-means wrapped with the random swap meta-heuristics, and the HG-means (Method 3.2.e) is the hybrid genetic algorithm with k-means as local search.

The Fuzzy c-means algorithm (Method 3.2.f) was initially developed by Dunn in 1973 and tries to solve the MSSC problem. The algorithm is very similar to k-means, but rather than hard assignments, the proposal is to make it soft giving to each point and cluster a weight. The algorithm has two steps: the first computes the ownership weights (soft) of each point $\in X$ and the second updates the weighted clusters centers.

In 2002, Dundar *et al.* [65] improved the results of GMM using a regularization after the empirical estimation of the shape matrix. The application of this regularization in GMM is known as RGMM (Method 3.2.h).

In 2013, Halbe *et al.* [66] introduced the use of adaptive α in the regularization of covariance matrices. In their work, the shape matrices were regularized using the LW methodology (Method 3.2.i). Moreover, their work has shown a significant improvement of results by using these matrices.

In 2012, Zhao *et al.* [67] proposed the application of RS using the EM algorithm, and it was called the random swap EM (RSEM) algorithm. The RSEM performs T iterations and in each iteration the p component is removed (3-29) and added to a new position (3-30), represented with the data point q (3-31). Both indices are chosen using an Uniform distribution.

$$p = \mathcal{U}(1, k) \tag{3-29}$$

$$q = \mathcal{U}(1, n) \tag{3-30}$$

In the swap operation, only the cluster center is changed (3-31). The others parameters, shape matrix (3-32) and weights (3-33), are kept the same as the previous iteration, to maintain the same magnitude of the previous iteration.

$$\vec{\mu}_p = \vec{x}_q \tag{3-31}$$

$$\Sigma_p = \Sigma_p^b \tag{3-32}$$

$$\pi_p = \pi_p^b \tag{3-33}$$

The RSEM framework is described below in Algorithm 2, and as in the original RS algorithm, keeping the best current solution.

Algorithm 2 RSEM Algorithm

```
Initialize solution
for $T$ times do
    Perturbation
    Normalize the weights $\pi$ to sum 1
    Expectation Maximization
    if the solution is better then
        Set the best solution
    end if
end for
return best solution
```

In 2005, Pernkopf *et al.* [68] proposed another meta-heuristics for the GMM: a genetic algorithm. Along with some definitions, one of the highlights is the recombination of crossovers which is made using the single-point crossover [72, 73].

It is clear looking at Table 3.2 that there are some missing algorithms. Many of the possible configurations have never been tested in the literature. We propose, in the following section, some missing methods.

4 Solution Approaches

This chapter describes two ellipsoidal clustering local searches, a soft and a hard one, along with three possibilities of shape matrices regularization. In the last sessions of the chapter, two meta-heuristics are proposed to improve the solution: a random swap perturbation and a hybrid genetic algorithm.

4.1 k-ellipses Clustering

The k-ellipses (KE) heuristic finds a local optimum of the EMSSC problem. The designed method seeks to identify beyond the round shapes; it generalizes to adapt the eccentricity of the ellipse. The basis of the designed method is the k-means. That is, it has the two steps: assignment and update. The former fixes the shapes and centers of the clusters and seeks to optimize the assignments, and the latter fixes the assignments and optimize the clusters' shapes and centers.

The general structure is to initialize the solution and iterate the reassignment and the re-computation of the centers and shapes steps.

Algorithm 3 k-ellipses Clustering

```
Initialize solution
while the stopping criterion has not been met do
    Compute the $d$ Mahalanobis distances
    Assigns the cluster of each sample $w$
    Recompute the clusters centers $\mu$
    Recompute the shape matrices $\Sigma$
end while
return solution
```

Equation (4-1) describes the objective function of EMSSC problem: minimize the sum of squared distances between the points to their respective cluster center.

$$\text{minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{k} w_{i,j} \cdot d_{i,j} \tag{4-1}$$

4.1.1 Initial Solution

The algorithm generates randomly initial clusters by assigning a different data sample to each cluster, according to a discrete uniform distribution. The starting shape of each cluster is spherical, so the shape matrices are $d \times d$ identity matrices.

4.1.2 Distances

The first step of the iteration routine is to compute the distances between the points and the clusters. Those values are stored in the $n \times k$ matrix d. For each cluster, there is a shape matrix associated with it, and these modify the distance. So, the value of $d_{i,j}$ is the Mahalanobis distance between the data sample $\vec{x}_i$ and the cluster center $\vec{\mu}_j$, considering the shape matrix Σ_j.

$$d_{i,j} = \sqrt{(\vec{x}_i - \vec{\mu}_j)^\top \Sigma_j^{-1} (\vec{x}_i - \vec{\mu}_j)} \tag{4-2}$$

However, it is not necessary to invert the Σ_j matrix to obtain distances, for this, it is necessary to obtain the Cholesky decomposition of the shape matrix.

$$\Sigma_j = L_j \cdot L_j^\top \tag{4-3}$$

The matrix L_j is a lower triangular matrix with real and positive diagonal entries. Since Σ_j is a symmetric positive-definite matrix, it has a unique Cholesky decomposition [74]. There are several mathematical operations applied to Equation (4-2) rearranging it to Equation (4-8).

$$d_{i,j}^2 = (\vec{x}_i - \vec{\mu}_j)^\top \Sigma_j^{-1} (\vec{x}_i - \vec{\mu}_j) \tag{4-4}$$
$$d_{i,j}^2 = (\vec{x}_i - \vec{\mu}_j)^\top \left(L_j L_j^\top\right)^{-1} (\vec{x}_i - \vec{\mu}_j) \tag{4-5}$$
$$d_{i,j}^2 = (\vec{x}_i - \vec{\mu}_j)^\top \left(L_j^{\top^{-1}} L_j^{-1}\right) (\vec{x}_i - \vec{\mu}_j) \tag{4-6}$$
$$d_{i,j}^2 = \left(L_j^{-1} (\vec{x}_i - \vec{\mu}_j)\right)^\top \left(L_j^{-1} (\vec{x}_i - \vec{\mu}_j)\right) \tag{4-7}$$
$$d_{i,j} = L_j^{-1} (\vec{x}_i - \vec{\mu}_j) \tag{4-8}$$

Since the matrix L_j is a lower triangular matrix, it is possible to find the distance between the sample $\vec{x}_i$ and the cluster center $\vec{\mu}_j$ computing just a forward substitution in L_j [50], present in Equation (4-9). The computational complexity of a forward substitution is $\mathcal{O}(n^2)$ [75].

$$L_j d_{i,j} = \vec{x}_i - \vec{\mu}_j \tag{4-9}$$

Even with similar computational complexity, the linear solution produces more stable matrices than the inverse.

4.1.3 Assignment Step

The assignment step is the same as the k-means because the clusters' shapes have already distorted the distances. The sample $\vec{x}_i$ belongs to a cluster j if its the smallest distance among all clusters' centers.

The matrix w have in its elements the hard assignments values. A sample $\vec{x}_i$ belongs to a cluster j, the position in the matrix is 1, otherwise is 0.

$$w_{i,j} = \begin{cases} 1, & d_{i,j} \leq d_{i,l} \qquad \forall\, l \in \{1, ..., k\} \\ 0, & \text{otherwise.} \end{cases} \tag{4-10}$$

4.1.4 Update Step

The update step changes the clusters' parameters, which are the clusters' centers and shape matrices. Equation (4-11) represents the center of mass of the points belonging to the cluster.

$$\vec{\mu}_j = \frac{\sum_{i=1}^{n} w_{i,j}\vec{x}_i}{\sum_{i=1}^{n} w_{i,j}} \tag{4-11}$$

The last step of the iteration is to update the shape of the clusters, i.e., the computation of the new shape matrices. The description of the workflow summary of the shape matrix estimation is in Algorithm 4.

To identify the ellipsoidal shapes, as previously described, is necessary to compute the empirical shape matrix by estimation the covariance matrix. Equation (4-12) represent each element of the shape matrix, shown in Equation (4-13). The current matrix multiplication techniques, such as those present in *BLAS* library [76], have a dedicated implementation and makes use of the

Algorithm 4 Recompute the shape matrices

```
for each cluster $j$ do
    Compute the empirical shape matrix
    Regularize the matrix
    Normalize the matrix
end for
return shape matrices
```

cache hierarchy. So, the matrix multiplication implementation can have a good performance in practice.

$$\mathrm{cov}_j^{(p,q)} = \frac{\sum_{i=1}^{n} w_{i,j} \left(x_i^{(p)} - \mu_j^{(p)}\right)\left(x_i^{(q)} - \mu_j^{(q)}\right)}{\sum_{i=1}^{n} w_{i,j}} \tag{4-12}$$

$$\Sigma_j = \begin{bmatrix} \mathrm{cov}_j^{(1,1)} & \cdots & \mathrm{cov}_j^{(1,d)} \\ \vdots & \ddots & \vdots \\ \mathrm{cov}_j^{(d,1)} & \cdots & \mathrm{cov}_j^{(d,d)} \end{bmatrix} \tag{4-13}$$

In the case of the sample amount in a cluster is less than d, the algorithm sets the shape matrix as the identity. If even though the shape matrix is not positive-definite, its forced by changing the eigenvalues to be at least a ϵ to fix this numerical issues. Another problem is when a cluster has no samples, to solve it the algorithm applies a Random Swap iteration to define the center (which we present in Section 4.3.1). The regularization reduces these numerical issues on the estimation of the shape matrix. After the estimation of the empirical shape matrix, the algorithm can apply a regularization, and Equation (3-15) describes it. In Chapter 5 we evaluate all the three regularization methods.

If there are applications of regularization in the heuristic, the method is called Regularized k-ellipses (RKE). The names of the RKE methods using the regularizations are RKE (Shrunk), RKE (OAS) and RKE (LW).

The formulation of the EMSSC, in Chapter 2, has Constraint (2-30). This constraint ensures that the matrices have a non-zero determinant, preventing the clusters associated with these matrices from disappearing, satisfying the constraint. The shape matrices are normalized after the estimation. Equation (4-14) fixes the determinant of Σ in 1. Equation (4-15) is the application of the property of determinant $\det(cA) = c^n \det(A)$ (for an $n \times n$ matrix A) [77] to obtain the parameter α in Equation (4-16).

$$\det(\alpha\Sigma) = 1 \tag{4-14}$$

$$\alpha^d \det(\Sigma) = 1 \tag{4-15}$$

$$\alpha = \frac{1}{(\det \Sigma)^{\frac{1}{d}}} \tag{4-16}$$

Equation (4-17) shows the normalization of the shape matrix, which is the multiplication of α by the original matrix Σ.

$$\Sigma_j^{norm} = \alpha_j \Sigma_j = \frac{1}{(\det \Sigma_j)^{\frac{1}{d}}} \cdot \Sigma_j \tag{4-17}$$

4.2 Gaussian Mixture Models Clustering

Since the GMM is very important for this work, this section describes it. We propose three modifications for the GMM. The first one is the application of the OAS regularization; the other two are meta-heuristics which we explain at the end of this chapter.

The GMM is a widely known algorithm in the literature [78, 79]. It corresponds to a heuristic for the MLE problem. The GMM assumes that a mixture of Gaussian distributions models the data.

$$\sum_{j=1}^{k} \pi_j \, \mathcal{N}(X \mid \vec{\mu}_j, \Sigma_j) \tag{4-18}$$

Equation (4-19) describes the multivariate Gaussian distribution, which is the generalization of the one dimension normal distribution to the $\mathbb{R}^d$. Besides that, this distribution is non-degenerate because the shape matrix Σ is positive definite.

$$\mathcal{N}(\vec{x}_i \mid \vec{\mu}_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma_j}} \exp\left(-\frac{1}{2}(\vec{x}_i - \vec{\mu}_j)^\top \Sigma_j^{-1} (\vec{x}_i - \vec{\mu}_j)\right) \tag{4-19}$$

In addition to the clusters' centers and shape matrices, GMM has two other variables: the ownership and mixing weights. The mixing weights π are the same as the mixture of models. The ownership weights, represented by the variable γ, is a $n \times k$ matrix which represents the soft labels, the value $\gamma_{i,j}$ is the probability of the sample x_i belongs to cluster j. Algorithm 5 describes the general structure of the GMM.

Algorithm 5 GMM Clustering

```
Initialize solution
while the stopping criterion has not been met do
    Compute $\gamma$ ownership weights
    Recompute the mixing weights $\pi$
    Recompute the clusters centers $\mu$
    Recompute the shape matrices $\Sigma$
end while
return solution
```

4.2.1 Initial Solution

In the same way as the hard version, the initial clusters centers are randomly generated by assigning a different data sample to each cluster, according to a discrete uniform distribution. The starting shape is spherical.

After setting the initial points, there is a hard assignment step to define the initial ownership weights (4-20).

$$\gamma_{i,j} = \begin{cases} 1, & d_{i,j} \leq d_{i,l} \qquad \forall\, l \in \{1, ..., k\} \\ 0, & \text{otherwise.} \end{cases} \tag{4-20}$$

The values of $\vec{\mu}$, Σ and π are initially updated using the maximization step, the explanation of this step is in the following subsections.

4.2.2 Expectation Step

Given the current π, μ and Σ, this step computes the estimated ownership weights of each sample belonging to each cluster. Equation (4-21) describes the expectation step.

$$\gamma_{i,j} = \frac{\pi_j \, \mathcal{N}(\vec{x}_i \mid \vec{\mu}_j, \Sigma_j)}{\sum_{c=1}^{k} \pi_c \, \mathcal{N}(\vec{x}_i \mid \vec{\mu}_c, \Sigma_c)} \tag{4-21}$$

4.2.3 Maximization Step

Given the ownership weights, the maximization step estimates the new model by updating the clusters' parameters π, $\vec{\mu}$, Σ, shown in Equation (4-22), Equation (4-23) and Equation (4-24), respectively.

$$\pi_j = \frac{\sum_{i=1}^{n} \gamma_{i,j}}{\sum_{j=1}^{k} \sum_{i=1}^{n} \gamma_{i,j}} \tag{4-22}$$

$$\vec{\mu}_j = \frac{\sum_{i=1}^{n} \gamma_{i,j} \vec{x}_i}{\sum_{i=1}^{n} \gamma_{i,j}} \tag{4-23}$$

$$\Sigma_j = \frac{1}{\sum_{i=1}^{n} \gamma_{i,j}} \sum_{i=1}^{n} \gamma_{i,j} \left(\vec{x}_i - \vec{\mu}_j\right) \left(\vec{x}_i - \vec{\mu}_j\right)^{\top} \tag{4-24}$$

After the estimation of the empirical shape matrix, there is the regularization, by applying Equation (3-15). The three regularizations previously seen are used to improve the matrices estimation.

4.2.4 Convergence criterion

Each iteration increases the likelihood of the data, such that the method is guaranteed to converge to a local optimum [44]. The GMM objective function (4-25) is the likelihood objective function, described in the MLE problem (2-19), with the Gaussian distributions.

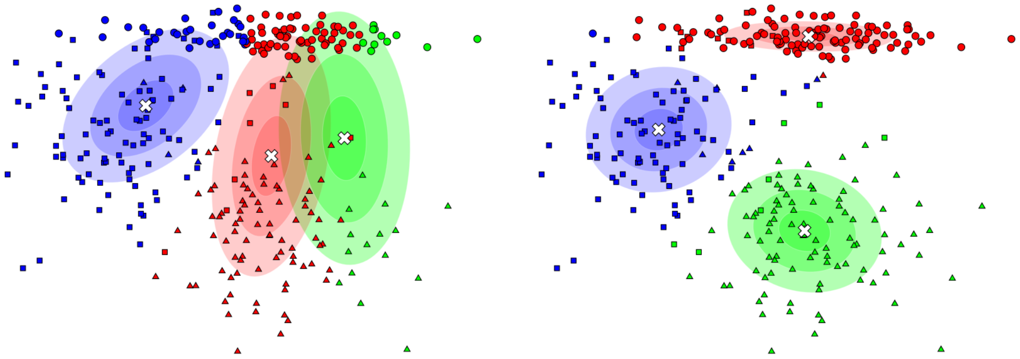
$$\text{maximize} \quad \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \pi_j \; \mathcal{N}(\vec{x}_i \mid \vec{\mu}_j, \Sigma_j) \right) \tag{4-25}$$

If necessary, at the end of the algorithm, assign only one class to a sample. It is possible to allocate each sample to it is the most probable class.

4.3 Meta-heuristics

The algorithms seen in the two previous sessions are local searches. These methods can only improve the objective function, and therefore must terminate in the first local optimum found.

The ellipsoidal methods have many degrees of freedom, due to their large number of parameters. Therefore, to improve the performance of the results, we present four meta-heuristics based on two frameworks: the random swap and the hybrid genetic algorithm. The framework differs according to the type of assignment: a soft and hard random swap algorithm and a soft and hard genetic algorithm. All meta-heuristics throughout this section allow some deteriorations of the objective function at specific steps of the search. Figure 4.1 shows that a single local search can get stuck in an unsatisfactory local optimum. The meta-heuristics have a better solution space search and can reach better results.



4.1(a): Single local search. 4.1(b): Meta-heuristic.

Figure 4.1: Example of the final results of a single local search and a meta-heuristic of ellipsoidal methods.

4.3.1 Random Swap Ellipsoidal Clustering

The first version of the Random Swap is called Random Swap k-ellipses (RS-KE) clustering. The algorithm follows the same idea as the RS framework but applying the k-ellipses clustering local search instead of k-means. Equation (4-26) and Equation (4-27) present the indices of the removed and added cluster swap, respectively, using a uniform distribution.

$$p = \mathcal{U}\left(1, k\right) \tag{4-26}$$

$$q = \mathcal{U}\left(1, n\right) \tag{4-27}$$

The new cluster center p is the sample of index q and the new shape matrix is the identity, as can be seen in Equation (4-28) and Equation (4-29), respectively. The shape matrix of the swapped cluster is set as an identity to reset the shape to spherical.

$$\vec{\mu}_p = \vec{x}_q \tag{4-28}$$

$$\Sigma_p = I_{d\times d} \tag{4-29}$$

Algorithm 6 defines the generic RS clustering. Among all the random swap iterations, the method keeps the best solution, until it finds a better one.

Algorithm 6 Random Swap clustering framework

```
Initialize the solution $c$
for $T$ times do
    Generate a solution $c'$ by perturbing $c$
    Apply clustering local improvement on $c'$
    if the solution $c'$ is better than $c$ then
        Set $c'$ as best solution $c$
    end if
end for
return best solution $c$
```

Analogously to RS-KE, the version of the GMM is called Random Swap Gaussian Mixture Models (RS-GMM) clustering, and there is only one difference between them: since the shape matrices of the GMM can assume arbitrary values, when performing the random swap, a normalization of the matrix is done to keep the magnitude. Otherwise, the matrices could become smaller and smaller, and then some clusters would tend to disappear.

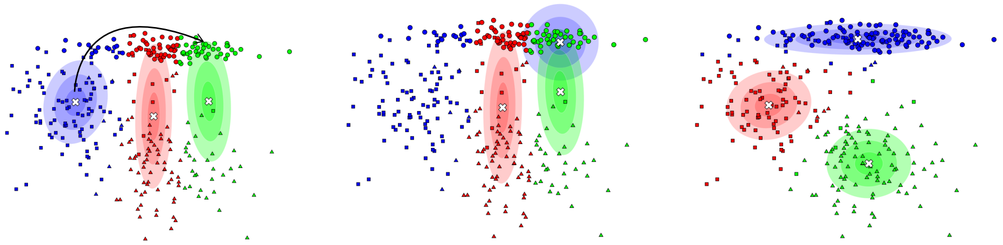
$$\vec{\mu}_p = \vec{x}_q \tag{4-30}$$

The new cluster center is the same as the RS (4-30), along with the indices p (4-26) and q (4-27). As in the RS-HRE, there is a redefinition of the cluster that is chosen to be reallocated, meaning the new shape matrix is the identity matrix. However, do not modify the magnitude of the current clusters, a mean of all the determinants of the shape matrices of the current clusters is computed (4-31). Then the normalization of the matrix is done with this value (4-32).

$$\text{size} = \frac{1}{k}\sum_{j=1}^{k} \det \Sigma_j \tag{4-31}$$

$$\Sigma_p^{norm} = \frac{1}{(\text{size})^{\frac{1}{d}}} \cdot I_{d\times d} \tag{4-32}$$

Figure 4.2 shows an example of an RS move in a hard ellipsoidal methodology, each frame represents an instant of time, from left to right.



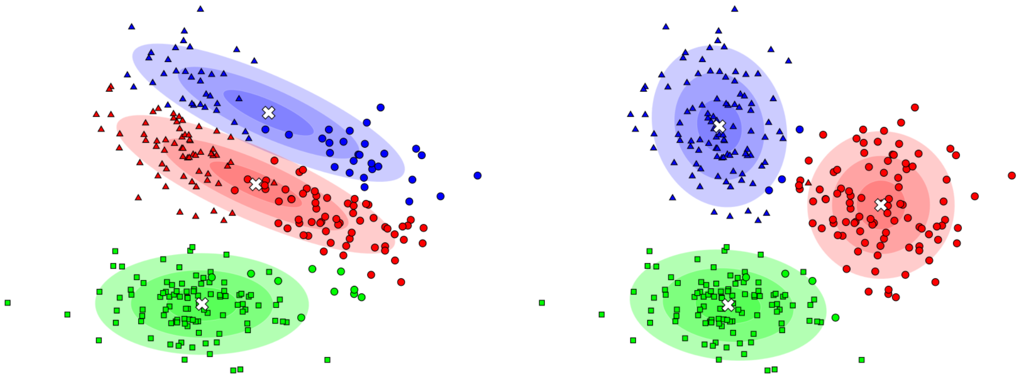
4.2(a): A cluster and data sample are randomly selected.

4.2(b): The algorithm swaps the cluster to the selected data sample and changes the shape matrix to identity.

4.2(c): The algorithm makes a local improvement using the current solution.

Figure 4.2: Example of a Random Swap move in a hard ellipsoidal methodologies.

It is possible to apply the different regularizations by merely changing the local search algorithm. The names of the RKE methods are RS-RKE (Shrunk), RS-RKE (OAS) and RS-RKE (LW). So analogously, the ones using the RGMM are RS-RGMM (Shrunk), RS-RGMM (OAS) and RS-RGMM (LW). Figure 4.3 is an example of the final results of a single local search and RS algorithm, the meta-heuristic escape from the local optimum and performed better.



4.3(a): Single local search.

4.3(b): Random swap (RS) algorithm.

Figure 4.3: Example of the final results of a single local search and a RS algorithm, using the hard ellipsoidal formulation.

4.3.2 Hybrid Genetic Ellipsoidal Clustering

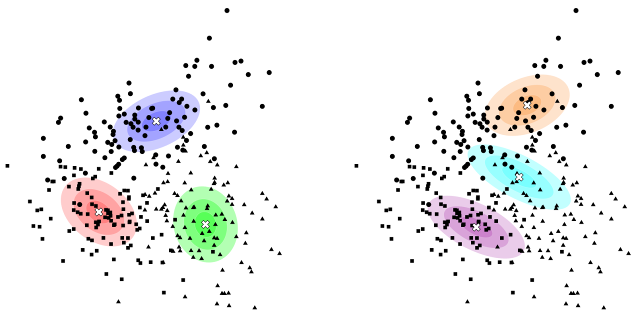
The basis of the hybrid genetic algorithm proposed in this work is the HG-means framework [22]. Gribel *et al.* proposed the HG-means using the k-means as the local search. In this work, we keep the main framework, but we adapted the local search to the ellipsoidal methods: KE and GMM. Algorithm 7 presents the general work-flow of the HG-means.

Algorithm 7 Genetic algorithm framework

```
Initialize population
for $T$ times do
    Select parents $p_1$ and $p_2$
    Generate an offspring $\theta$ from $p_1$ and $p_2$ (crossover)
    Generate an individual $\theta'$ by mutating $\theta$ (mutation)
    Apply clustering local improvement on $\theta'$
    Add $\theta'$ to the population
    if population size = maximum size then
        Select survivors
    end if
end for
return best solution
```

The tuple of the array of clusters' centers $\{\vec{\mu}_1, ..., \vec{\mu}_k\}$ and the array of shape matrices $\{\Sigma_1, ..., \Sigma_k\}$ represents a solution. Also, for the comparison of solutions is considered the value of their respective objective function.

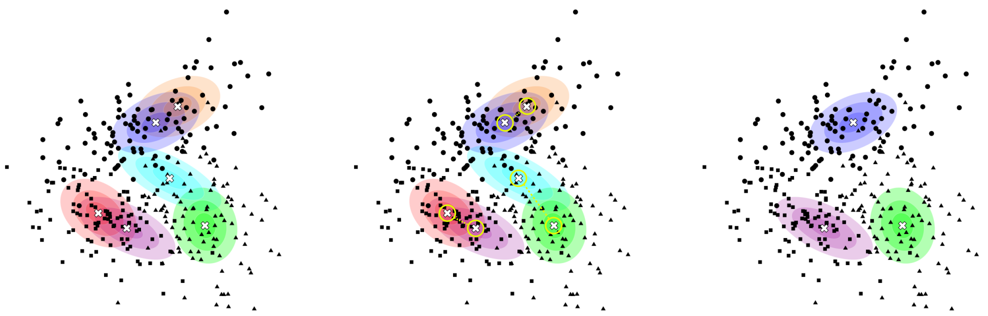
After the generations of the initial population, there is the application of the three operations, until satisfying a criterion. The first one is the selection, which is the selection method of the parents of the new offspring; the second one is the crossover of these parents, generating a new offspring; the last one is the application of a mutation in this offspring. When the population exceeds the maximum size, there is the application of the survivor selection phase keeping the best individuals. The selection operator is the same as the HG-means, with a binary tournament [80]. The binary tournament picks up two random solutions, based on a uniform probability distribution, and selects the one with the best fitness. The newly generated solution comes from these two parents. Figure 4.4 shows an example of 2 selected parents.



4.4(a): First parent. 4.4(b): Second parent.

Figure 4.4: Example of 2 selected parents of a Hybrid Genetic Ellipsoidal Clustering.

The crossover operator needs to match the centers of two parents solutions to generate the offspring. To find it is necessary to solve a minimum weighted bipartite matching. The actual implementation of the algorithm does not consider parameter α presented in Gribel *et al.* [22]. Let $G = (V, E)$ be a complete weighted graph, V be vertex set and E edge set. The vertices can be divided into two disjoint and independent sets $\mathcal{X}$ and $\mathcal{Y}$. A bipartite graph needs that every weighted edge in U connects a vertex in $\mathcal{X}$ to one in $\mathcal{Y}$. The minimum weighted bipartite matching is a bipartite graph in which the edges weights sum is the minimum. The problem is solved using the Hungarian algorithm [81]. After matching the two parents solutions, the algorithm chooses the offspring's clusters: the method selects each cluster among all the matched pairs, with a uniform random selection. The offspring inherits each cluster chosen, and it is the center and its respective shape matrix. Figure 4.5 shows an example of an offspring generation using two selected parents.



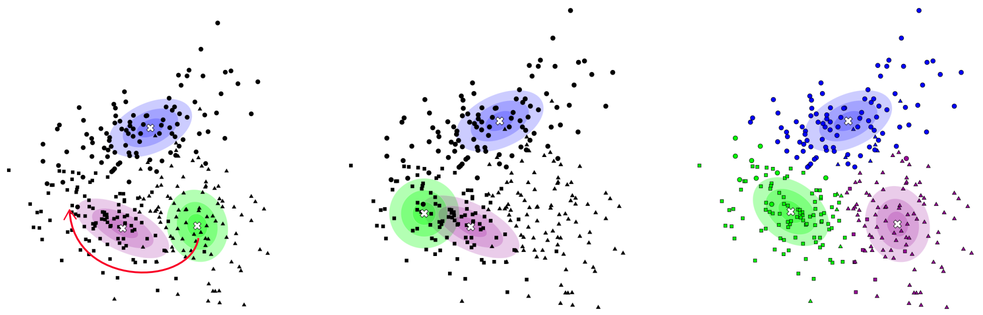
4.5(a): Selected parents.

4.5(b): Matched centers of selected parents.

4.5(c): Offspring with the selected clusters.

Figure 4.5: Example of the offspring generation.

The mutation operator is one random swap step, that is, one cluster is relocated to another position, resetting its shape matrix. This operator is what differentiates the soft genetic algorithm from the hard. After the mutation, the algorithm uses the solution as the starting point for a local improvement until it converges to a local optimum. Figure 4.6 shows an example of the mutation step of an offspring.



4.6(a): Relocation of a cluster center.

4.6(b): Cluster shape reset.

4.6(c): Application of a local improvement at the current solution.

Figure 4.6: Example of the mutation step.

The Hybrid Genetic K-ellipses (HG-KE) Clustering uses the KE as the local improvement, and the RS-KE step for the mutation, leaving the new identity matrix with the determinant equal to 1. On the other hand, the Hybrid Genetic Gaussian Mixture Models (HG-GMM) Clustering uses the GMM as the local improvement. The mutation operator is a step of RS-GMM, which modifies the value of the determinant to maintain the magnitude of the shape matrices of the clusters. In the same way as the RS, the regularized local searches can be applied to the HG. The names of the RKE methods are HG-

RKE (Shrunk), HG-RKE (OAS) and HG-RKE (LW). Also, the ones using the RGMM are HG-RGMM (Shrunk), HG-RGMM (OAS) and HG-RGMM (LW).

4.4 Notation

The following tables summarize the notations of the proposed algorithms, so that is easier to make comparisons in the Chapter 5. The right arrow → on the right of specific rows indicates the variations of methodologies proposed in this work. Table 4.1 is about the hard methods, and Table 4.2 are the soft ones.

		1	2	3	4	5	6	7	8	9	10
		Shape					**Asg.**		**Opt.**		
			Ellipsoidal								
	Algorithm	Spherical	Empirical	Shrunk	Ledoit Wolf	OAS	Hard	Soft	Local Search	Random Swap	Hybrid Genetic
	K-means	×					×		×		
	RS-means	×					×			×	
	HG-means	×					×				×
	K-means++	×					×		×		
	KE		×				×		×		
→	RS-KE		×				×			×	
→	HG-KE		×				×				×
→	RKE (Shrunk)			×			×		×		
→	RS-RKE (Shrunk)			×			×			×	
→	HG-RKE (Shrunk)			×			×				×
→	RKE (LW)				×		×		×		
→	RS-RKE (LW)				×		×			×	
→	HG-RKE (LW)				×		×				×
→	RKE (OAS)					×	×		×		
→	RS-RKE (OAS)					×	×			×	
→	HG-RKE (OAS)					×	×				×

Table 4.1: Aliases of hard methods. The → indicates the variations of methodologies proposed in this work.

		1	2	3	4	5	6	7	8	9	10
		Shape					**Asg.**		**Opt.**		
	Algorithm		Ellipsoidal								
		Spherical	Empirical	Shrunk	Ledoit Wolf	OAS	Hard	Soft	Local Search	Random Swap	Hybrid Genetic
	C-means	×						×	×		
→	RS C-means	×						×		×	
→	HG C-means	×						×			×
	GMM		×					×	×		
→	RS-GMM		×					×		×	
→	HG-GMM		×					×			×
	RGMM (Shrunk)			×				×	×		
→	RS-RGMM (Shrunk)			×				×		×	
→	HG-RGMM (Shrunk)			×				×			×
	RGMM (LW)				×			×	×		
→	RS-RGMM (LW)				×			×		×	
→	HG-RGMM (LW)				×			×			×
→	RGMM (OAS)					×		×	×		
→	RS-RGMM (OAS)					×		×		×	
→	HG-RGMM (OAS)					×		×			×

Table 4.2: Aliases of soft methods. The → indicates the variations of methodologies proposed in this work

5
Computational Experiments

In this chapter, we describe the computational experiments and detail the results. We artificially generated the benchmark instances varying four attributes: number of samples, number of features, number of clusters and separability between the clusters (presented in Section 5.2). These four attributes help to evaluate and identify the limitations of both existing and proposed methods.

This work focuses on comparing the methods regarding their quality of solutions. We left the comparisons regarding CPU time as future works. The implementations are purely in *Julia* *v*0.6.4 [23]. The similarity measure to compare the results is the Adjusted Rand Index (ARI).

The comparisons in this chapter analyze the three main differences between the fundamentals: hard times soft classification, spherical times ellipsoidal shapes, and local times global optimization. We describe all the details and differences between these algorithms features in Chapter 3. We benchmark all the local searches in this work with ten single runs with different starting points, keeping the best results. The meta-heuristics have only a single run.

5.1
Implementation Details

The meta-heuristic parameters are chosen based on superficial tests made during the development of the algorithms. We left a better calibration study as future works. The number of RS iterations chosen is 100. Also, the parameters of HG are the following:

- Minimum population: 4.
- Maximum population: 10.
- Maximum number of iterations: 100.

5.2 Datasets

A controlled testing environment is essential to this work. As we evaluate several algorithms, the inconsistency of a real-world dataset could undermine the analysis of the results. To reach a controlled testing environment, we generated artificial datasets.

We generate artificial datasets using the *clusterGeneration* package [82] in *R* programming language [83]. The package implements the proposed methodology in Qiu *et al.* [84]. The measure of separation between two clusters is the index J defined in another Qiu *et al.* work [85].

To compute the separation index, the method projects the points of a cluster in a direction $\vec{a}$. Considering the lower (L_j) and upper (U_j) 0.025 percentile of the projected cluster j in the direction $\vec{a}$, Equation (5-1) describes the quantile version of the separation index J between the clusters p and q.

$$J_{pq}(\vec{a}) = \frac{L_q - U_p}{U_q - L_p} \tag{5-1}$$

Figure 5.1 shows an example of a lower and upper percentile of two clusters in a direction $\vec{a}$.

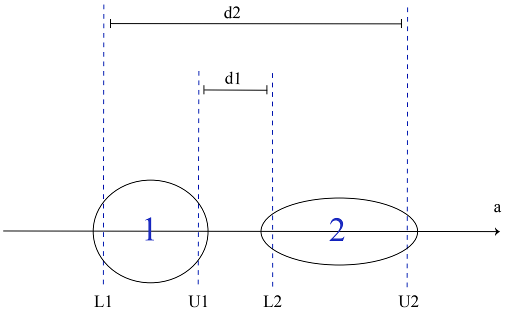


Figure 5.1: Illustration of two clusters (1 and 2) with its lower (L) and upper (U) percentiles in the direction $\vec{a}$. The two measures $d_1 = L_2 - U_1$ and $d_2 = U_2 - L_1$ are necessary to compute the separation index, which is $J(\vec{a}) = \frac{d_1}{d_2} = \frac{L_2 - U_1}{U_2 - L_1}$.

Equation (5-2) describes the optimal separation index J^*_{ab} between two clusters a and b, which is where the quantile version of the separation index is maximized.

$$J^*_{pq} = \underset{\vec{a}}{\text{maximize}} \quad J_{pq}(\vec{a}) \tag{5-2}$$

The algorithm produces for each cluster a Gaussian distribution with two parameters: the average (cluster center) and covariance matrix (cluster shape). Then, the algorithm spreads the clusters in a simplex mesh, and choose the distributions parameters considering the separation index J. Figure 5.2 illustrates a simplex mesh of clusters in a two-dimensional space. The neighboring clusters are those that have a single edge linking them.

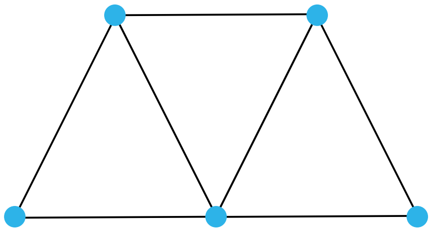


Figure 5.2: Visualization of a two-dimensional simplex mesh. The package allocates a cluster center in each vertex and scales edges' size according to the separability index. The neighboring clusters

After the generation of the elliptical distributions, the algorithm produces the samples of each one of the k clusters. In this work, there are 4 parameters to generate the instances:

- Number of clusters (k).
- Cluster size (m), that is, the number of samples per cluster (total number of samples $n = k \cdot m$).
- Separability (c), which means the degree of separation J between a cluster and its neighboring clusters. This index belongs to the range $[-1, +1]$. The higher the index, more separated are the clusters.
- Number of non-noisy variables (d), which represents the features of the observations in a dimensional space.

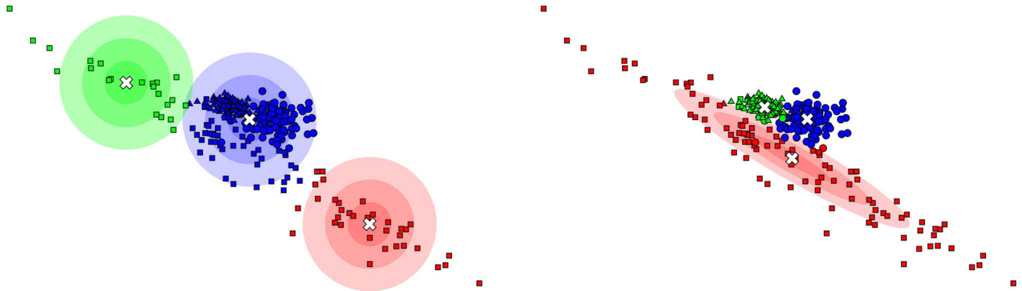
The other parameter used in the package does not vary according to the instance: the minimum and maximum eigenvalues of the shape matrices. The eigenvalues are randomly chosen with a uniform probability over the range $[1, 200]$. Ultimately, these eigenvalues generate the shape of the cluster, because they are the eigenvalues of the positive definite matrix that represents the ellipsoidal shape.

Table 5.1 summarizes artificially created datasets. For each configuration, we generate the indicated number of instances. For example, there are 100 instances with the settings (n: 1000, k: 10, c: -0.26, d: 70) and 15 instances with the settings (n: 2000, k: 20, c: 0.21, d: 40). There are two significant groups of the generated data: the instances with 100 points per cluster and those with 500 points. Those with smaller cluster size are to analyze the impact of dimensionality, clusters number and separability; the remaining ones are to analyze the impact of cluster size.

Instances	Number of samples (n)	Samples per cluster (m)	Number of Clusters (k)	Separability (c)	Number of features (d)
100	300	100	3	$\{-0.26, -0.10, 0.01, 0.21\}$	$\{2, 5, 10, 20, 30, 40, 50, 60, 70\}$
100	1000	100	10	$\{-0.26, -0.10, 0.01, 0.21\}$	$\{2, 5, 10, 20, 30, 40\}$
15	2000	100	20	$\{-0.26, -0.10, 0.01, 0.21\}$	$\{2, 5, 10, 20, 30, 40\}$
100	1500	500	3	$\{0.01\}$	$\{50, 75, 100, 125, 150\}$

Table 5.1: Parameters of the artificially generated instances.

This work analyses how the algorithms behave in simple datasets with ellipsoidal shapes. We left the analysis of the performance in a very high dimension or with a large number of clusters as future work. Figure 5.3 shows an instance that has only two features and three clusters, but spherical algorithms perform poorly.



5.3(a): K-means

5.3(b): RS-GMM (OAS)

Figure 5.3: A comparison between the actual implementation of the algorithms showing that spherical methodolgies may have difficulty classifying instances.

5.3 Similarity Measure

During the tests, the results are compared to their correct classification through the Adjusted Rand Index (ARI) [86]. The ARI is the corrected version of the Rand Index [87, 88], widely known and most used similarity measure for clustering analysis [89]. This measure establishes a baseline by using the expected similarity of all pair-wise comparisons between the classifications.

Let U and V be two clustering classifications, and $t_{ij} = |U_i \cap V_j|$ be the classifications in common considering the clusters i for U and j for V. Table 5.2 defines the contingency table and Equation (5-3) defines the ARI formula.

		V			Total
		v_1	$\dots$	v_2	
U	u_1	t_{11}	$\dots$	t_{1k}	a_1
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	u_k	t_{k1}	$\dots$	t_{kk}	a_r
Total		b_1	$\dots$	b_s	

Table 5.2: Contingency table of ARI.

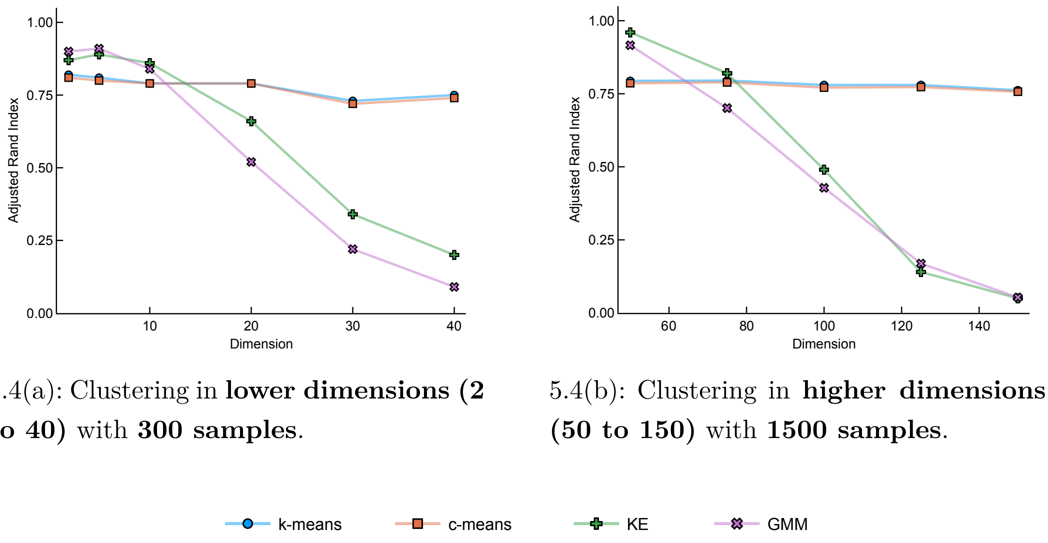
$$\text{ARI} = \frac{\sum_{ij}\binom{t_{ij}}{2} - \sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}/\binom{t}{2}}{\frac{1}{2}\sum_i \binom{a_i}{2} + \frac{1}{2}\sum_j \binom{b_j}{2} - \sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}/\binom{t}{2}} \tag{5-3}$$

The ARI values belong to the range $[-1, 1]$. When it is close to 0, means that is as good as a random classification. When is $+1$, the classifications are identical up to a permutation, i.e., correct.

5.4 Experimental Results

5.4.1 The Impact of the Size of Clusters

In order to analyze the impact of the size of clusters, i.e., the number of points per cluster, we compare the clustering performance of the "pure" algorithms (no regularization and no meta-heuristics): k-means, c-means, k-ellipses, and GMM. We compare the algorithms with two different cluster sizes: 100 and 500 points.



5.4(a): Clustering in **lower dimensions (2 to 40)** with **300 samples**.

5.4(b): Clustering in **higher dimensions (50 to 150)** with **1500 samples**.

Figure 5.4: Comparison between algorithms with n : 300 and n : 1500 instances. Note that the domain of the charts are different.

In Figure 5.4(a) we can see that k-means' and c-means' clustering quality (measured by ARI) are not much affected by the dimension, i.e., the number of features in each sample, although there is a slight decrease. However, the performance of the ellipsoidal methods decreases steeply when the number of dimensions increases. Since the c-means performs strictly worse than k-means (you can see it in the Appendix A tables), from here, we will only analyze the hard method.

The amount of parameters in the ellipsoidal methodology causes this problem. While the number of parameters to be estimated in spherical clustering is $\mathcal{O}(kd)$, in the ellipsoidal clustering is $\mathcal{O}(kd^2)$. Therefore we need much more points to estimate parameters in ellipsoidal clustering accurately. This problem is confirmed by Figure 5.4(b), in which we increase the number of points per cluster and performance in dimension 50 is much better again. Moreover, as the dimension grows, the quality of the clustering degrades again.

We discuss the shape matrix estimation in Section 3, it is a relevant problem in statistics, and there are proposed solutions in the literature. The empirical conclusion here is that shape matrices $d \times d$ need $\mathcal{O}(d^2)$ points to be adequately estimated.

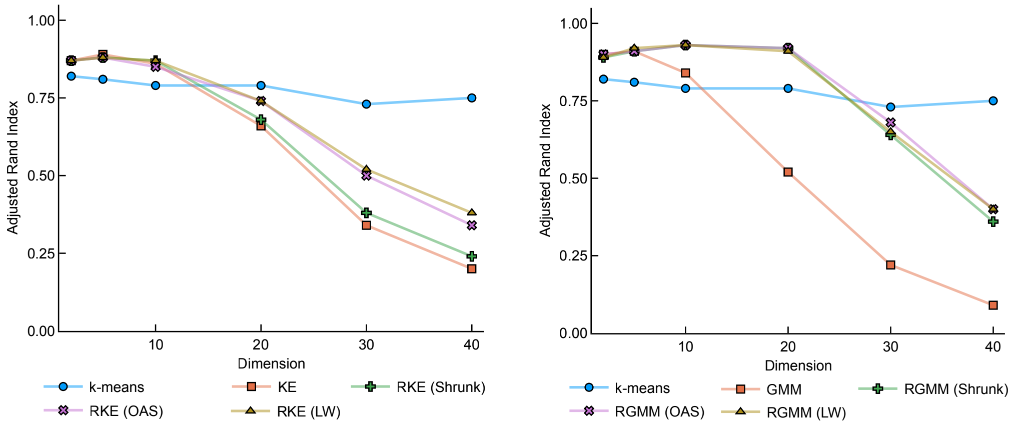
5.4.2 The Impact of Dimensionality

As previously seen, shape matrices need a sufficient number of points to be estimated. If there are enough points and the estimation is good, the results do not tend to degrade. However, it is not necessarily what happens in the real world. There are two methodologies capable of improving the estimation and the solution space search: regularization and meta-heuristics.

5.4.2.1 Regularization

When the number of features is significantly large compared to the sample size, the shape matrices can become ill-conditioned and poorly estimated. The regularization techniques were developed to overcome this difficulty. The benchmark tested the three presented methodologies in Chapter 3.2.2.1: Shrunk, OAS, and LW.

Figure 5.5 shows that, as expected, methods with regularization improve the estimation of the shape matrices, and therefore, the final results. When there are enough samples per cluster (usually d^2), the methods have similar results. Figure 5.5 also shows that both hard and soft methods improved the results. However, the regularization impacted more the soft methodologies: the degradation of the results in the soft methods starts at a larger dimension than the hard ones.



5.5(a): Hard algorithms.

5.5(b): Soft algorithms.

Figure 5.5: Comparison between regularization methodologies (n : 300, k : 3, c : 0.01).

5.4.2.2 Meta-heuristics

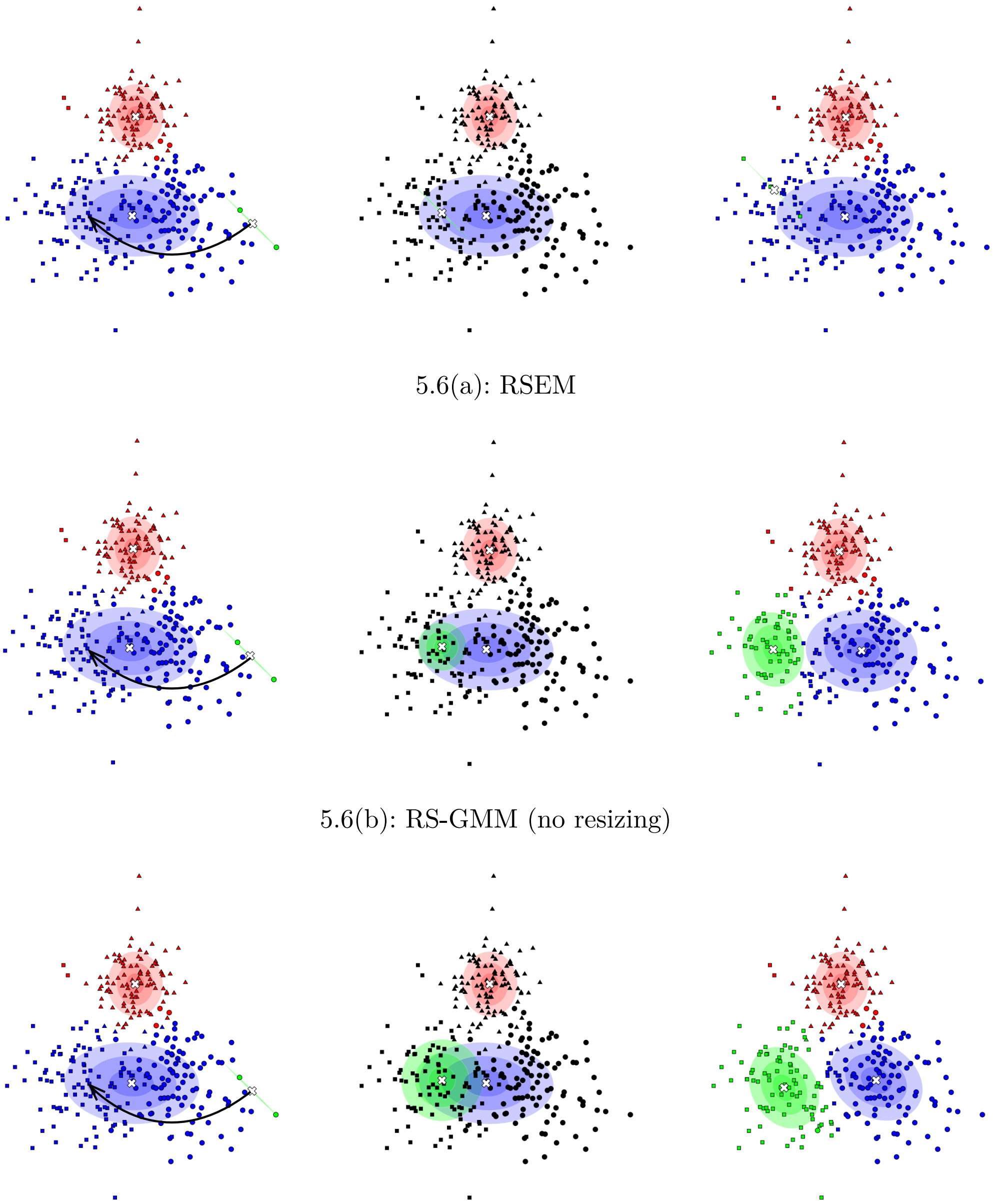
When using meta-heuristics, there is an expectation of a better search in the solution space and therefore, find better solutions. Along both meta-heuristics seen, there is a step in common: an RS iteration, which changes the position of a cluster. In hard methodologies, the swapped cluster matrix is set merely to identity, because there is a restriction that fixes the determinant of shape matrices in 1: this way the clusters do not disappear. On the other hand, the soft methods do not have this restriction. Their shape matrices can assume arbitrary values. There are three ways of handling the new shape matrix when doing a random swap iteration in a soft meta-heuristic. The selected swap cluster:

– RSEM: Keep the same shape matrix as the previous iteration.

– RS-GMM (no resizing): Reset the shape matrix to identity.

– RS-GMM: Reset the shape matrix to identity, and resize it to keep the same magnitude as the other shape matrices (explained in Equation (4-31) and Equation (4-32)).

Figure 5.6 shows an example of the three ways of swap possibilities: the left panel illustrates the moment before the random swap; the center panel is the moment before the swap with the points unassigned; the right panel is the results after the GMM improvements. Figure 5.7 compare the results of these.



5.6(a): RSEM

5.6(b): RS-GMM (no resizing)

5.6(c): RS-GMM

Figure 5.6: An example of the swap possibilities in soft methods. Three moments are shown: before the random swap (left panel); right after the swap with unassigned the points (center panel); after the GMM local improvement (right panel).

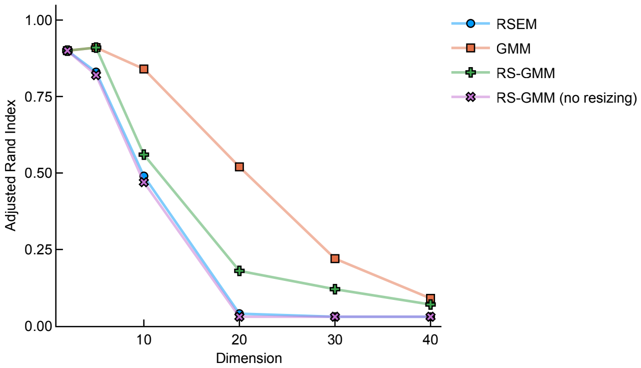


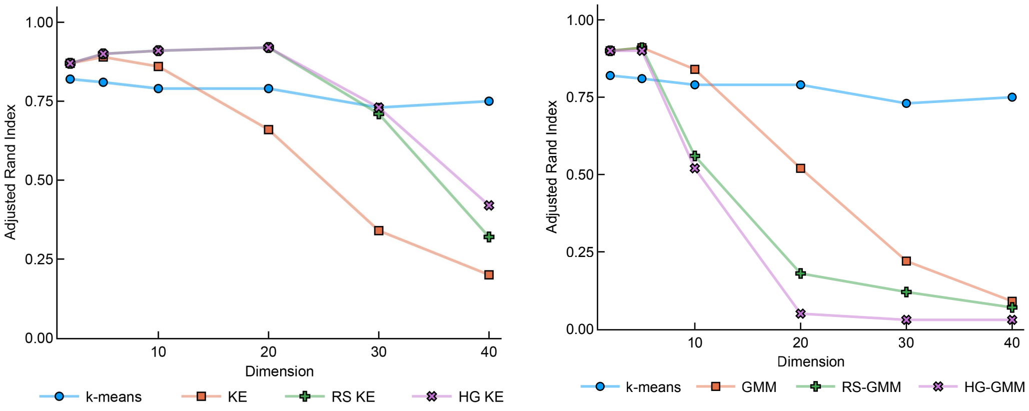
Figure 5.7: Comparison between random swap in soft methods (n : 300, k : 3, c : 0.01).

Until $d = 5$, the RS-GMM method maintains the GMM result, while the other two methods do not. However, the results are still not satisfactory. It may seem counter-intuitive since meta-heuristics can only accept solutions equal to or better than local searches. It is important to analyze an RS iteration to understand what is happening. The chosen instance is where the results start to get worse: (n : 300, k : 3, d : 10, c : 0.01). Table 5.3 shows the RS iterations in this instance. Each line represents an iteration that was accepted.

Iteration	ARI	Obj. Func.	Cluster 1		Cluster 2		Cluster 3	
			Det.	Eig. diff.	Det.	Eig. diff.	Det.	Eig. diff.
1	0.83	−22.23423	10^6	23.3	10^5	11.9	10^6	11.5
2	0.85	−22.23375	10^6	23.2	10^5	11.8	10^6	11.5
4	0.85	−22.23333	10^6	23.2	10^5	11.8	10^6	11.6
6	0.86	−22.23242	10^6	23.2	10^5	11.7	10^6	11.5
9	0.88	−22.23163	10^6	23.3	10^5	11.7	10^6	11.5
16	0.87	−22.23159	10^6	23.2	10^5	11.7	10^6	11.5
25	0.88	−22.23158	10^6	23.3	10^5	11.7	10^6	11.6
37	0.87	−22.23156	10^6	23.3	10^5	11.7	10^6	11.6
45	0.90	−22.23064	10^6	22.8	10^5	11.5	10^6	11.4
49	0.49	−22.05719	10^6	25.3	10^{-21}	24746893.6	10^7	12.7
53	0.47	−22.04634	10^5	26.0	10^{-21}	24746893.7	10^7	11.2
70	0.48	−21.99079	10^6	25.0	10^{-21}	26978527.1	10^7	11.9

Table 5.3: Accepted RS iterations in an instance of (n : 300, k : 3, d : 10, c : 0.01).

Along the iterations, the objective function and the ARI get better, until iteration 49. At this swap, the difference between the eigenvalues gets huge, and the size of the cluster becomes very small. That means that the cluster became very small and disappeared. The value of the function keep improving, but the ARI only gets worse. After this, the current solution can never recover the ARI, and the result of the RS gets worse than a single local search. When the cluster size is small, the matrices estimations gets ill-conditioned, and the results start to degrade. Without regularization, high-quality solutions of the proposed model may be very distant from the desired classifications. Figure 5.8 shows comparisons of both hard and soft meta-heuristics.



5.8(a): Hard algorithms.

5.8(b): Soft algorithms.

Figure 5.8: Comparison between meta-heuristics methodologies (n : 300, k : 3, c : 0.01). RS and HG exploit well the solution space in the hard methods, but the genetic algorithm seems to take advantage on a higher dimensions.

As seen in Figure 5.8(a), both the RS and the HG present proper exploitation of the solution space of the hard method, with the latter seeming to take advantage on a higher dimension instance. Even though the meta-heuristics improved the results, regularization can also be used to stabilize the shape matrix estimation. The next subsection tests the improvement caused by the regularization with meta-heuristics.

5.4.2.3 Regularization and Meta-heuristics

As previously seen, the regularizations could solve the problem of the ill-conditioned matrices in RS-GMM. Figure 5.9 adds RS-RGMM (Shrunk), and reveals that meta-heuristics and the regularization together improve a lot the results.

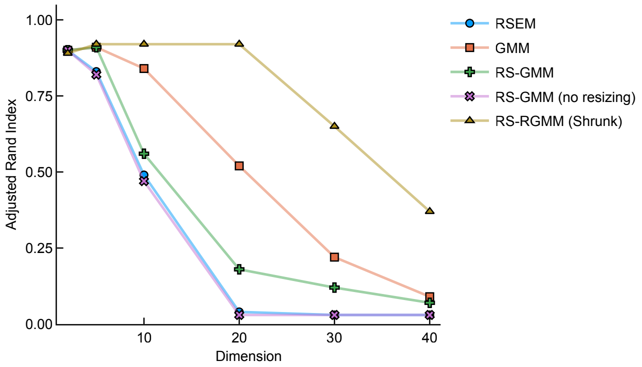
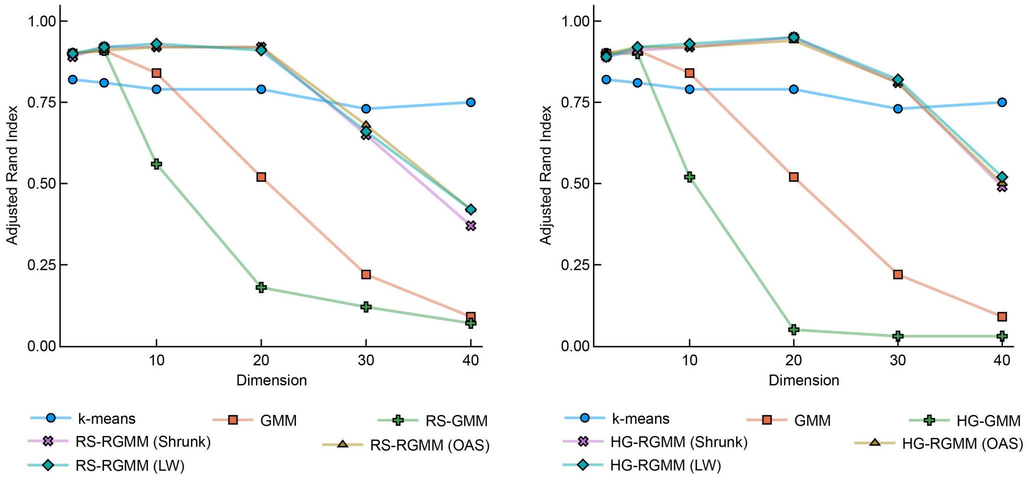


Figure 5.9: Comparison between GMM Random Swap and RS-RGMM (Shrunk) methods (n : 300, k : 3, c : 0.01).

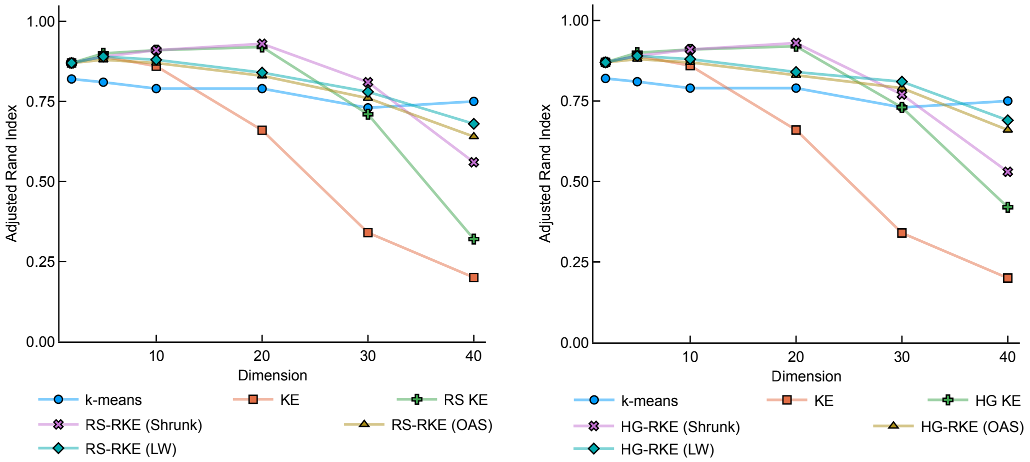
Figure 5.10 presents two charts with soft methods: Figure 5.10(a) is the comparison os the RS with different regularizations, while Figure 5.10(b) is the same but with HG. Figure 5.11 is similar but considering the hard methods.



5.10(a): RS algorithms.

5.10(b): HG algorithms.

Figure 5.10: Comparison between meta-heuristics with regularization soft methods (n : 300, k : 3, c : 0.01).



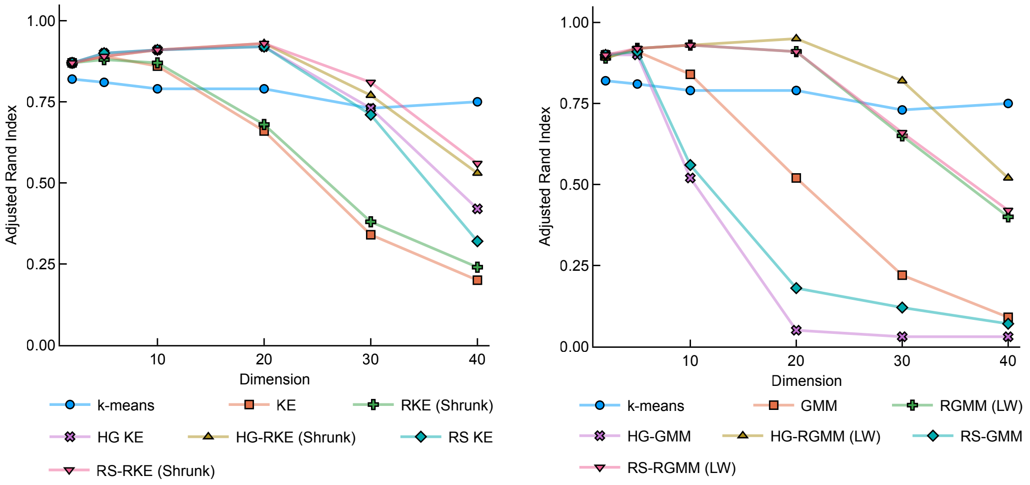
5.11(a): RS algorithms.

5.11(b): HG algorithms.

Figure 5.11: Comparison between meta-heuristics with regularization hard methods ($n : 300$, $k : 3$, $c : 0.01$).

For the $k = 3$, the results of RS-RKE, HG-RKE, RS-RGMM, and HG-RGMM are pretty close. We choose to compare the regularizations with slightly better results: Shrunk and LW, for hard and soft, respectively.

Figure 5.12(a) presents the comparison of the hard methods between the regularization (Shrunk), the meta-heuristics and both together. Figure 5.12(b) is the same comparison, but with soft methods and LW as the regularization.



5.12(a): Hard algorithms.

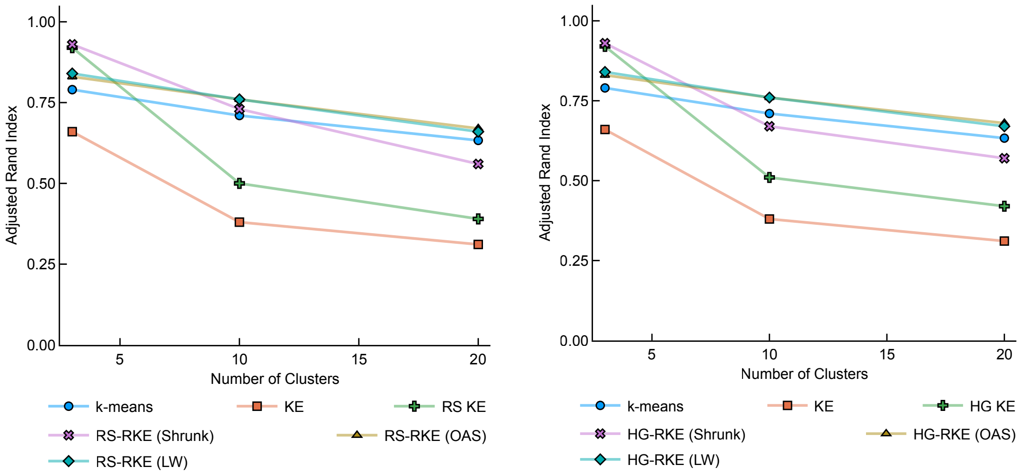
5.12(b): Soft algorithms.

Figure 5.12: Comparison between regularized meta-heuristics methodologies ($n : 300$, $k : 3$, $c : 0.01$).

The KE and the GMM have significant improvement with the implementation of meta-heuristics and regularization. The shape matrix estimations are very good, even with few points. The degradation only starts at dimension 20. Only after the dimension 30, the results of k-means get better than the ellipsoidal.

5.4.3 The Impact of the Number of Clusters

When the number of clusters grows, the first thing to observe is which regularization has remained more stable. Figure 5.13 compares the regularizations of the hard methods, with the number of features fixed in 20.

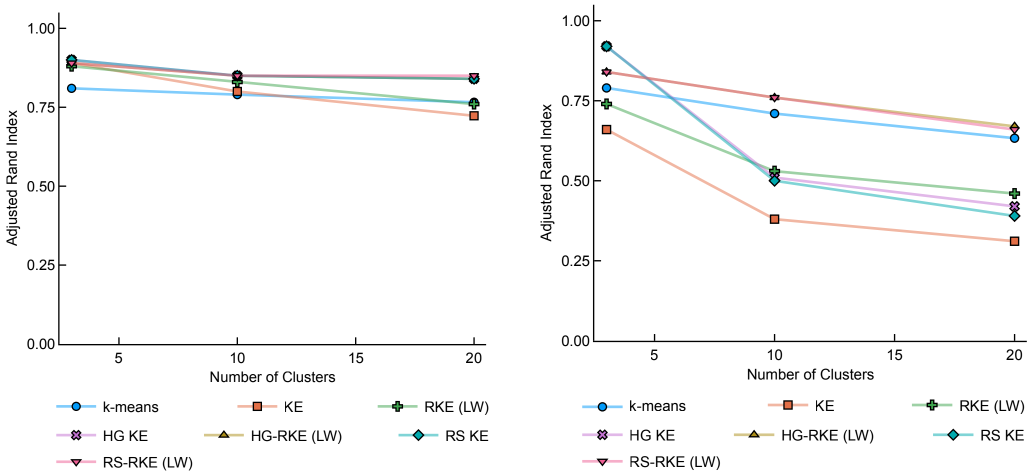


5.13(a): Random swap algorithms.

5.13(b): Hybrid genetic algorithms.

Figure 5.13: Comparison between regularization methodologies of the hard methods (m : 100, c : 0.01, d : 20).

It seems that LW had better results than the others. Figure 5.14 compares different meta-heuristics for number of features at 5 and 20.



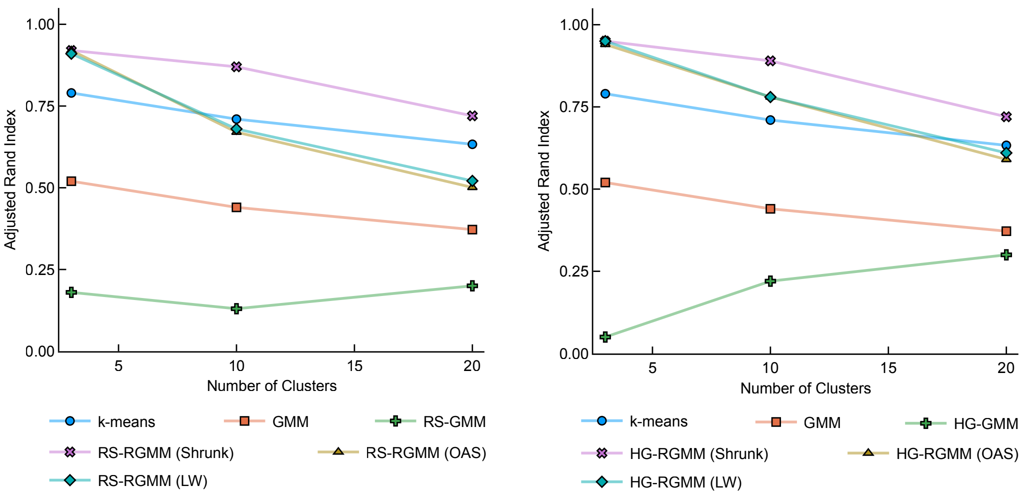
5.14(a): Number of features (d): 5

5.14(b): Number of features (d): 20

Figure 5.14: Comparison between the regularized meta-heuristics methodologies of hard methods (m : 100, c : 0.01).

For a high number of clusters and when it has enough points, the results reveal that meta-heuristics make more difference than regularization. However, when the dimension grows along with the number of clusters, regularization makes more difference. The two together are still the best choice, securing the best results.

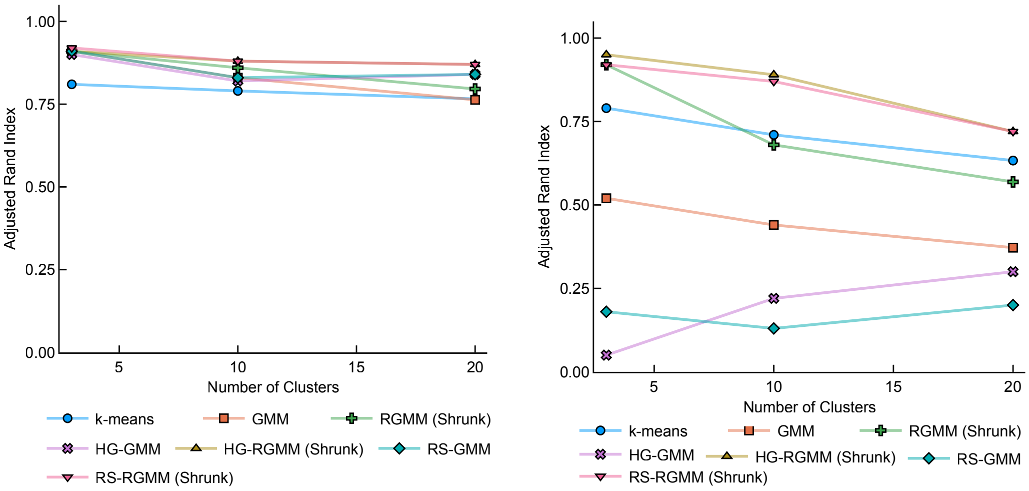
Figure 5.14 is the comparison of regularization methodologies with the soft methods. The Shrunk methodology performed very well when the number of clusters is large.



5.15(a): Random swap algorithms.

5.15(b): Hybrid genetic algorithms.

Figure 5.15: Comparison between regularization methodologies of soft methods (m : 100, c : 0.01, d : 20).



5.16(a): Number of features (d): 5

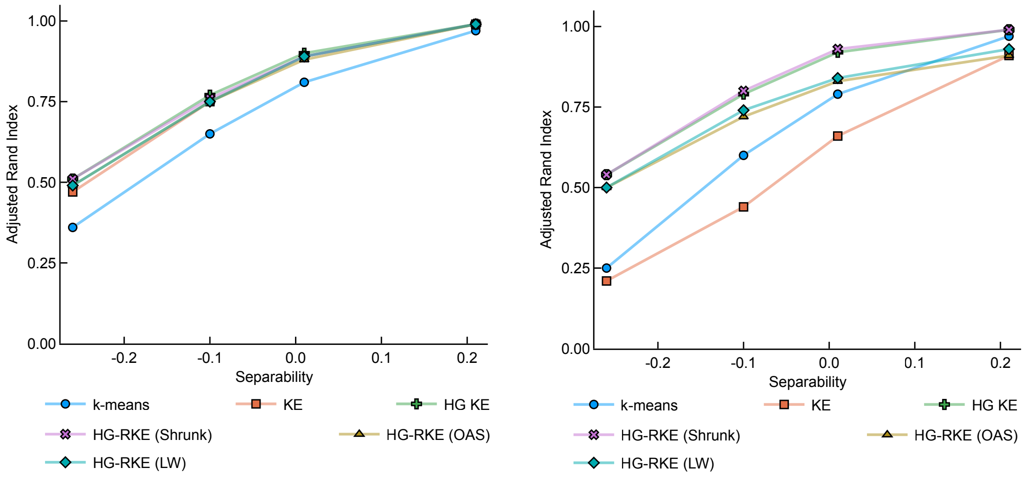
5.16(b): Number of features (d): 20

Figure 5.16: Comparison between regularized meta-heuristics methodologies of soft methods (m : 100, c : 0.01).

The analyzes made for the hard methods are analogous to those of the soft. The methods with meta-heuristics and regularizations continue to perform very well. The RS-GMM and the HG-GMM had good performance when the cluster number is high, and the size is low.

5.4.4 The Impact of Separability

After analyzing the impact of cluster size, dimensionality and number of clusters, the last variable to be analyzed is the separability. Among the meta-heuristics, we choose the HG for comparisons, as it shows to be more robust than the RS in more scenarios. Figure 5.17 compares the regularizations of the hard methods, with the number of clusters fixed at 3.



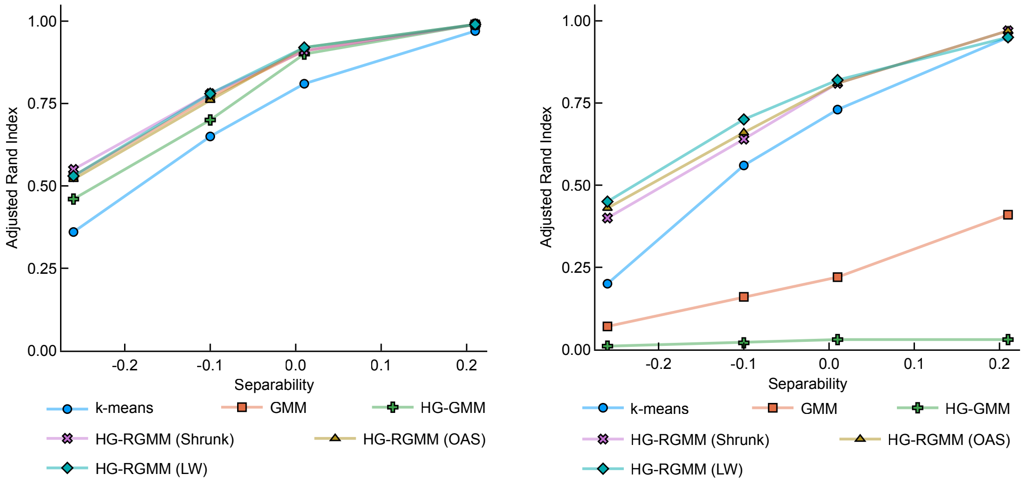
5.17(a): Number of features (d): 5

5.17(b): Number of features (d): 20

Figure 5.17: Comparison between regularization methodologies of hard methods (n : 300, k : 3).

Figure 5.17(a) shows that when there are enough points, the ellipsoidal methods perform better than k-means. When the separability is lower, the ellipsoidal methods perform much better than the spherical. However, when clusters separate, the performance of k-means improves a lot, getting close to the ellipsoidal.

Figure 5.17(b) shows that the OAS and LW regularizations degrade the results when the number of features increases. This degradation is because OAS and LW change the shape matrix much more than Shrunk. HG usually performs very well when searching the solution space. Therefore, regularizations that estimate the data shape too spherical may work against the methodology, leading to poorer classifications. So, using a regularization that changes the shape matrix minimally to keep it well conditioned seems to be the best option, that is why Shrunk stands out.



5.18(a): Number of features (d): 5

5.18(b): Number of features (d): 30

Figure 5.18: Comparison between regularization methodologies of soft methods (n : 300, k : 3).

In the lower dimensions, soft and hard methods tend to behave similarly. However, in higher dimensions, they seem to behave better. Figure 5.18(b) shows that even in dimension 30 the results of the ellipsoidal methods are still better than the k-means.

5.5 Summary

After the analysis, among the proposed methods we select some that produce the best results for the final comparisons: HG-RKE (Shrunk) and HG-RGMM (Shrunk). Beyond obtaining better results when the number of clusters is high, these methodologies were very robust when there are few points for the shape matrices estimation.

Table 5.4 compares the methods with the following instances settings: $m \in \{100\}$, $c \in \{0.26, -0.10, 0.01, 0.21\}$, $d \in \{2, 5, 20\}$, $k \in \{3, 10, 20\}$. Table 5.5 compares the methods with another instances settings: $m \in \{500\}$, $c \in \{0.01\}$, $d \in \{50, 75, 100, 125\}$, $k \in \{3\}$. The results show that elliptical methods are more advantageous than spherical ones, obtaining better performance in most instances. The soft methodologies, on the artificially generated datasets, outperformed the hard ones.

We also apply our algorithms to 18 datasets from the UC Irvine repository [2]. Table 5.6 compares these datasets with the selected methods. The results are almost the same as the artificial regarding shapes: the ellipsoidal algorithms outperformed the spherical. The GMM has some good results; however, the objective function of these datasets is not as reliable as the artificial ones. The full results tables are in Appendix A.

c	d	k: 3 and n: 300				k: 10 and n: 1000				k: 20 and n: 2000			
		HG-means	HG-RKE (Shrunk)	GMM	HG-RGMM (Shrunk)	HG-means	HG-RKE (Shrunk)	GMM	HG-RGMM (Shrunk)	HG-means	HG-RKE (Shrunk)	GMM	HG-RGMM (Shrunk)
-0.26	2	**0.46**	0.45	0.43	0.41	0.46	0.46	0.44	**0.48**	0.46	**0.47**	0.43	**0.47**
-0.26	5	0.36	0.51	0.53	**0.55**	0.31	0.35	0.33	**0.38**	0.29	0.32	0.29	**0.35**
-0.26	20	0.26	0.54	0.28	**0.68**	0.19	0.16	0.11	**0.37**	0.14	0.09	0.08	**0.20**
-0.10	2	0.69	0.74	**0.75**	0.75	0.71	0.74	0.67	**0.75**	0.71	**0.74**	0.62	**0.74**
-0.10	5	0.64	0.76	0.77	**0.78**	0.60	0.68	0.62	**0.70**	0.59	0.66	0.56	**0.68**
-0.10	20	0.60	0.80	0.41	**0.86**	0.48	0.43	0.26	**0.68**	0.41	0.31	0.22	**0.52**
0.01	2	0.82	0.87	**0.90**	0.89	0.85	0.88	0.82	**0.90**	0.85	0.88	0.76	**0.90**
0.01	5	0.81	0.89	**0.91**	**0.91**	0.79	0.85	0.83	**0.88**	0.78	0.84	0.76	**0.87**
0.01	20	0.79	0.93	0.52	**0.95**	0.71	0.67	0.44	**0.89**	0.64	0.57	0.37	**0.78**
0.21	2	0.97	**0.99**	**0.99**	**0.99**	0.98	**0.99**	0.93	**0.99**	0.98	**0.99**	0.85	**0.99**
0.21	5	0.97	**0.99**	**0.99**	**0.99**	0.97	**0.99**	0.97	**0.99**	0.96	**0.99**	0.91	**0.99**
0.21	20	0.97	0.99	0.70	**1.00**	0.95	0.95	0.72	**1.00**	0.94	0.89	0.68	**0.99**

Table 5.4: Comparison between the selected methods (HG-means, HG-RKE (Shrunk), GMM and HG-RGMM (Shrunk)) with some meaningful instances (m : 100).

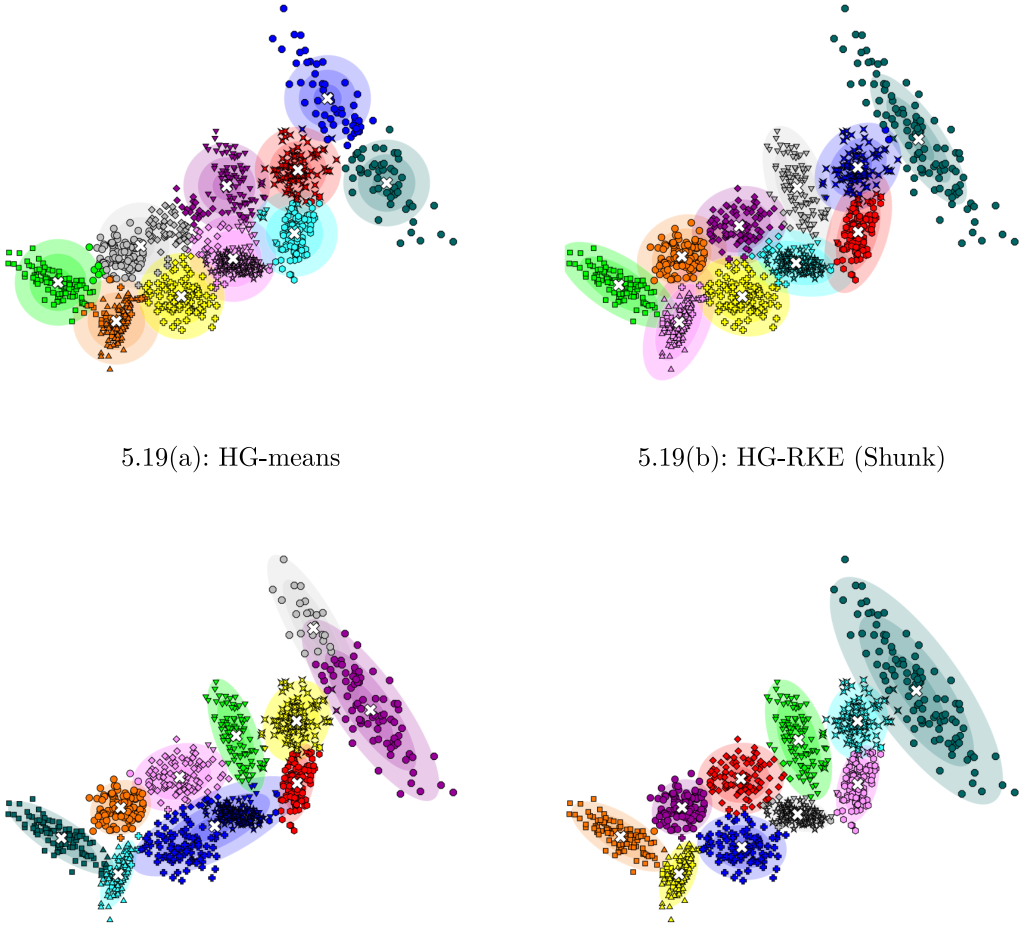
c	d	k: 3 and n: 1500			
		HG-means	HG-RKE (Shrunk)	GMM	HG-RGMM (Shrunk)
0.01	50	0.79	**1.00**	0.92	**1.00**
0.01	75	0.79	0.99	0.70	**1.00**
0.01	100	0.78	0.95	0.43	**1.00**
0.01	125	0.78	0.62	0.17	**0.87**

Table 5.5: Comparison between the selected methods (HG-means, HG-RKE (Shrunk), GMM and HG-RGMM (Shrunk)) with some meaningful instances (m : 500).

Figure 5.19 illustrates the results of the best methods in an instance where the number of clusters is 10.

Dataset	n	d	k	HG-means	HG-RKE (Shrunk)	GMM	HG-RGMM (Shrunk)
breast	569	30	2	0.49	0.52	**0.71**	0.02
egg	12811	11	10	0.05	0.09	**0.10**	**0.10**
glass	214	9	6	0.26	**0.28**	0.21	0.23
heart	270	13	2	0.03	0.06	**0.07**	0.02
ionosphere	351	34	2	0.18	**0.77**	0.31	0.74
iris	150	4	3	0.73	**0.94**	0.90	**0.94**
libras	360	90	15	0.31	0.31	0.25	**0.33**
magic	19020	10	2	0.06	0.04	0.07	**0.11**
page blocks	5473	10	5	0.01	**0.09**	0.06	0.01
pendigits	10995	16	10	0.53	0.73	0.51	**0.78**
recognition	20000	16	26	0.13	0.08	0.16	**0.24**
seeds	210	7	3	0.72	0.72	**0.83**	0.72
segmentation	2310	19	7	0.34	**0.50**	0.45	0.43
skin	245057	3	2	-0.04	**0.74**	0.44	0.08
study	403	5	4	0.30	**0.58**	0.30	0.49
wholesale	440	6	6	0.11	0.11	0.11	**0.15**
wines	178	13	3	0.37	0.40	**0.84**	0.36
yeast	1484	8	10	0.15	0.16	0.06	**0.20**

Table 5.6: Comparison between the selected methods (HG-means, HG-RKE (Shrunk), GMM and HG-RGMM (Shrunk)) with some UCI datasets [2].



5.19(a): HG-means

5.19(b): HG-RKE (Shunk)

5.19(c): GMM

5.19(d): HG-GMM (Shrunk)

Figure 5.19: The final results of a subset of the proposed methods in a $(n : 1000, k : 10, d : 2, c : 0.01)$ instance.

6
Conclusions and Future Work

In this work, we study the general problem of clustering. Several mathematical problem formulations are investigated. Out of these, two are extensively studied in the literature with formal mathematical formulations: the minimum sum-of-squares clustering (MSSC); and the maximum likelihood estimation (MLE) using Gaussian distributions (GMM). The ellipsoidal clustering with hard assignments had never been formally detailed as a mathematical optimization problem, notwithstanding the fact that previous works presented the problem and proposed algorithmic heuristics to solve it. This work presents a mathematical optimization formulation for the ellipsoidal minimum sum of squares clustering (EMSSC) problem. We also extensively compared different the two assignment types and the cluster shapes.

The simple definition and implementation of the k-means made it one of the most popular algorithms in unsupervised clustering though it has a significant limitation when considering the cluster's shapes: when working with real-world datasets, it is not common to find clusters with a perfectly spherical shape. Meanwhile, ellipsoidal methodologies have a larger domain of possible solutions, due to the considerable amount of parameters in the problem. Even though these methodologies have great potential since they are a generalization of the spherical algorithms, on account of their higher degrees of freedom, they demand a better algorithm to search for solutions.

Several ellipsoidal algorithms have been proposed in this work with the intention to improve the results of the clustering problem. The tested methods combine regularization techniques with a Random Swap and a Hybrid Genetic algorithm. In order to compare all the different algorithms, a broad benchmark of datasets was generated; this way, the performance of each one could be tested under different conditions.

The conclusions drawn from the results are as follows. Spherical methods have a limitation that prevents them from achieving excellent results in instances with low separability. Using ellipsoidal algorithms significantly improved the results, but only when there are sufficient observations to estimate the shape matrices correctly. Furthermore, achieving excellent results in high dimension instances can be challenging for ellipsoidal methods because of the number of parameters that need to be estimated. When the clusters have few observations in high dimension, there is over-fitting, and the solution diverges, compromising the results. Using regularization techniques restrict the shape matrices increasing the lower bound of its eigenvalues. Moreover, these matrices become more well-conditioned. On the tests, they accomplished good results with both assignments (hard and soft) although the soft assignments achieved better results due to the higher number of parameters estimated during their iterations. Since hard methods have a limitation on the size of their shape matrices, meta-heuristics efficiently escaped from a local minimum. Soft methods do not have this control over their formulation so that the matrices can have different sizes. It has been shown that meta-heuristics in the soft algorithms require regularization of the matrices to work correctly.

The random swap with regularization methodology has a straightforward implementation and achieved excellent results. Furthermore, hybrid genetic algorithms with regularizations worked better on more challenging instances. Both regularizations with meta-heuristics achieve good results when the instances are not so challenging. However, when the number of clusters increases, the Shrunk regularization obtained significantly improved results in comparison with others.

The regularizations OAS and LW calculate the parameter δ adaptively, which is just another variation of the method. When there is a meta-heuristic that finds good solutions, a regularization that minimally changes the matrices is seems to work better. Shrunk fulfills this role, therefore the effectiveness of the HG-RGMM (Shunk).

A significant result is that both hard and soft ellipsoidal methods performed better than the spherical methods, especially after the search improvements, even when fewer points were present in the dataset. In tests with the artificial datasets, the soft ellipsoidal algorithms outperformed the hard ellipsoidal.

Regarding future works, the inclusion of an adaptive regularization into the GMM problem is suggested. Is also suggested, to analyze the behavior of ellipsoidal methods in higher dimensions. The use of better data structures to improve the performance of the ellipsoidal methods should translate into shorter CPU times and can also be investigated. Both meta-heuristics still require a more careful calibration. In order to allow for a consistent comparison between the future works and the results presented in this document, Julia packages will be made available as well as a testbench of datasets.

Bibliography

[1] **Infographic: What happens in an internet minute in 2018?** http://www.visualcapitalist.com/internet-minute-2018/. Accessed: 2018-08-10.

[2] DHEERU, D.; KARRA TANISKIDOU, E.. **UCI machine learning repository**, 2017.

[3] SEGARAN, T.; HAMMERBACHER, J.. **Beautiful data: the stories behind elegant data solutions**. " O'Reilly Media, Inc.", 2009.

[4] REINSEL, D.; GANTZ, J. ; RYDNING, J.. **Data age 2025: The evolution of data to life-critical**. Don't Focus on Big Data, 2017.

[5] HILBERT, M.; LÓPEZ, P.. **The world's technological capacity to store, communicate, and compute information**. Science, p. 1200970, 2011.

[6] WITTEN, I. H.; FRANK, E.; HALL, M. A. ; PAL, C. J.. **Data Mining: Practical machine learning tools and techniques**. Morgan Kaufmann, 2016.

[7] JOHN WALKER, S.. **Big data: A revolution that will transform how we live, work, and think**, 2014.

[8] **Fuel of the future; the data economy**. The Economist (US), May 2017.

[9] NASRABADI, N. M.. **Pattern recognition and machine learning**. Journal of Electronic Imaging, 16(4):729, 2007.

[10] MITCHELL, T. M.; OTHERS. **Machine learning. 1997**. Burr Ridge, IL: McGraw Hill, 45(37):870–877, 1997.

[11] FREUND, Y.; SCHAPIRE, R. E.. **A decision-theoretic generalization of on-line learning and an application to boosting**. Journal of computer and system sciences, 55(1):119–139, 1997.

[12] NG, A. Y.; JORDAN, M. I. ; WEISS, Y.. **On spectral clustering: Analysis and an algorithm.** In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 849–856, 2002.

[13] BOSER, B. E.; GUYON, I. M. ; VAPNIK, V. N.. **A training algorithm for optimal margin classifiers.** In: PROCEEDINGS OF THE FIFTH ANNUAL WORKSHOP ON COMPUTATIONAL LEARNING THEORY, p. 144–152. ACM, 1992.

[14] HASTIE, T.; TIBSHIRANI, R. ; FRIEDMAN, J.. **Unsupervised learning.** In: THE ELEMENTS OF STATISTICAL LEARNING, p. 485–585. Springer, 2009.

[15] CHEN, M.; MAO, S. ; LIU, Y.. **Big data: A survey.** Mobile Networks and Applications, 19(2):171–209, 2014.

[16] HEARST, M. A.; DUMAIS, S. T.; OSUNA, E.; PLATT, J. ; SCHOLKOPF, B.. **Support vector machines.** IEEE Intelligent Systems and their applications, 13(4):18–28, 1998.

[17] HO, T. K.. **Random decision forests.** In: DOCUMENT ANALYSIS AND RECOGNITION, 1995., PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON, volumen 1, p. 278–282. IEEE, 1995.

[18] LLOYD, S.. **Least squares quantization in pcm.** IEEE Transactions on Information Theory, 28(2):129–137, 1982.

[19] FREY, B. J.; DUECK, D.. **Clustering by passing messages between data points.** Science, 315(5814):972–976, 2007.

[20] GLOVER, F. W.; KOCHENBERGER, G. A.. **Handbook of metaheuristics**, volumen 57. Springer Science & Business Media, 2006.

[21] FRÄNTI, P.; KIVIJÄRVI, J.. **Randomised local search algorithm for the clustering problem.** 3:358–369, 01 2000.

[22] GRIBEL, D.; VIDAL, T.. **Hg-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering.** CoRR, abs/1804.09813, 2018.

[23] BEZANSON, J.; KARPINSKI, S.; SHAH, V. B. ; EDELMAN, A.. **Julia: A fast dynamic language for technical computing.** CoRR, abs/1209.5145, 2012.

[24] TAN, P.-N.; OTHERS. **Introduction to data mining**. Pearson Education India, 2006.

[25] ESTIVILL-CASTRO, V.. **Why so many clustering algorithms: a position paper**. ACM SIGKDD Explorations Newsletter, 4(1):65–75, 2002.

[26] ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. ; OTHERS. **A density-based algorithm for discovering clusters in large spatial databases with noise.** In: KDD, volumen 96, p. 226–231, 1996.

[27] **Sigkdd news : 2014 sigkdd test of time award**. `http://www.kdd.org/News/view/2014-sigkdd-test-of-time-award`. (Accessed on 08/05/2018).

[28] MACQUEEN, J.; OTHERS. **Some methods for classification and analysis of multivariate observations**. In: PROCEEDINGS OF THE FIFTH BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY, volumen 1, p. 281–297. Oakland, CA, USA, 1967.

[29] VINOD, H. D.. **Integer programming and the theory of grouping**. Journal of the American Statistical Association, 64(326):506–519, 1969.

[30] SHERALI, H. D.; DESAI, J.. **A global optimization rlt-based approach for solving the hard clustering problem**. Journal of Global Optimization, 32(2):281–306, 2005.

[31] ALOISE, D.; HANSEN, P. ; LIBERTI, L.. **An improved column generation algorithm for minimum sum-of-squares clustering**. Mathematical Programming, 131(1-2):195–220, 2012.

[32] BAGIROV, A. M.. **Modified global k-means algorithm for minimum sum-of-squares clustering problems**. Pattern Recognition, 41(10):3192–3199, 2008.

[33] XU, R.; WUNSCH, D.. **Survey of clustering algorithms**. IEEE Transactions on Neural Networks, 16(3):645–678, 2005.

[34] JAIN, A. K.. **Data clustering: 50 years beyond k-means**. Pattern Recognition Letters, 31(8):651–666, 2010.

[35] GAN, G.; MA, C. ; WU, J.. **Data clustering: theory, algorithms, and applications**, volumen 20. Siam, 2007.

[36] ALOISE, D.; DESHPANDE, A.; HANSEN, P. ; POPAT, P.. **Np-hardness of euclidean sum-of-squares clustering**. Machine Learning, 75(2):245–248, 2009.

[37] INABA, M.; KATOH, N. ; IMAI, H.. **Applications of weighted voronoi diagrams and randomization to variance-based k-clustering**. In: PROCEEDINGS OF THE TENTH ANNUAL SYMPOSIUM ON COMPUTATIONAL GEOMETRY, p. 332–339. ACM, 1994.

[38] KANUNGO, T.; MOUNT, D. M.; NETANYAHU, N. S.; PIATKO, C. D.; SILVERMAN, R. ; WU, A. Y.. **An efficient k-means clustering algorithm: Analysis and implementation**. IEEE Transactions on Pattern Analysis & Machine Intelligence, (7):881–892, 2002.

[39] BLUM, A.; HOPCROFT, J. ; KANNAN, R.. **Foundations of data science**. Vorabversion Eines Lehrbuchs, 2016.

[40] DEMPSTER, A. P.; LAIRD, N. M. ; RUBIN, D. B.. **Maximum likelihood from incomplete data via the em algorithm**. Journal of the Royal Statistical Society. Series B (methodological), p. 1–38, 1977.

[41] WU, C. J.. **On the convergence properties of the em algorithm**. The Annals of Statistics, p. 95–103, 1983.

[42] XU, L.; JORDAN, M. I.. **On convergence properties of the em algorithm for gaussian mixtures**. Neural computation, 8(1):129–151, 1996.

[43] BILMES, J. A.; OTHERS. **A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models**. International Computer Science Institute, 4(510):126, 1998.

[44] MCLACHLAN, G.; KRISHNAN, T.. **The EM algorithm and extensions**, volumen 382. John Wiley & Sons, 2007.

[45] WAGSTAFF, K.; CARDIE, C.; ROGERS, S.; SCHRÖDL, S. ; OTHERS. **Constrained k-means clustering with background knowledge**. In: ICML, volumen 1, p. 577–584, 2001.

[46] BOUVEYRON, C.; GIRARD, S. ; SCHMID, C.. **High-dimensional data clustering**. Computational Statistics & Data Analysis, 52(1):502–519, 2007.

[47] CERIOLI, A.. **K-means cluster analysis and mahalanobis metrics: a problematic match or an overlooked opportunity**. Statistica Applicata, 17(1), 2005.

[48] MORALES-ESTEBAN, A.; MARTÍNEZ-ÁLVAREZ, F.; SCITOVSKI, S. ; SCITOVSKI, R.. **A fast partitioning algorithm using adaptive mahalanobis clustering with application to seismic zoning**. Computers & Geosciences, 73:132–141, 2014.

[49] MAROŠEVIĆ, T.; SCITOVSKI, R.. **Multiple ellipse fitting by center-based clustering**. Croatian Operational Research Review, 6(1):43–53, 2015.

[50] STRANG, G.; STRANG, G.; STRANG, G. ; STRANG, G.. **Introduction to linear algebra**, volumen 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[51] GOLUB, G. H.; VAN LOAN, C. F.. **Matrix computations**, volumen 3. JHU Press, 2012.

[52] LIU, Y.; ZHANG, H. H. ; WU, Y.. **Hard or soft classification? large-margin unified machines**. Journal of the American Statistical Association, 106(493):166–177, 2011.

[53] FRIENDLY, M.; MONETTE, G.; FOX, J. ; OTHERS. **Elliptical insights: understanding statistical methods through elliptical geometry**. Statistical Science, 28(1):1–39, 2013.

[54] BANFIELD, J. D.; RAFTERY, A. E.. **Model-based gaussian and non-gaussian clustering**. Biometrics, p. 803–821, 1993.

[55] MAHALANOBIS, P. C.. **On the generalized distance in statistics**. National Institute of Science of India, 1936.

[56] HAWKINS, D. M.. **The problem of overfitting**. Journal of Chemical Information and Computer Sciences, 44(1):1–12, 2004.

[57] EVERITT, B.; SKRONDAL, A.. **The cambridge dictionary of statistics 2002**. Cambridge, Cambridge.

[58] LEDOIT, O.; WOLF, M.. **A well-conditioned estimator for large-dimensional covariance matrices**. Journal of Multivariate Analysis, 88(2):365–411, 2004.

[59] CHEN, Y.; WIESEL, A.; ELDAR, Y. C. ; HERO, A. O.. **Shrinkage algorithms for mmse covariance estimation**. IEEE Transactions on Signal Processing, 58(10):5016–5029, 2010.

[60] BLUM, C.; ROLI, A.. **Metaheuristics in combinatorial optimization: Overview and conceptual comparison**. ACM Computing Surveys (CSUR), 35(3):268–308, 2003.

[61] LOURENÇO, H. R.; MARTIN, O. C. ; STÜTZLE, T.. **Iterated local search**. In: HANDBOOK OF METAHEURISTICS, p. 320–353. Springer, 2003.

[62] FRÄNTI, P.. **Efficiency of random swap clustering**. Journal of Big Data, 5(1):13, 2018.

[63] ARTHUR, D.; VASSILVITSKII, S.. **k-means++: The advantages of careful seeding**. In: PROCEEDINGS OF THE EIGHTEENTH ANNUAL ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS, p. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[64] BEZDEK, J. C.; EHRLICH, R. ; FULL, W.. **Fcm: The fuzzy c-means clustering algorithm**. Computers & Geosciences, 10(2-3):191–203, 1984.

[65] DUNDAR, M. M.; LANDGREBE, D.. **A model-based mixture-supervised classification approach in hyperspectral data analysis**. IEEE Transactions on Geoscience and Remote Sensing, 40(12):2692–2699, 2002.

[66] HALBE, Z.; BORTMAN, M. ; ALADJEM, M.. **Regularized mixture density estimation with an analytical setting of shrinkage intensities**. IEEE Transactions on Neural Networks and Learning Systems, 24(3):460–470, 2013.

[67] ZHAO, Q.; HAUTAMÄKI, V.; KÄRKKÄINEN, I. ; FRÄNTI, P.. **Random swap em algorithm for gaussian mixture models**. Pattern Recognition Letters, 33(16):2120–2126, 2012.

[68] PERNKOPF, F.; BOUCHAFFRA, D.. **Genetic-based em algorithm for learning gaussian mixture models**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1344–1348, 2005.

[69] BORMAN, S.. **The expectation maximization algorithm-a short tutorial**. 2004.

[70] BERKHIN, P.. **A survey of clustering data mining techniques**. In: GROUPING MULTIDIMENSIONAL DATA, p. 25–71. Springer, 2006.

[71] PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M. ; DUCHESNAY, E.. **Scikit-learn: Machine learning in Python**. Journal of Machine Learning Research, 12:2825–2830, 2011.

[72] MICHALEWICZ, Z.; FOGEL, D. B.. **How to solve it: modern heuristics**. Springer Science & Business Media, 2013.

[73] FOGEL, D. B.. **Evolutionary algorithms in theory and practice**. Complexity, 2(4):26–27, 1997.

[74] GOLUB, G. H.; VAN LOAN, C. F.. **Matrix computations. 1996**. Johns Hopkins University, Press, Baltimore, MD, USA, p. 374–426, 1996.

[75] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. ; STEIN, C.. **Introduction to algorithms**. MIT press, 2009.

[76] LAWSON, C. L.; HANSON, R. J.; KINCAID, D. R. ; KROGH, F. T.. **Basic linear algebra subprograms for fortran usage**. ACM Transactions on Mathematical Software (TOMS), 5(3):308–323, 1979.

[77] POOLE, D.. **Linear algebra: A modern introduction**. Cengage Learning, 2014.

[78] MOORE, A. W.. **Very fast em-based mixture model clustering using multiresolution kd-trees**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 543–549, 1999.

[79] ZIVKOVIC, Z.. **Improved adaptive gaussian mixture model for background subtraction**. In: PATTERN RECOGNITION, 2004. ICPR 2004. PROCEEDINGS OF THE 17TH INTERNATIONAL CONFERENCE ON, volumen 2, p. 28–31. IEEE, 2004.

[80] GOLDBERG, D. E.; DEB, K.. **A comparative analysis of selection schemes used in genetic algorithms**. In: FOUNDATIONS OF GENETIC ALGORITHMS, volumen 1, p. 69–93. Elsevier, 1991.

[81] KUHN, H. W.. **The hungarian method for the assignment problem**. Naval Research Logistics Quarterly, 2(1-2):83–97, 1955.

[82] QIU, W.; JOE., H.. **clusterGeneration: Random Cluster Generation (with Specified Degree of Separation)**, 2015. R package version 1.3.4.

[83] R CORE TEAM. **R: A Language and Environment for Statistical Computing**. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[84] QIU, W.; JOE, H.. **Generation of random clusters with specified degree of separation**. Journal of Classification, 23(2):315–334, 2006.

[85] QIU, W.; JOE, H.. **Separation index and partial membership for clustering**. Computational statistics & data analysis, 50(3):585–603, 2006.

[86] HUBERT, L.; ARABIE, P.. **Comparing partitions**. Journal of Classification, 2(1):193–218, 1985.

[87] RAND, W. M.. **Objective criteria for the evaluation of clustering methods**. Journal of the American Statistical Association, 66(336):846–850, 1971.

[88] VINH, N. X.; EPPS, J. ; BAILEY, J.. **Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance**. Journal of Machine Learning Research, 11(Oct):2837–2854, 2010.

[89] STEINLEY, D.. **Properties of the hubert-arable adjusted rand index**. Psychological Methods, 9(3):386, 2004.

[90] FRÄNTI, P.; REZAEI, M. ; ZHAO, Q.. **Centroid index: cluster level similarity measure**. Pattern Recognition, 47(9):3034–3045, 2014.

A
Benchmark Tables

This Appendix provides the results of the research with $m \in \{100\}$. Beyond the ARI, these table provides another similarity measure, the Centroid Index, described in Fränti *et al.* work [90].

			k-means			RS-means			HG-means			kmeans++			KE			RS-KE			HG-KE		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
3	-0.26	2	0.46	0.09	0.02	0.46	0.09	0.02	0.46	0.09	0.02	0.46	0.09	0.03	0.45	0.12	0.06	0.43	0.13	0.06	0.42	0.13	0.07
3	-0.26	5	0.36	0.12	0.07	0.36	0.11	0.07	0.36	0.11	0.07	0.36	0.11	0.06	0.47	0.14	0.01	0.51	0.16	0.02	0.51	0.16	0.03
3	-0.26	10	0.32	0.11	0.12	0.32	0.11	0.10	0.32	0.11	0.10	0.33	0.11	0.09	0.42	0.17	0.09	0.59	0.16	0.01	0.60	0.15	0.01
3	-0.26	20	0.25	0.11	0.11	0.26	0.12	0.14	0.26	0.11	0.14	0.25	0.12	0.09	0.21	0.12	0.29	0.58	0.22	0.06	0.54	0.21	0.04
3	-0.26	30	0.20	0.12	0.21	0.20	0.12	0.21	0.20	0.12	0.23	0.19	0.12	0.18	0.08	0.06	0.48	0.29	0.17	0.37	0.30	0.14	0.31
3	-0.26	40	0.17	0.11	0.21	0.18	0.12	0.21	0.18	0.11	0.19	0.17	0.11	0.21	0.04	0.03	0.45	0.10	0.13	0.61	0.12	0.11	0.64
3	-0.26	50	0.14	0.10	0.21	0.16	0.10	0.22	0.16	0.10	0.21	0.13	0.10	0.22	0.03	0.03	0.48	0.05	0.07	0.58	0.07	0.06	0.66
3	-0.26	60	0.13	0.08	0.23	0.14	0.09	0.17	0.14	0.10	0.26	0.13	0.09	0.26	0.02	0.02	0.45	0.04	0.04	0.63	0.06	0.05	0.65
3	-0.26	70	0.10	0.07	0.33	0.11	0.07	0.35	0.12	0.08	0.29	0.10	0.08	0.32	0.02	0.01	0.45	0.05	0.06	0.59	0.07	0.06	0.64
3	-0.10	2	0.69	0.11	0.02	0.69	0.11	0.02	0.69	0.11	0.02	0.69	0.11	0.02	0.73	0.07	0.00	0.74	0.06	0.00	0.74	0.06	0.00
3	-0.10	5	0.65	0.10	0.01	0.64	0.10	0.01	0.64	0.10	0.01	0.65	0.10	0.00	0.75	0.08	0.00	0.77	0.07	0.00	0.77	0.06	0.00
3	-0.10	10	0.61	0.14	0.03	0.61	0.14	0.03	0.61	0.14	0.03	0.61	0.14	0.03	0.70	0.15	0.03	0.81	0.08	0.00	0.81	0.08	0.00
3	-0.10	20	0.60	0.12	0.01	0.60	0.11	0.02	0.60	0.11	0.02	0.59	0.12	0.01	0.44	0.18	0.14	0.81	0.11	0.00	0.79	0.12	0.01
3	-0.10	30	0.56	0.13	0.02	0.58	0.12	0.03	0.58	0.12	0.03	0.56	0.13	0.03	0.21	0.11	0.29	0.51	0.21	0.17	0.53	0.17	0.08
3	-0.10	40	0.55	0.13	0.01	0.57	0.12	0.03	0.57	0.12	0.04	0.56	0.13	0.02	0.09	0.07	0.46	0.21	0.17	0.45	0.26	0.16	0.36
3	-0.10	50	0.52	0.14	0.02	0.54	0.13	0.02	0.54	0.14	0.02	0.52	0.14	0.03	0.06	0.04	0.30	0.11	0.12	0.52	0.13	0.11	0.48
3	-0.10	60	0.46	0.16	0.02	0.49	0.14	0.03	0.49	0.15	0.04	0.46	0.16	0.01	0.05	0.04	0.52	0.08	0.06	0.59	0.09	0.06	0.62
3	-0.10	70	0.44	0.16	0.05	0.47	0.15	0.03	0.47	0.13	0.02	0.46	0.15	0.05	0.05	0.04	0.54	0.07	0.06	0.50	0.11	0.07	0.58
3	0.01	2	0.82	0.09	0.01	0.82	0.09	0.01	0.82	0.09	0.01	0.82	0.09	0.01	0.87	0.05	0.00	0.87	0.04	0.00	0.87	0.04	0.00
3	0.01	5	0.81	0.10	0.00	0.81	0.10	0.00	0.81	0.10	0.00	0.80	0.10	0.00	0.89	0.05	0.00	0.90	0.04	0.00	0.90	0.04	0.00
3	0.01	10	0.79	0.10	0.01	0.79	0.10	0.01	0.79	0.10	0.01	0.79	0.10	0.01	0.86	0.11	0.01	0.91	0.05	0.00	0.91	0.05	0.00
3	0.01	20	0.79	0.11	0.01	0.79	0.11	0.01	0.79	0.11	0.01	0.79	0.12	0.01	0.66	0.19	0.03	0.92	0.08	0.00	0.92	0.07	0.00
3	0.01	30	0.73	0.17	0.03	0.74	0.16	0.04	0.74	0.16	0.04	0.74	0.15	0.02	0.34	0.16	0.27	0.71	0.22	0.12	0.73	0.20	0.11
3	0.01	40	0.75	0.14	0.02	0.75	0.14	0.02	0.75	0.13	0.02	0.75	0.13	0.02	0.20	0.13	0.32	0.32	0.19	0.38	0.42	0.18	0.29
3	0.01	50	0.71	0.17	0.04	0.72	0.17	0.05	0.72	0.17	0.05	0.71	0.17	0.02	0.11	0.08	0.47	0.18	0.14	0.50	0.20	0.14	0.60
3	0.01	60	0.74	0.14	0.02	0.76	0.13	0.02	0.75	0.13	0.01	0.75	0.12	0.00	0.09	0.06	0.52	0.12	0.10	0.47	0.15	0.10	0.56
3	0.01	70	0.69	0.17	0.05	0.71	0.17	0.05	0.71	0.16	0.04	0.70	0.17	0.03	0.09	0.07	0.48	0.11	0.08	0.54	0.15	0.08	0.59
3	0.21	2	0.97	0.05	0.00	0.97	0.05	0.00	0.97	0.05	0.00	0.97	0.05	0.00	0.99	0.02	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	5	0.97	0.06	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.99	0.02	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	10	0.97	0.04	0.00	0.97	0.04	0.00	0.97	0.04	0.00	0.97	0.04	0.00	0.98	0.04	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	20	0.97	0.05	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.91	0.12	0.02	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	30	0.95	0.11	0.00	0.96	0.10	0.00	0.96	0.10	0.00	0.96	0.10	0.01	0.69	0.21	0.10	0.96	0.12	0.04	0.96	0.11	0.04
3	0.21	40	0.96	0.10	0.01	0.96	0.11	0.01	0.96	0.10	0.01	0.95	0.11	0.01	0.53	0.19	0.16	0.82	0.23	0.11	0.85	0.20	0.10
3	0.21	50	0.97	0.04	0.00	0.97	0.03	0.00	0.97	0.03	0.00	0.97	0.05	0.00	0.40	0.20	0.27	0.60	0.32	0.23	0.66	0.28	0.26
3	0.21	60	0.97	0.04	0.00	0.97	0.04	0.00	0.97	0.04	0.00	0.97	0.05	0.00	0.33	0.17	0.31	0.37	0.24	0.29	0.53	0.25	0.39
3	0.21	70	0.97	0.06	0.00	0.97	0.05	0.00	0.97	0.05	0.00	0.97	0.06	0.00	0.30	0.16	0.44	0.24	0.14	0.40	0.35	0.18	0.47

			k-means			RS-means			HG-means			kmeans++			KE			RS-KE			HG-KE		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
10	-0.26	2	0.46	0.04	0.59	0.46	0.04	0.61	0.46	0.04	0.59	0.46	0.04	0.60	0.44	0.05	0.82	0.42	0.06	1.31	0.43	0.05	1.11
10	-0.26	5	0.30	0.04	0.72	0.31	0.04	0.71	0.31	0.04	0.66	0.30	0.04	0.78	0.28	0.05	1.38	0.32	0.07	1.25	0.34	0.07	1.04
10	-0.26	10	0.22	0.04	0.94	0.23	0.05	0.72	0.23	0.05	0.68	0.22	0.05	0.85	0.16	0.04	1.86	0.27	0.08	1.34	0.26	0.07	1.32
10	-0.26	20	0.16	0.04	1.45	0.19	0.04	1.15	0.19	0.04	1.21	0.17	0.04	1.39	0.08	0.02	2.55	0.16	0.06	2.60	0.15	0.05	2.10
10	-0.26	30	0.13	0.04	1.83	0.16	0.05	1.61	0.16	0.05	1.50	0.13	0.04	1.76	0.04	0.01	2.90	0.05	0.03	3.11	0.06	0.02	2.62
10	-0.26	40	0.10	0.03	2.27	0.12	0.04	2.03	0.12	0.04	2.02	0.10	0.03	2.15	0.03	0.01	3.07	0.02	0.02	4.15	0.03	0.01	2.93
10	-0.10	2	0.71	0.04	0.04	0.71	0.04	0.03	0.71	0.04	0.03	0.71	0.04	0.03	0.69	0.06	0.23	0.74	0.04	0.01	0.74	0.04	0.00
10	-0.10	5	0.59	0.04	0.03	0.60	0.04	0.03	0.60	0.04	0.03	0.59	0.04	0.04	0.55	0.07	0.76	0.67	0.05	0.03	0.68	0.04	0.01
10	-0.10	10	0.51	0.05	0.10	0.51	0.05	0.12	0.51	0.05	0.12	0.51	0.05	0.08	0.39	0.07	1.26	0.58	0.11	0.35	0.58	0.10	0.25
10	-0.10	20	0.47	0.07	0.32	0.48	0.07	0.31	0.48	0.07	0.27	0.48	0.06	0.26	0.21	0.05	2.00	0.33	0.09	2.01	0.37	0.08	1.45
10	-0.10	30	0.43	0.06	0.57	0.46	0.06	0.30	0.46	0.06	0.33	0.44	0.07	0.45	0.11	0.03	2.61	0.13	0.06	2.84	0.18	0.05	2.05
10	-0.10	40	0.38	0.08	0.99	0.42	0.08	0.74	0.42	0.09	0.74	0.38	0.08	0.86	0.06	0.02	3.02	0.05	0.03	4.18	0.08	0.03	2.67
10	0.01	2	0.85	0.04	0.03	0.85	0.04	0.03	0.85	0.04	0.03	0.85	0.04	0.02	0.87	0.04	0.04	0.88	0.02	0.00	0.88	0.02	0.00
10	0.01	5	0.79	0.03	0.00	0.79	0.03	0.00	0.79	0.03	0.00	0.79	0.03	0.00	0.80	0.06	0.14	0.85	0.02	0.00	0.85	0.03	0.00
10	0.01	10	0.72	0.04	0.04	0.72	0.04	0.02	0.72	0.04	0.02	0.72	0.04	0.06	0.62	0.09	0.79	0.81	0.06	0.09	0.80	0.07	0.08
10	0.01	20	0.71	0.06	0.08	0.71	0.06	0.04	0.71	0.07	0.05	0.71	0.06	0.02	0.38	0.08	1.72	0.50	0.08	1.59	0.51	0.10	1.29
10	0.01	30	0.67	0.08	0.20	0.68	0.10	0.27	0.67	0.11	0.25	0.68	0.08	0.17	0.19	0.05	2.51	0.22	0.09	2.50	0.21	0.07	2.51
10	0.01	40	0.68	0.06	0.19	0.69	0.07	0.15	0.69	0.06	0.14	0.69	0.05	0.08	0.12	0.04	2.66	0.09	0.04	3.78	0.10	0.04	3.32
10	0.21	2	0.96	0.05	0.15	0.98	0.03	0.01	0.98	0.03	0.01	0.98	0.03	0.01	0.97	0.05	0.10	0.99	0.01	0.00	0.99	0.01	0.00
10	0.21	5	0.97	0.03	0.02	0.97	0.02	0.00	0.97	0.02	0.00	0.97	0.02	0.00	0.98	0.03	0.05	0.99	0.01	0.00	0.99	0.01	0.00
10	0.21	10	0.96	0.03	0.00	0.96	0.02	0.00	0.96	0.02	0.00	0.96	0.03	0.00	0.90	0.07	0.35	0.98	0.01	0.00	0.98	0.02	0.01
10	0.21	20	0.95	0.04	0.03	0.95	0.03	0.00	0.95	0.04	0.00	0.95	0.04	0.00	0.68	0.08	1.21	0.78	0.10	0.91	0.83	0.12	0.55
10	0.21	30	0.94	0.04	0.06	0.95	0.03	0.00	0.95	0.03	0.00	0.95	0.03	0.00	0.50	0.09	2.00	0.52	0.10	1.70	0.55	0.12	1.69
10	0.21	40	0.93	0.06	0.13	0.95	0.04	0.00	0.95	0.04	0.00	0.95	0.04	0.01	0.33	0.08	2.39	0.26	0.10	2.65	0.26	0.09	2.39

			k-means			RS-means			HG-means			kmeans++			KE			RS-KE			HG-KE		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
20	-0.26	2	0.46	0.03	1.07	0.46	0.02	1.00	0.46	0.02	1.07	0.45	0.03	1.20	0.43	0.03	2.13	0.42	0.04	2.40	0.44	0.04	2.10
20	-0.26	5	0.29	0.02	2.80	0.29	0.01	2.13	0.29	0.02	2.20	0.29	0.02	2.27	0.25	0.02	3.27	0.29	0.05	3.30	0.30	0.04	3.00
20	-0.26	10	0.20	0.02	2.40	0.21	0.02	2.33	0.22	0.02	2.47	0.20	0.02	3.01	0.14	0.02	4.93	0.18	0.05	4.66	0.19	0.04	4.29
20	-0.26	20	0.12	0.02	3.80	0.14	0.02	2.73	0.14	0.02	2.80	0.12	0.02	3.76	0.05	0.01	5.47	0.07	0.02	6.24	0.08	0.01	5.20
20	-0.26	30	0.09	0.02	4.53	0.11	0.02	3.80	0.12	0.02	3.93	0.09	0.02	4.71	0.03	0.01	6.27	0.04	0.01	6.55	0.03	0.01	5.90
20	-0.26	40	0.07	0.02	5.53	0.09	0.02	4.33	0.09	0.03	4.33	0.07	0.02	5.15	0.02	0.01	6.60	0.02	0.01	9.50	0.02	0.01	8.20
20	-0.10	2	0.70	0.04	0.53	0.71	0.03	0.20	0.71	0.03	0.20	0.71	0.03	0.16	0.66	0.04	1.47	0.73	0.03	0.03	0.73	0.02	0.00
20	-0.10	5	0.57	0.03	0.60	0.59	0.02	0.20	0.59	0.02	0.20	0.58	0.03	0.35	0.48	0.03	2.47	0.64	0.05	0.36	0.63	0.05	0.37
20	-0.10	10	0.48	0.05	0.67	0.49	0.04	0.40	0.49	0.04	0.40	0.48	0.04	0.79	0.33	0.04	3.33	0.47	0.06	2.10	0.48	0.06	1.91
20	-0.10	20	0.39	0.05	1.07	0.41	0.04	0.20	0.41	0.04	0.27	0.39	0.04	1.18	0.16	0.02	4.47	0.21	0.04	5.21	0.22	0.04	4.57
20	-0.10	30	0.33	0.04	2.13	0.37	0.04	1.27	0.37	0.03	1.20	0.34	0.04	2.03	0.08	0.01	6.60	0.10	0.02	5.76	0.10	0.02	5.80
20	-0.10	40	0.31	0.06	2.27	0.35	0.04	1.13	0.34	0.04	1.67	0.31	0.05	2.46	0.05	0.01	6.40	0.06	0.02	9.30	0.06	0.01	7.57
20	0.01	2	0.81	0.05	0.80	0.85	0.02	0.07	0.85	0.02	0.07	0.85	0.03	0.12	0.81	0.05	1.00	0.87	0.01	0.00	0.88	0.02	0.00
20	0.01	5	0.77	0.02	0.33	0.78	0.02	0.00	0.78	0.02	0.00	0.78	0.03	0.06	0.72	0.04	1.13	0.84	0.02	0.00	0.84	0.02	0.00
20	0.01	10	0.70	0.04	0.27	0.71	0.03	0.07	0.71	0.03	0.07	0.71	0.03	0.09	0.54	0.06	2.60	0.73	0.05	0.81	0.74	0.04	0.60
20	0.01	20	0.63	0.04	0.53	0.64	0.03	0.27	0.64	0.03	0.27	0.64	0.03	0.21	0.31	0.06	4.33	0.39	0.06	4.12	0.42	0.06	3.54
20	0.01	30	0.60	0.04	0.93	0.62	0.03	0.27	0.62	0.04	0.33	0.60	0.06	0.73	0.17	0.03	5.27	0.21	0.05	5.27	0.21	0.04	5.00
20	0.01	40	0.58	0.05	1.07	0.61	0.04	0.40	0.62	0.03	0.27	0.59	0.04	0.73	0.09	0.02	5.73	0.10	0.02	8.00	0.11	0.02	6.54
20	0.21	2	0.89	0.04	1.33	0.98	0.01	0.00	0.98	0.01	0.00	0.98	0.01	0.01	0.91	0.03	1.07	0.99	0.01	0.00	0.99	0.01	0.00
20	0.21	5	0.92	0.04	0.73	0.96	0.02	0.00	0.96	0.02	0.00	0.97	0.01	0.00	0.93	0.03	0.60	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	10	0.93	0.04	0.47	0.96	0.02	0.00	0.96	0.02	0.00	0.96	0.02	0.04	0.86	0.04	1.07	0.97	0.02	0.11	0.97	0.02	0.03
20	0.21	20	0.92	0.03	0.20	0.94	0.02	0.00	0.94	0.02	0.00	0.94	0.02	0.00	0.64	0.05	2.93	0.72	0.05	2.49	0.78	0.06	1.71
20	0.21	30	0.91	0.04	0.27	0.93	0.03	0.00	0.93	0.03	0.00	0.93	0.04	0.08	0.49	0.04	4.20	0.53	0.06	3.59	0.59	0.07	2.92
20	0.21	40	0.89	0.04	0.60	0.94	0.02	0.00	0.94	0.02	0.00	0.92	0.03	0.13	0.33	0.04	5.33	0.32	0.06	5.20	0.39	0.05	3.66

			RKE (Shrunk)			RS-RKE (Shrunk)			HG-RKE (Shrunk)			RKE (OAS)			RS-RKE (OAS)			HG-RKE (OAS)			RKE (LW)			RS-RKE (LW)			HG-RKE (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
3	-0.26	2	0.46	0.11	0.05	0.46	0.12	0.04	0.45	0.12	0.03	0.45	0.12	0.05	0.43	0.12	0.05	0.43	0.13	0.04	0.45	0.12	0.07	0.43	0.13	0.05	0.43	0.13	0.05
3	-0.26	5	0.46	0.13	0.01	0.51	0.14	0.02	0.51	0.14	0.02	0.45	0.14	0.02	0.49	0.14	0.03	0.49	0.14	0.02	0.46	0.13	0.02	0.49	0.14	0.03	0.49	0.14	0.02
3	-0.26	10	0.40	0.16	0.13	0.58	0.14	0.00	0.58	0.14	0.00	0.37	0.14	0.09	0.51	0.13	0.01	0.51	0.14	0.01	0.38	0.14	0.06	0.51	0.14	0.01	0.51	0.14	0.00
3	-0.26	20	0.21	0.10	0.25	0.59	0.19	0.02	0.54	0.18	0.04	0.23	0.11	0.18	0.51	0.16	0.03	0.50	0.16	0.05	0.23	0.10	0.19	0.52	0.15	0.03	0.50	0.16	0.04
3	-0.26	30	0.10	0.06	0.39	0.29	0.16	0.12	0.30	0.13	0.09	0.13	0.10	0.37	0.39	0.16	0.03	0.36	0.15	0.07	0.13	0.10	0.40	0.40	0.17	0.09	0.35	0.17	0.08
3	-0.26	40	0.06	0.04	0.38	0.16	0.10	0.36	0.19	0.10	0.34	0.07	0.06	0.42	0.27	0.13	0.30	0.24	0.14	0.32	0.11	0.11	0.42	0.31	0.14	0.30	0.31	0.16	0.21
3	-0.26	50	0.04	0.03	0.41	0.09	0.06	0.46	0.13	0.07	0.48	0.05	0.03	0.37	0.22	0.11	0.52	0.18	0.09	0.59	0.07	0.08	0.38	0.24	0.11	0.48	0.19	0.12	0.50
3	-0.26	60	0.03	0.02	0.45	0.09	0.06	0.42	0.11	0.05	0.47	0.03	0.03	0.57	0.15	0.09	0.67	0.16	0.09	0.57	0.04	0.05	0.62	0.20	0.10	0.66	0.16	0.10	0.53
3	-0.26	70	0.03	0.02	0.47	0.08	0.06	0.47	0.11	0.06	0.39	0.03	0.02	0.47	0.10	0.07	0.58	0.12	0.08	0.55	0.03	0.04	0.59	0.15	0.09	0.78	0.12	0.09	0.60
3	-0.10	2	0.73	0.07	0.00	0.74	0.06	0.00	0.74	0.06	0.00	0.73	0.07	0.00	0.74	0.06	0.00	0.74	0.06	0.00	0.73	0.08	0.01	0.74	0.06	0.00	0.74	0.06	0.00
3	-0.10	5	0.74	0.08	0.00	0.76	0.07	0.00	0.76	0.07	0.00	0.74	0.08	0.00	0.75	0.08	0.00	0.75	0.08	0.00	0.74	0.08	0.01	0.75	0.08	0.00	0.75	0.08	0.00
3	-0.10	10	0.71	0.14	0.02	0.81	0.07	0.00	0.81	0.07	0.00	0.70	0.14	0.02	0.77	0.08	0.00	0.76	0.09	0.00	0.70	0.13	0.00	0.76	0.09	0.00	0.77	0.09	0.01
3	-0.10	20	0.46	0.17	0.08	0.81	0.09	0.00	0.80	0.11	0.00	0.53	0.17	0.04	0.71	0.11	0.00	0.72	0.10	0.00	0.53	0.18	0.05	0.72	0.11	0.00	0.74	0.10	0.00
3	-0.10	30	0.24	0.11	0.22	0.61	0.18	0.01	0.55	0.18	0.02	0.33	0.13	0.20	0.67	0.14	0.01	0.67	0.14	0.00	0.33	0.14	0.20	0.67	0.13	0.01	0.68	0.15	0.01
3	-0.10	40	0.13	0.07	0.34	0.31	0.14	0.11	0.34	0.11	0.11	0.20	0.10	0.33	0.44	0.15	0.06	0.44	0.17	0.15	0.22	0.11	0.28	0.48	0.16	0.10	0.49	0.18	0.08
3	-0.10	50	0.10	0.06	0.27	0.22	0.10	0.21	0.25	0.09	0.22	0.14	0.08	0.34	0.35	0.11	0.44	0.34	0.11	0.39	0.16	0.10	0.33	0.39	0.13	0.43	0.37	0.14	0.32
3	-0.10	60	0.07	0.05	0.46	0.17	0.08	0.30	0.21	0.09	0.28	0.09	0.06	0.52	0.30	0.10	0.57	0.28	0.12	0.57	0.10	0.09	0.51	0.34	0.09	0.64	0.33	0.11	0.51
3	-0.10	70	0.06	0.04	0.45	0.14	0.07	0.44	0.20	0.08	0.32	0.07	0.05	0.40	0.25	0.11	0.61	0.26	0.11	0.60	0.09	0.10	0.48	0.29	0.12	0.62	0.31	0.13	0.53
3	0.01	2	0.87	0.05	0.00	0.87	0.05	0.00	0.87	0.05	0.00	0.87	0.05	0.00	0.87	0.04	0.00	0.87	0.04	0.00	0.87	0.05	0.00	0.87	0.04	0.00	0.87	0.04	0.00
3	0.01	5	0.88	0.04	0.00	0.89	0.04	0.00	0.89	0.04	0.00	0.88	0.05	0.00	0.88	0.04	0.00	0.88	0.04	0.00	0.88	0.05	0.00	0.89	0.04	0.00	0.89	0.04	0.00
3	0.01	10	0.87	0.10	0.02	0.91	0.05	0.00	0.91	0.05	0.00	0.85	0.10	0.02	0.87	0.06	0.00	0.87	0.07	0.00	0.87	0.08	0.01	0.88	0.06	0.00	0.88	0.06	0.00
3	0.01	20	0.68	0.18	0.02	0.93	0.05	0.00	0.93	0.05	0.00	0.74	0.17	0.01	0.83	0.11	0.01	0.83	0.12	0.00	0.74	0.18	0.03	0.84	0.12	0.00	0.84	0.12	0.00
3	0.01	30	0.38	0.16	0.20	0.81	0.14	0.03	0.77	0.16	0.04	0.50	0.19	0.15	0.76	0.16	0.01	0.79	0.15	0.02	0.52	0.18	0.11	0.78	0.16	0.05	0.81	0.14	0.03
3	0.01	40	0.24	0.11	0.10	0.56	0.17	0.05	0.53	0.15	0.06	0.34	0.15	0.25	0.64	0.18	0.02	0.66	0.19	0.02	0.38	0.15	0.19	0.68	0.17	0.03	0.69	0.18	0.05
3	0.01	50	0.17	0.08	0.37	0.37	0.15	0.17	0.41	0.14	0.15	0.24	0.13	0.32	0.47	0.16	0.31	0.45	0.16	0.28	0.28	0.14	0.30	0.50	0.15	0.25	0.53	0.20	0.25
3	0.01	60	0.15	0.09	0.35	0.28	0.13	0.16	0.36	0.12	0.12	0.17	0.10	0.50	0.41	0.09	0.43	0.39	0.13	0.45	0.21	0.12	0.38	0.43	0.12	0.55	0.45	0.18	0.40
3	0.01	70	0.12	0.08	0.35	0.27	0.12	0.20	0.30	0.10	0.23	0.14	0.07	0.46	0.34	0.12	0.54	0.36	0.13	0.58	0.18	0.13	0.51	0.41	0.17	0.67	0.42	0.16	0.52
3	0.21	2	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	5	0.99	0.02	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.98	0.04	0.00	0.98	0.02	0.00	0.99	0.02	0.00	0.99	0.02	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	10	0.98	0.04	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.98	0.05	0.00	0.98	0.02	0.00	0.98	0.03	0.00	0.98	0.03	0.00	0.98	0.02	0.00	0.98	0.02	0.00
3	0.21	20	0.92	0.11	0.01	0.99	0.06	0.01	0.99	0.01	0.00	0.91	0.11	0.01	0.92	0.06	0.00	0.91	0.07	0.00	0.92	0.11	0.01	0.92	0.09	0.00	0.93	0.08	0.00
3	0.21	30	0.74	0.20	0.04	0.96	0.11	0.04	0.98	0.09	0.02	0.77	0.21	0.07	0.76	0.15	0.01	0.78	0.16	0.02	0.79	0.20	0.02	0.77	0.16	0.05	0.78	0.16	0.03
3	0.21	40	0.58	0.17	0.04	0.88	0.13	0.04	0.92	0.11	0.04	0.53	0.18	0.17	0.62	0.18	0.07	0.64	0.19	0.11	0.55	0.20	0.20	0.61	0.20	0.08	0.63	0.20	0.13
3	0.21	50	0.47	0.16	0.16	0.77	0.16	0.03	0.87	0.09	0.01	0.43	0.16	0.32	0.50	0.17	0.23	0.53	0.19	0.25	0.44	0.18	0.32	0.52	0.17	0.22	0.55	0.22	0.26
3	0.21	60	0.43	0.15	0.16	0.68	0.14	0.03	0.79	0.14	0.04	0.33	0.10	0.53	0.40	0.10	0.48	0.39	0.14	0.55	0.34	0.13	0.49	0.43	0.13	0.46	0.42	0.17	0.49
3	0.21	70	0.37	0.17	0.24	0.56	0.17	0.07	0.77	0.14	0.02	0.27	0.10	0.66	0.39	0.10	0.67	0.33	0.11	0.80	0.28	0.10	0.69	0.40	0.11	0.64	0.37	0.16	0.59

			RKE (Shrunk)			RS-RKE (Shrunk)			HG-RKE (Shrunk)			RKE (OAS)			RS-RKE (OAS)			HG-RKE (OAS)			RKE (LW)			RS-RKE (LW)			HG-RKE (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
10	-0.26	2	0.46	0.05	0.54	0.46	0.06	0.75	0.46	0.05	0.72	0.46	0.05	0.69	0.45	0.06	0.92	0.45	0.06	0.91	0.46	0.05	0.69	0.45	0.06	0.85	0.45	0.06	0.84
10	-0.26	5	0.28	0.04	1.21	0.34	0.06	0.80	0.35	0.06	0.72	0.29	0.04	1.22	0.33	0.06	0.96	0.33	0.06	0.94	0.29	0.05	1.15	0.33	0.05	0.88	0.34	0.06	0.89
10	-0.26	10	0.17	0.04	1.78	0.27	0.07	1.12	0.27	0.06	1.01	0.18	0.04	1.63	0.28	0.06	1.01	0.27	0.06	1.08	0.18	0.04	1.72	0.28	0.06	0.97	0.27	0.06	0.94
10	-0.26	20	0.09	0.03	2.40	0.14	0.05	2.11	0.16	0.05	1.93	0.11	0.03	2.36	0.24	0.08	1.30	0.20	0.06	1.61	0.12	0.03	2.08	0.25	0.07	1.16	0.20	0.05	1.53
10	-0.26	30	0.05	0.02	2.75	0.06	0.02	2.80	0.08	0.03	2.63	0.07	0.02	2.54	0.19	0.06	1.93	0.14	0.05	1.98	0.08	0.02	2.47	0.20	0.06	1.79	0.15	0.05	1.90
10	-0.26	40	0.03	0.01	2.86	0.03	0.01	2.91	0.04	0.01	2.77	0.05	0.01	2.75	0.14	0.05	2.12	0.10	0.04	2.40	0.06	0.02	2.86	0.16	0.05	2.05	0.12	0.04	2.19
10	-0.10	2	0.73	0.04	0.03	0.74	0.03	0.01	0.74	0.03	0.01	0.72	0.05	0.08	0.74	0.03	0.00	0.74	0.03	0.00	0.72	0.05	0.10	0.74	0.03	0.00	0.74	0.03	0.00
10	-0.10	5	0.58	0.07	0.36	0.68	0.04	0.00	0.68	0.04	0.00	0.59	0.07	0.38	0.67	0.04	0.00	0.67	0.04	0.01	0.59	0.07	0.37	0.68	0.04	0.00	0.67	0.04	0.01
10	-0.10	10	0.41	0.07	1.02	0.64	0.07	0.05	0.63	0.07	0.09	0.44	0.07	0.92	0.61	0.07	0.14	0.60	0.07	0.15	0.44	0.08	0.85	0.62	0.07	0.19	0.61	0.07	0.13
10	-0.10	20	0.24	0.05	1.84	0.40	0.08	1.24	0.43	0.08	1.04	0.30	0.07	1.63	0.58	0.09	0.38	0.54	0.08	0.32	0.31	0.06	1.56	0.59	0.08	0.46	0.53	0.08	0.46
10	-0.10	30	0.13	0.04	2.46	0.17	0.05	2.21	0.24	0.06	1.92	0.21	0.04	1.91	0.48	0.07	0.81	0.42	0.07	1.12	0.23	0.05	1.82	0.39	0.07	1.16	0.43	0.07	1.26
10	-0.10	40	0.07	0.02	2.74	0.09	0.03	2.65	0.12	0.04	2.43	0.14	0.04	2.22	0.39	0.07	1.45	0.31	0.07	1.67	0.18	0.05	1.99	0.31	0.07	1.71	0.36	0.07	1.52
10	0.01	2	0.87	0.04	0.04	0.88	0.02	0.00	0.88	0.02	0.00	0.87	0.04	0.08	0.88	0.02	0.00	0.88	0.02	0.00	0.87	0.04	0.04	0.88	0.02	0.00	0.88	0.02	0.00
10	0.01	5	0.82	0.05	0.06	0.85	0.02	0.00	0.85	0.02	0.00	0.83	0.04	0.04	0.85	0.02	0.00	0.85	0.02	0.00	0.83	0.04	0.06	0.85	0.02	0.00	0.85	0.02	0.00
10	0.01	10	0.67	0.09	0.60	0.83	0.04	0.01	0.83	0.05	0.02	0.69	0.08	0.51	0.81	0.05	0.03	0.81	0.05	0.01	0.69	0.09	0.51	0.81	0.05	0.03	0.81	0.05	0.02
10	0.01	20	0.43	0.07	1.42	0.73	0.09	0.27	0.67	0.08	0.51	0.52	0.07	1.12	0.76	0.07	0.15	0.76	0.07	0.13	0.53	0.07	1.11	0.76	0.08	0.30	0.76	0.07	0.19
10	0.01	30	0.26	0.06	2.09	0.51	0.09	1.21	0.44	0.08	1.45	0.38	0.07	1.53	0.66	0.10	0.45	0.64	0.08	0.57	0.42	0.07	1.41	0.65	0.08	0.89	0.64	0.08	0.82
10	0.01	40	0.16	0.04	2.37	0.37	0.07	1.78	0.29	0.06	1.86	0.30	0.06	1.81	0.56	0.07	1.10	0.55	0.07	1.15	0.35	0.08	1.66	0.58	0.07	1.14	0.57	0.06	1.15
10	0.21	2	0.97	0.04	0.10	0.99	0.01	0.00	0.99	0.01	0.00	0.97	0.04	0.11	0.99	0.01	0.00	0.99	0.01	0.00	0.97	0.05	0.15	0.99	0.01	0.00	0.99	0.01	0.00
10	0.21	5	0.98	0.04	0.06	0.99	0.01	0.00	0.99	0.01	0.00	0.98	0.03	0.03	0.99	0.01	0.00	0.99	0.01	0.00	0.98	0.03	0.03	0.99	0.01	0.00	0.99	0.01	0.00
10	0.21	10	0.94	0.06	0.16	0.98	0.01	0.00	0.98	0.02	0.00	0.94	0.06	0.16	0.97	0.02	0.00	0.98	0.02	0.00	0.95	0.05	0.12	0.98	0.03	0.03	0.98	0.02	0.02
10	0.21	20	0.75	0.07	0.84	0.95	0.05	0.09	0.95	0.04	0.02	0.78	0.07	0.66	0.91	0.04	0.02	0.93	0.04	0.03	0.79	0.07	0.69	0.90	0.07	0.23	0.92	0.06	0.14
10	0.21	30	0.59	0.07	1.52	0.84	0.07	0.35	0.80	0.07	0.41	0.66	0.07	1.13	0.78	0.07	0.42	0.82	0.07	0.33	0.67	0.08	1.16	0.74	0.07	0.87	0.81	0.08	0.50
10	0.21	40	0.44	0.08	1.99	0.67	0.08	0.99	0.66	0.08	0.84	0.56	0.08	1.43	0.67	0.07	0.89	0.73	0.07	0.82	0.58	0.08	1.49	0.65	0.06	1.17	0.70	0.07	0.90

			RKE (Shrunk)			RS-RKE (Shrunk)			HG-RKE (Shrunk)			RKE (OAS)			RS-RKE (OAS)			HG-RKE (OAS)			RKE (LW)			RS-RKE (LW)			HG-RKE (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
20	-0.26	2	0.45	0.03	2.00	0.46	0.03	1.39	0.47	0.03	1.17	0.45	0.03	1.80	0.46	0.03	1.58	0.46	0.03	1.63	0.44	0.03	2.10	0.45	0.04	1.78	0.46	0.03	1.48
20	-0.26	5	0.26	0.03	3.40	0.32	0.04	2.14	0.32	0.04	2.39	0.26	0.03	3.40	0.31	0.04	2.35	0.32	0.04	2.35	0.27	0.03	3.37	0.32	0.04	2.36	0.32	0.04	2.52
20	-0.26	10	0.15	0.02	4.60	0.21	0.04	3.54	0.21	0.04	3.61	0.17	0.03	4.20	0.22	0.04	3.37	0.22	0.04	3.26	0.16	0.02	3.99	0.22	0.04	3.17	0.23	0.03	3.24
20	-0.26	20	0.06	0.01	6.00	0.08	0.02	5.68	0.09	0.02	5.13	0.07	0.01	5.87	0.12	0.03	4.64	0.12	0.03	4.48	0.08	0.01	5.26	0.12	0.03	4.45	0.12	0.03	4.72
20	-0.26	30	0.03	0.01	6.27	0.05	0.02	6.31	0.05	0.01	6.19	0.05	0.01	5.80	0.09	0.02	5.30	0.09	0.02	5.27	0.05	0.01	5.65	0.09	0.02	5.00	0.10	0.02	5.07
20	-0.26	40	0.02	0.01	6.53	0.03	0.01	6.44	0.03	0.01	7.00	0.04	0.01	6.27	0.07	0.02	5.90	0.06	0.01	6.20	0.04	0.01	6.05	0.06	0.02	5.70	0.07	0.02	5.53
20	-0.10	2	0.70	0.03	0.93	0.74	0.02	0.04	0.74	0.02	0.04	0.69	0.03	0.93	0.74	0.02	0.03	0.74	0.02	0.00	0.68	0.03	0.96	0.74	0.02	0.00	0.75	0.03	0.00
20	-0.10	5	0.51	0.03	2.53	0.66	0.03	0.06	0.66	0.04	0.06	0.53	0.03	2.00	0.66	0.04	0.09	0.65	0.03	0.14	0.54	0.04	1.77	0.66	0.03	0.09	0.65	0.03	0.07
20	-0.10	10	0.35	0.03	3.33	0.57	0.06	0.67	0.56	0.05	0.85	0.37	0.04	3.00	0.57	0.05	0.57	0.55	0.04	0.66	0.40	0.04	2.71	0.57	0.05	0.54	0.55	0.05	0.67
20	-0.10	20	0.18	0.02	4.60	0.30	0.05	3.76	0.31	0.05	3.41	0.23	0.03	3.53	0.43	0.07	1.97	0.40	0.05	2.14	0.25	0.04	3.83	0.42	0.05	1.91	0.43	0.06	1.67
20	-0.10	30	0.09	0.01	6.60	0.17	0.04	4.90	0.17	0.03	4.69	0.14	0.03	5.07	0.26	0.04	3.45	0.30	0.05	3.11	0.17	0.03	4.47	0.34	0.05	3.34	0.31	0.05	3.33
20	-0.10	40	0.05	0.01	6.47	0.10	0.02	4.90	0.09	0.02	5.31	0.11	0.02	5.67	0.20	0.05	4.10	0.24	0.05	4.03	0.14	0.03	5.05	0.28	0.04	4.47	0.25	0.05	4.07
20	0.01	2	0.84	0.04	0.73	0.88	0.01	0.02	0.88	0.01	0.00	0.83	0.03	0.87	0.88	0.01	0.00	0.88	0.01	0.00	0.82	0.04	0.90	0.88	0.01	0.00	0.88	0.01	0.00
20	0.01	5	0.76	0.03	1.07	0.84	0.02	0.00	0.84	0.02	0.00	0.75	0.03	1.00	0.85	0.01	0.00	0.84	0.02	0.00	0.76	0.04	0.91	0.85	0.01	0.00	0.84	0.02	0.00
20	0.01	10	0.59	0.05	2.27	0.82	0.03	0.00	0.81	0.03	0.02	0.61	0.04	1.73	0.80	0.03	0.04	0.80	0.03	0.06	0.64	0.05	1.71	0.80	0.03	0.04	0.79	0.04	0.07
20	0.01	20	0.35	0.06	3.87	0.56	0.06	2.41	0.57	0.06	1.97	0.43	0.06	3.00	0.67	0.05	0.84	0.68	0.05	0.80	0.46	0.05	2.81	0.66	0.06	1.12	0.67	0.04	0.87
20	0.01	30	0.20	0.04	4.93	0.36	0.05	4.12	0.36	0.05	3.58	0.33	0.04	3.40	0.49	0.05	2.27	0.56	0.05	1.94	0.38	0.05	3.31	0.58	0.06	2.60	0.57	0.05	2.00
20	0.01	40	0.12	0.03	5.47	0.24	0.03	4.90	0.23	0.04	4.29	0.25	0.05	4.60	0.39	0.06	3.47	0.49	0.05	2.69	0.30	0.05	3.90	0.51	0.03	2.68	0.48	0.05	3.13
20	0.21	2	0.90	0.03	1.33	0.99	0.01	0.00	0.99	0.01	0.00	0.90	0.03	1.33	0.99	0.01	0.00	0.99	0.01	0.00	0.90	0.03	1.23	0.99	0.01	0.00	0.99	0.01	0.00
20	0.21	5	0.94	0.03	0.67	0.99	0.00	0.00	0.99	0.00	0.00	0.92	0.04	0.80	0.98	0.00	0.00	0.99	0.00	0.00	0.92	0.03	0.86	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	10	0.90	0.05	0.93	0.98	0.01	0.00	0.98	0.01	0.00	0.89	0.06	1.27	0.98	0.01	0.00	0.98	0.01	0.00	0.90	0.04	1.05	0.98	0.02	0.07	0.98	0.01	0.01
20	0.21	20	0.70	0.04	2.40	0.88	0.04	0.76	0.89	0.03	0.65	0.73	0.05	1.93	0.90	0.04	0.33	0.92	0.03	0.20	0.76	0.05	1.87	0.89	0.06	0.47	0.91	0.06	0.53
20	0.21	30	0.55	0.04	3.07	0.73	0.04	2.16	0.76	0.05	1.69	0.64	0.06	2.67	0.77	0.05	1.60	0.81	0.04	1.07	0.65	0.05	2.56	0.76	0.06	1.67	0.80	0.04	1.53
20	0.21	40	0.41	0.03	4.80	0.58	0.04	3.10	0.62	0.05	2.50	0.54	0.04	3.53	0.67	0.05	2.27	0.72	0.04	1.93	0.57	0.05	3.24	0.64	0.04	2.20	0.71	0.04	2.13

			c-means			RS c-means			HG c-means			GMM			RS-GMM			HG-GMM		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
3	-0.26	2	0.45	0.09	0.02	0.45	0.09	0.02	0.45	0.09	0.02	0.43	0.13	0.05	0.42	0.14	0.13	0.38	0.15	0.34
3	-0.26	5	0.34	0.12	0.10	0.34	0.12	0.10	0.34	0.12	0.10	0.53	0.16	0.06	0.51	0.18	0.17	0.46	0.18	0.33
3	-0.26	10	0.31	0.11	0.13	0.31	0.11	0.13	0.31	0.11	0.13	0.50	0.22	0.34	0.20	0.24	0.54	0.15	0.23	0.56
3	-0.26	20	0.25	0.12	0.11	0.25	0.12	0.13	0.25	0.12	0.13	0.28	0.20	0.55	0.07	0.17	0.73	0.03	0.10	0.72
3	-0.26	30	0.20	0.12	0.19	0.20	0.12	0.20	0.20	0.12	0.23	0.07	0.09	0.54	0.04	0.08	0.59	0.01	0.01	0.71
3	-0.26	40	0.18	0.11	0.22	0.18	0.11	0.19	0.18	0.11	0.17	0.03	0.04	0.70	0.03	0.06	0.71	0.02	0.02	0.68
3	-0.26	50	0.15	0.10	0.31	0.15	0.10	0.34	0.15	0.10	0.28	0.02	0.02	0.58	0.02	0.05	0.60	0.02	0.02	0.64
3	-0.26	60	0.14	0.09	0.25	0.15	0.10	0.28	0.16	0.10	0.27	0.01	0.01	0.55	0.01	0.01	0.58	0.01	0.01	0.52
3	-0.26	70	0.12	0.08	0.32	0.13	0.09	0.31	0.13	0.09	0.36	0.01	0.01	0.60	0.02	0.02	0.65	0.02	0.03	0.54
3	-0.10	2	0.69	0.12	0.02	0.69	0.12	0.02	0.69	0.12	0.02	0.75	0.10	0.01	0.74	0.09	0.02	0.74	0.10	0.02
3	-0.10	5	0.64	0.12	0.02	0.64	0.12	0.02	0.64	0.12	0.02	0.77	0.12	0.02	0.75	0.13	0.03	0.70	0.17	0.14
3	-0.10	10	0.60	0.15	0.03	0.60	0.14	0.03	0.60	0.14	0.03	0.74	0.21	0.16	0.37	0.32	0.35	0.35	0.32	0.39
3	-0.10	20	0.59	0.11	0.02	0.60	0.11	0.01	0.60	0.11	0.01	0.41	0.25	0.39	0.12	0.22	0.62	0.02	0.09	0.63
3	-0.10	30	0.56	0.13	0.04	0.56	0.14	0.05	0.56	0.14	0.05	0.16	0.16	0.61	0.09	0.13	0.64	0.02	0.01	0.69
3	-0.10	40	0.56	0.14	0.04	0.56	0.14	0.04	0.56	0.14	0.04	0.05	0.06	0.54	0.05	0.06	0.55	0.02	0.01	0.56
3	-0.10	50	0.52	0.16	0.07	0.52	0.15	0.06	0.52	0.15	0.06	0.03	0.03	0.50	0.03	0.02	0.54	0.03	0.02	0.63
3	-0.10	60	0.47	0.15	0.03	0.48	0.14	0.05	0.48	0.15	0.05	0.03	0.02	0.59	0.03	0.02	0.62	0.03	0.03	0.60
3	-0.10	70	0.47	0.15	0.06	0.48	0.14	0.06	0.47	0.15	0.10	0.03	0.02	0.61	0.03	0.03	0.67	0.03	0.03	0.47
3	0.01	2	0.81	0.12	0.01	0.81	0.12	0.01	0.81	0.12	0.01	0.90	0.04	0.00	0.90	0.04	0.00	0.90	0.04	0.00
3	0.01	5	0.80	0.10	0.00	0.80	0.10	0.00	0.80	0.10	0.00	0.91	0.05	0.01	0.91	0.07	0.02	0.90	0.09	0.03
3	0.01	10	0.79	0.11	0.01	0.79	0.12	0.02	0.79	0.12	0.02	0.84	0.18	0.13	0.56	0.29	0.29	0.52	0.32	0.23
3	0.01	20	0.79	0.12	0.01	0.79	0.12	0.01	0.79	0.12	0.01	0.52	0.25	0.41	0.18	0.24	0.54	0.05	0.13	0.71
3	0.01	30	0.72	0.18	0.06	0.73	0.18	0.06	0.73	0.18	0.06	0.22	0.20	0.54	0.12	0.18	0.61	0.03	0.07	0.70
3	0.01	40	0.74	0.14	0.02	0.74	0.16	0.04	0.74	0.16	0.04	0.09	0.10	0.58	0.07	0.08	0.58	0.03	0.05	0.67
3	0.01	50	0.71	0.18	0.05	0.71	0.19	0.06	0.71	0.19	0.06	0.05	0.05	0.65	0.04	0.05	0.66	0.04	0.05	0.51
3	0.01	60	0.74	0.14	0.02	0.74	0.15	0.03	0.74	0.15	0.03	0.05	0.05	0.59	0.04	0.03	0.61	0.05	0.03	0.57
3	0.01	70	0.70	0.17	0.04	0.70	0.19	0.06	0.70	0.19	0.07	0.05	0.04	0.48	0.05	0.05	0.60	0.05	0.05	0.43
3	0.21	2	0.97	0.05	0.00	0.97	0.05	0.00	0.97	0.05	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	5	0.97	0.06	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	10	0.97	0.04	0.00	0.97	0.04	0.00	0.97	0.04	0.00	0.99	0.06	0.00	0.90	0.18	0.04	0.87	0.20	0.10
3	0.21	20	0.96	0.09	0.01	0.96	0.09	0.01	0.96	0.09	0.01	0.70	0.27	0.36	0.38	0.30	0.46	0.19	0.26	0.43
3	0.21	30	0.95	0.12	0.02	0.95	0.12	0.01	0.95	0.12	0.01	0.41	0.22	0.51	0.23	0.23	0.58	0.03	0.07	0.70
3	0.21	40	0.95	0.14	0.02	0.95	0.15	0.03	0.95	0.15	0.03	0.23	0.18	0.58	0.15	0.18	0.49	0.05	0.07	0.58
3	0.21	50	0.97	0.04	0.00	0.97	0.03	0.00	0.97	0.03	0.00	0.15	0.13	0.61	0.10	0.11	0.70	0.06	0.03	0.65
3	0.21	60	0.97	0.05	0.00	0.97	0.05	0.00	0.97	0.05	0.00	0.13	0.11	0.71	0.08	0.06	0.78	0.08	0.04	0.61
3	0.21	70	0.97	0.06	0.00	0.97	0.06	0.00	0.97	0.06	0.00	0.12	0.10	0.59	0.09	0.06	0.69	0.10	0.05	0.58

			c-means			RS c-means			HG c-means			GMM			RS-GMM			HG-GMM		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
10	-0.26	2	0.34	0.06	2.33	0.37	0.05	2.61	0.37	0.05	2.63	0.44	0.06	1.46	0.45	0.06	1.40	0.44	0.06	1.59
10	-0.26	5	0.30	0.04	0.71	0.30	0.04	0.66	0.30	0.04	0.65	0.33	0.07	1.60	0.33	0.08	1.99	0.33	0.08	1.85
10	-0.26	10	0.22	0.05	0.78	0.22	0.05	0.79	0.22	0.05	0.77	0.22	0.07	2.13	0.20	0.06	3.22	0.23	0.07	2.69
10	-0.26	20	0.16	0.06	1.54	0.17	0.06	1.59	0.17	0.06	1.56	0.11	0.05	3.03	0.02	0.03	3.56	0.07	0.05	3.32
10	-0.26	30	0.13	0.06	2.00	0.14	0.06	2.04	0.14	0.06	2.06	0.07	0.03	3.01	0.01	0.02	3.44	0.02	0.03	3.40
10	-0.26	40	0.11	0.05	2.52	0.11	0.06	2.63	0.11	0.06	2.46	0.04	0.02	3.05	0.01	0.01	3.72	0.01	0.01	3.37
10	-0.10	2	0.47	0.08	1.88	0.53	0.08	2.10	0.52	0.08	1.96	0.67	0.07	0.74	0.72	0.07	0.29	0.72	0.07	0.24
10	-0.10	5	0.58	0.05	0.04	0.58	0.05	0.05	0.58	0.04	0.06	0.62	0.09	0.76	0.62	0.07	1.01	0.62	0.07	1.06
10	-0.10	10	0.49	0.10	0.20	0.49	0.11	0.27	0.49	0.11	0.24	0.52	0.10	1.46	0.38	0.08	2.82	0.46	0.08	2.22
10	-0.10	20	0.45	0.13	0.54	0.45	0.13	0.58	0.45	0.13	0.62	0.26	0.09	2.51	0.05	0.06	3.82	0.14	0.07	3.16
10	-0.10	30	0.42	0.11	0.73	0.43	0.12	0.72	0.43	0.11	0.72	0.13	0.05	2.90	0.01	0.02	3.41	0.01	0.03	3.34
10	-0.10	40	0.36	0.14	1.38	0.36	0.16	1.51	0.36	0.16	1.50	0.07	0.04	2.96	0.01	0.01	3.41	0.01	0.02	3.30
10	0.01	2	0.56	0.13	1.66	0.65	0.11	1.73	0.63	0.11	1.71	0.82	0.07	0.48	0.90	0.02	0.00	0.90	0.02	0.00
10	0.01	5	0.78	0.03	0.00	0.78	0.03	0.00	0.78	0.03	0.00	0.83	0.06	0.31	0.83	0.06	0.42	0.82	0.06	0.48
10	0.01	10	0.70	0.09	0.09	0.70	0.10	0.13	0.70	0.10	0.13	0.74	0.09	0.92	0.58	0.10	2.26	0.63	0.08	1.88
10	0.01	20	0.70	0.10	0.10	0.69	0.11	0.16	0.70	0.10	0.15	0.44	0.11	2.20	0.13	0.08	3.68	0.22	0.08	2.88
10	0.01	30	0.63	0.19	0.51	0.62	0.21	0.60	0.63	0.21	0.59	0.21	0.09	2.71	0.02	0.03	3.41	0.03	0.05	3.15
10	0.01	40	0.66	0.13	0.34	0.66	0.15	0.36	0.66	0.14	0.36	0.13	0.05	2.65	0.01	0.02	3.40	0.02	0.03	3.11
10	0.21	2	0.70	0.15	1.44	0.82	0.14	1.32	0.82	0.14	1.30	0.93	0.07	0.48	0.99	0.00	0.00	0.99	0.00	0.00
10	0.21	5	0.96	0.04	0.06	0.97	0.02	0.00	0.97	0.02	0.00	0.97	0.05	0.17	0.99	0.00	0.00	0.99	0.00	0.00
10	0.21	10	0.96	0.03	0.00	0.96	0.03	0.00	0.96	0.03	0.00	0.94	0.06	0.40	0.86	0.06	1.05	0.87	0.05	0.84
10	0.21	20	0.95	0.04	0.03	0.95	0.04	0.01	0.95	0.04	0.01	0.72	0.10	1.62	0.38	0.09	2.58	0.46	0.10	2.06
10	0.21	30	0.94	0.06	0.09	0.94	0.09	0.05	0.94	0.09	0.05	0.47	0.09	2.32	0.12	0.09	3.08	0.17	0.08	2.79
10	0.21	40	0.93	0.09	0.16	0.93	0.13	0.12	0.93	0.12	0.11	0.31	0.08	2.77	0.02	0.02	3.32	0.04	0.06	2.99

			c-means			RS c-means			HG c-means			GMM			RS-GMM			HG-GMM		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
20	-0.26	2	0.28	0.03	7.67	0.36	0.01	10.60	0.35	0.01	10.47	0.43	0.02	3.13	0.45	0.04	3.53	0.45	0.04	3.47
20	-0.26	5	0.26	0.03	3.13	0.27	0.02	2.93	0.27	0.02	3.00	0.29	0.04	4.80	0.30	0.05	4.33	0.30	0.04	3.73
20	-0.26	10	0.19	0.04	2.93	0.19	0.05	3.00	0.20	0.05	3.00	0.19	0.03	4.67	0.21	0.04	5.93	0.21	0.06	5.40
20	-0.26	20	0.11	0.05	3.98	0.12	0.04	3.20	0.13	0.04	3.20	0.08	0.02	6.13	0.06	0.03	6.80	0.08	0.02	6.13
20	-0.26	30	0.08	0.04	5.15	0.09	0.05	4.93	0.09	0.05	4.60	0.04	0.01	6.67	0.01	0.01	7.00	0.03	0.02	6.80
20	-0.26	40	0.06	0.04	5.76	0.09	0.04	5.20	0.09	0.04	4.87	0.03	0.01	6.93	0.01	0.01	7.13	0.02	0.01	7.00
20	-0.10	2	0.33	0.04	7.57	0.43	0.01	10.80	0.43	0.03	10.00	0.62	0.06	2.73	0.71	0.06	0.93	0.69	0.07	1.13
20	-0.10	5	0.47	0.09	1.64	0.53	0.05	1.00	0.53	0.05	0.93	0.56	0.05	2.13	0.61	0.05	2.13	0.61	0.06	1.53
20	-0.10	10	0.47	0.05	0.77	0.45	0.10	0.53	0.45	0.09	0.53	0.45	0.07	3.20	0.44	0.07	5.13	0.46	0.06	4.47
20	-0.10	20	0.35	0.12	1.61	0.39	0.09	0.60	0.39	0.09	0.53	0.22	0.04	4.80	0.12	0.05	6.27	0.19	0.05	5.80
20	-0.10	30	0.31	0.11	2.52	0.29	0.13	2.67	0.29	0.13	3.13	0.09	0.02	6.40	0.02	0.03	7.07	0.05	0.03	7.27
20	-0.10	40	0.26	0.12	3.56	0.31	0.08	2.07	0.32	0.08	1.93	0.07	0.02	6.53	0.01	0.01	7.00	0.03	0.02	6.87
20	0.01	2	0.35	0.04	7.54	0.48	0.02	10.80	0.49	0.02	10.27	0.76	0.05	1.87	0.90	0.01	0.00	0.89	0.02	0.00
20	0.01	5	0.66	0.11	1.22	0.72	0.06	0.33	0.71	0.08	0.33	0.76	0.03	1.73	0.84	0.04	0.67	0.84	0.03	0.53
20	0.01	10	0.70	0.04	0.28	0.70	0.04	0.33	0.70	0.04	0.33	0.66	0.04	2.67	0.60	0.05	4.40	0.68	0.04	3.07
20	0.01	20	0.62	0.10	0.42	0.60	0.17	0.80	0.60	0.17	0.80	0.37	0.07	4.60	0.20	0.05	6.20	0.30	0.07	5.80
20	0.01	30	0.51	0.22	2.02	0.56	0.17	1.53	0.56	0.17	1.27	0.20	0.05	5.53	0.06	0.04	7.27	0.14	0.04	6.13
20	0.01	40	0.52	0.16	1.75	0.53	0.21	1.53	0.53	0.21	1.80	0.12	0.02	5.27	0.01	0.01	7.20	0.06	0.03	6.27
20	0.21	2	0.40	0.05	7.84	0.56	0.03	10.07	0.57	0.03	10.07	0.85	0.05	2.13	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	5	0.86	0.09	1.35	0.96	0.02	0.00	0.96	0.02	0.00	0.91	0.03	1.27	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	10	0.93	0.04	0.55	0.96	0.02	0.00	0.96	0.02	0.00	0.89	0.04	1.40	0.89	0.05	1.53	0.92	0.03	1.27
20	0.21	20	0.91	0.06	0.47	0.94	0.02	0.00	0.94	0.02	0.00	0.68	0.07	3.67	0.44	0.08	4.53	0.60	0.08	3.27
20	0.21	30	0.89	0.10	0.64	0.93	0.03	0.07	0.93	0.03	0.07	0.47	0.06	5.13	0.19	0.04	6.80	0.30	0.06	5.27
20	0.21	40	0.86	0.13	0.98	0.94	0.02	0.00	0.94	0.02	0.00	0.32	0.04	5.53	0.06	0.05	6.67	0.16	0.06	5.80

			RGMM (Shrunk)			RS-RGMM (Shrunk)			HG-RGMM (Shrunk)			RGMM (OAS)			RS-RGMM (OAS)			HG-RGMM (OAS)			RGMM (LW)			RS-RGMM (LW)			HG-RGMM (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
3	-0.26	2	0.43	0.11	0.03	0.44	0.12	0.09	0.41	0.14	0.19	0.43	0.13	0.06	0.44	0.13	0.11	0.41	0.15	0.19	0.44	0.12	0.03	0.45	0.12	0.04	0.45	0.13	0.07
3	-0.26	5	0.55	0.13	0.02	0.55	0.13	0.02	0.55	0.14	0.05	0.54	0.15	0.02	0.52	0.16	0.09	0.52	0.16	0.08	0.54	0.15	0.02	0.54	0.15	0.03	0.53	0.16	0.06
3	-0.26	10	0.61	0.16	0.04	0.58	0.19	0.17	0.60	0.19	0.15	0.60	0.18	0.07	0.59	0.19	0.20	0.61	0.19	0.12	0.59	0.17	0.04	0.58	0.18	0.18	0.60	0.18	0.12
3	-0.26	20	0.56	0.21	0.08	0.58	0.20	0.15	0.68	0.17	0.03	0.55	0.19	0.16	0.56	0.19	0.17	0.65	0.18	0.06	0.57	0.18	0.17	0.57	0.18	0.17	0.63	0.17	0.13
3	-0.26	30	0.29	0.15	0.41	0.34	0.14	0.50	0.40	0.13	0.29	0.33	0.15	0.44	0.35	0.15	0.50	0.43	0.14	0.40	0.37	0.18	0.46	0.37	0.18	0.49	0.45	0.21	0.42
3	-0.26	40	0.13	0.13	0.47	0.16	0.16	0.46	0.25	0.16	0.43	0.19	0.15	0.64	0.23	0.17	0.66	0.34	0.14	0.47	0.25	0.16	0.73	0.26	0.17	0.72	0.28	0.18	0.64
3	-0.26	50	0.05	0.06	0.42	0.07	0.12	0.41	0.12	0.16	0.44	0.09	0.12	0.59	0.12	0.17	0.61	0.24	0.18	0.58	0.17	0.17	0.64	0.17	0.17	0.65	0.14	0.18	0.64
3	-0.26	60	0.03	0.02	0.45	0.03	0.04	0.45	0.06	0.10	0.54	0.03	0.03	0.54	0.05	0.10	0.57	0.20	0.21	0.37	0.11	0.15	0.56	0.11	0.15	0.57	0.09	0.15	0.65
3	-0.26	70	0.02	0.02	0.52	0.03	0.08	0.51	0.08	0.14	0.41	0.02	0.02	0.58	0.07	0.14	0.59	0.27	0.21	0.29	0.11	0.15	0.68	0.13	0.17	0.67	0.10	0.16	0.67
3	-0.10	2	0.74	0.09	0.01	0.75	0.07	0.00	0.75	0.07	0.00	0.75	0.09	0.00	0.75	0.07	0.00	0.75	0.08	0.00	0.75	0.08	0.00	0.75	0.07	0.00	0.76	0.07	0.00
3	-0.10	5	0.79	0.06	0.00	0.79	0.06	0.00	0.78	0.07	0.00	0.78	0.08	0.00	0.77	0.10	0.00	0.76	0.12	0.03	0.79	0.07	0.00	0.79	0.08	0.00	0.78	0.09	0.00
3	-0.10	10	0.83	0.08	0.03	0.83	0.09	0.02	0.83	0.09	0.01	0.83	0.09	0.03	0.82	0.10	0.03	0.83	0.10	0.02	0.83	0.09	0.03	0.83	0.09	0.04	0.84	0.07	0.01
3	-0.10	20	0.78	0.15	0.02	0.78	0.15	0.02	0.86	0.08	0.00	0.78	0.15	0.06	0.78	0.14	0.06	0.85	0.09	0.00	0.79	0.15	0.07	0.80	0.14	0.06	0.86	0.08	0.00
3	-0.10	30	0.46	0.16	0.19	0.48	0.15	0.22	0.64	0.16	0.07	0.49	0.15	0.32	0.51	0.14	0.33	0.66	0.16	0.10	0.55	0.16	0.23	0.56	0.15	0.23	0.70	0.15	0.11
3	-0.10	40	0.20	0.12	0.40	0.21	0.13	0.43	0.33	0.11	0.34	0.24	0.16	0.45	0.26	0.16	0.48	0.38	0.13	0.41	0.31	0.14	0.40	0.31	0.15	0.40	0.38	0.14	0.35
3	-0.10	50	0.11	0.08	0.33	0.12	0.11	0.33	0.20	0.15	0.27	0.14	0.14	0.44	0.17	0.18	0.42	0.28	0.19	0.35	0.23	0.16	0.45	0.23	0.16	0.45	0.26	0.18	0.48
3	-0.10	60	0.07	0.05	0.51	0.09	0.10	0.50	0.14	0.14	0.39	0.08	0.09	0.55	0.12	0.16	0.53	0.24	0.22	0.35	0.17	0.14	0.58	0.18	0.15	0.61	0.19	0.16	0.52
3	-0.10	70	0.05	0.04	0.51	0.07	0.07	0.53	0.15	0.14	0.38	0.05	0.05	0.58	0.11	0.15	0.61	0.28	0.22	0.32	0.16	0.16	0.56	0.18	0.18	0.58	0.20	0.18	0.52
3	0.01	2	0.89	0.04	0.00	0.89	0.04	0.00	0.89	0.04	0.00	0.90	0.04	0.00	0.90	0.04	0.00	0.90	0.04	0.00	0.89	0.04	0.00	0.90	0.04	0.00	0.89	0.04	0.00
3	0.01	5	0.91	0.04	0.00	0.92	0.03	0.00	0.91	0.03	0.00	0.91	0.04	0.00	0.91	0.04	0.00	0.92	0.04	0.00	0.92	0.03	0.00	0.92	0.03	0.00	0.92	0.03	0.00
3	0.01	10	0.93	0.04	0.00	0.92	0.06	0.01	0.92	0.06	0.01	0.93	0.04	0.00	0.92	0.06	0.01	0.92	0.06	0.01	0.93	0.04	0.00	0.93	0.04	0.00	0.93	0.05	0.00
3	0.01	20	0.92	0.10	0.02	0.92	0.09	0.01	0.95	0.04	0.00	0.92	0.09	0.02	0.92	0.09	0.02	0.94	0.05	0.00	0.91	0.11	0.03	0.91	0.11	0.03	0.95	0.05	0.00
3	0.01	30	0.64	0.17	0.18	0.65	0.16	0.21	0.81	0.14	0.05	0.68	0.19	0.25	0.68	0.18	0.25	0.81	0.16	0.14	0.65	0.17	0.25	0.66	0.17	0.23	0.82	0.14	0.07
3	0.01	40	0.36	0.13	0.22	0.37	0.14	0.26	0.49	0.13	0.16	0.40	0.14	0.42	0.42	0.13	0.43	0.50	0.14	0.34	0.40	0.13	0.44	0.42	0.12	0.46	0.52	0.13	0.31
3	0.01	50	0.20	0.11	0.44	0.23	0.14	0.43	0.33	0.13	0.19	0.22	0.15	0.58	0.27	0.17	0.56	0.36	0.15	0.32	0.30	0.14	0.45	0.30	0.14	0.47	0.34	0.14	0.36
3	0.01	60	0.14	0.09	0.43	0.15	0.11	0.43	0.24	0.15	0.22	0.12	0.10	0.45	0.17	0.16	0.53	0.26	0.21	0.35	0.21	0.14	0.41	0.22	0.15	0.43	0.26	0.16	0.34
3	0.01	70	0.11	0.08	0.44	0.14	0.12	0.41	0.26	0.16	0.26	0.10	0.08	0.48	0.17	0.17	0.50	0.32	0.20	0.30	0.20	0.14	0.48	0.23	0.17	0.51	0.27	0.17	0.38
3	0.21	2	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	5	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	1.00	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00
3	0.21	10	1.00	0.01	0.00	1.00	0.01	0.00	1.00	0.01	0.00	1.00	0.01	0.00	1.00	0.01	0.00	0.99	0.01	0.00	1.00	0.01	0.00	1.00	0.01	0.00	1.00	0.01	0.00
3	0.21	20	0.99	0.05	0.01	0.99	0.04	0.01	1.00	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.99	0.01	0.00	0.98	0.07	0.02	0.98	0.07	0.02	1.00	0.01	0.00
3	0.21	30	0.88	0.16	0.08	0.89	0.15	0.07	0.97	0.10	0.03	0.88	0.17	0.09	0.88	0.17	0.10	0.97	0.10	0.04	0.82	0.17	0.12	0.82	0.17	0.12	0.95	0.11	0.05
3	0.21	40	0.67	0.17	0.11	0.67	0.17	0.16	0.85	0.13	0.04	0.66	0.21	0.25	0.67	0.20	0.33	0.82	0.20	0.13	0.61	0.18	0.19	0.61	0.18	0.23	0.79	0.16	0.11
3	0.21	50	0.51	0.16	0.15	0.52	0.15	0.20	0.70	0.13	0.02	0.50	0.16	0.33	0.51	0.14	0.46	0.62	0.17	0.14	0.46	0.18	0.19	0.47	0.18	0.25	0.66	0.17	0.14
3	0.21	60	0.44	0.14	0.15	0.45	0.14	0.16	0.61	0.14	0.02	0.38	0.17	0.37	0.41	0.15	0.43	0.47	0.15	0.21	0.42	0.16	0.20	0.43	0.15	0.24	0.56	0.20	0.12
3	0.21	70	0.38	0.16	0.24	0.39	0.16	0.24	0.57	0.13	0.03	0.28	0.15	0.49	0.32	0.14	0.53	0.43	0.10	0.30	0.38	0.17	0.27	0.38	0.17	0.29	0.54	0.16	0.13

			RGMM (Shrunk)			RS-RGMM (Shrunk)			HG-RGMM (Shrunk)			RGMM (OAS)			RS-RGMM (OAS)			HG-RGMM (OAS)			RGMM (LW)			RS-RGMM (LW)			HG-RGMM (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
10	-0.26	2	0.44	0.06	1.36	0.49	0.06	1.03	0.48	0.06	1.13	0.45	0.06	1.35	0.47	0.06	1.29	0.46	0.06	1.33	0.44	0.06	1.38	0.46	0.06	1.32	0.46	0.06	1.46
10	-0.26	5	0.35	0.06	1.11	0.38	0.07	0.83	0.38	0.07	0.89	0.34	0.07	1.42	0.35	0.08	1.34	0.36	0.08	1.30	0.34	0.07	1.46	0.35	0.08	1.43	0.36	0.08	1.25
10	-0.26	10	0.29	0.06	1.48	0.37	0.08	0.87	0.37	0.08	0.65	0.27	0.07	1.77	0.33	0.09	1.48	0.35	0.09	1.24	0.26	0.07	1.77	0.34	0.09	1.62	0.34	0.09	1.41
10	-0.26	20	0.21	0.06	2.28	0.36	0.11	1.51	0.37	0.09	1.54	0.16	0.06	2.66	0.26	0.08	2.43	0.29	0.09	2.58	0.16	0.06	2.54	0.23	0.10	2.66	0.25	0.11	2.68
10	-0.26	30	0.11	0.04	2.93	0.15	0.05	3.05	0.23	0.06	2.49	0.09	0.04	2.96	0.14	0.05	3.06	0.15	0.06	2.97	0.10	0.04	3.11	0.07	0.05	3.30	0.09	0.07	3.32
10	-0.26	40	0.06	0.03	2.79	0.07	0.03	2.85	0.13	0.04	2.76	0.06	0.03	2.87	0.07	0.04	3.34	0.08	0.04	3.53	0.08	0.03	2.86	0.02	0.01	7.27	0.06	0.04	3.50
10	-0.10	2	0.69	0.07	0.62	0.75	0.05	0.12	0.75	0.05	0.10	0.67	0.07	0.72	0.73	0.06	0.21	0.73	0.07	0.22	0.67	0.07	0.72	0.72	0.07	0.22	0.72	0.07	0.24
10	-0.10	5	0.68	0.06	0.18	0.70	0.05	0.08	0.70	0.05	0.13	0.63	0.08	0.65	0.67	0.07	0.35	0.67	0.08	0.42	0.63	0.09	0.64	0.67	0.07	0.40	0.67	0.07	0.38
10	-0.10	10	0.62	0.08	0.63	0.70	0.06	0.13	0.70	0.07	0.13	0.57	0.09	1.00	0.64	0.09	0.80	0.66	0.09	0.52	0.57	0.09	0.97	0.64	0.09	0.85	0.65	0.09	0.74
10	-0.10	20	0.46	0.09	1.75	0.66	0.09	0.63	0.68	0.09	0.49	0.38	0.09	2.07	0.48	0.10	1.89	0.56	0.10	1.55	0.39	0.09	2.10	0.48	0.10	1.70	0.56	0.11	1.54
10	-0.10	30	0.25	0.06	2.44	0.40	0.07	2.11	0.45	0.07	1.73	0.20	0.06	2.67	0.26	0.07	2.73	0.31	0.07	2.52	0.22	0.06	2.54	0.25	0.10	2.58	0.33	0.09	2.65
10	-0.10	40	0.13	0.04	2.73	0.20	0.05	2.80	0.27	0.06	2.44	0.11	0.04	2.83	0.11	0.06	3.47	0.15	0.07	3.14	0.14	0.05	2.84	0.09	0.07	3.10	0.17	0.09	3.07
10	0.01	2	0.84	0.07	0.43	0.90	0.02	0.01	0.90	0.02	0.01	0.83	0.07	0.47	0.91	0.02	0.00	0.91	0.02	0.00	0.83	0.07	0.47	0.90	0.02	0.00	0.90	0.02	0.00
10	0.01	5	0.86	0.04	0.12	0.88	0.02	0.00	0.88	0.02	0.00	0.84	0.06	0.25	0.87	0.02	0.00	0.87	0.02	0.00	0.84	0.06	0.23	0.87	0.03	0.00	0.87	0.03	0.00
10	0.01	10	0.84	0.06	0.25	0.88	0.04	0.02	0.88	0.03	0.01	0.79	0.08	0.53	0.86	0.06	0.18	0.87	0.05	0.10	0.79	0.08	0.60	0.85	0.07	0.28	0.85	0.06	0.24
10	0.01	20	0.68	0.09	1.22	0.87	0.07	0.18	0.89	0.05	0.04	0.58	0.08	1.62	0.67	0.09	1.20	0.78	0.09	0.82	0.58	0.09	1.61	0.68	0.08	1.03	0.78	0.08	0.78
10	0.01	30	0.41	0.08	2.06	0.57	0.09	1.53	0.66	0.10	1.23	0.31	0.09	2.52	0.37	0.09	2.47	0.47	0.10	2.01	0.34	0.09	2.28	0.38	0.10	2.35	0.51	0.11	2.02
10	0.01	40	0.24	0.05	2.20	0.32	0.08	2.19	0.44	0.06	1.88	0.19	0.06	2.50	0.19	0.08	3.16	0.25	0.07	2.71	0.22	0.05	2.36	0.20	0.09	2.74	0.31	0.10	2.54
10	0.21	2	0.94	0.06	0.46	0.99	0.00	0.00	0.99	0.00	0.00	0.94	0.07	0.44	0.99	0.00	0.00	0.99	0.00	0.00	0.93	0.07	0.48	0.99	0.00	0.00	0.99	0.00	0.00
10	0.21	5	0.97	0.05	0.17	0.99	0.00	0.00	0.99	0.00	0.00	0.97	0.05	0.17	0.99	0.00	0.00	0.99	0.00	0.00	0.97	0.05	0.17	0.99	0.00	0.00	0.99	0.00	0.00
10	0.21	10	0.97	0.05	0.19	0.99	0.00	0.00	0.99	0.00	0.00	0.94	0.06	0.33	0.99	0.00	0.00	0.99	0.00	0.00	0.94	0.06	0.33	0.99	0.00	0.00	0.99	0.00	0.00
10	0.21	20	0.89	0.07	0.65	0.99	0.02	0.04	1.00	0.00	0.00	0.82	0.07	0.95	0.89	0.07	0.34	0.99	0.02	0.02	0.82	0.07	0.98	0.89	0.06	0.47	0.98	0.03	0.04
10	0.21	30	0.72	0.08	1.38	0.88	0.06	0.53	0.97	0.03	0.04	0.61	0.09	1.73	0.68	0.08	1.06	0.85	0.07	0.65	0.59	0.08	1.86	0.65	0.09	1.16	0.82	0.08	0.78
10	0.21	40	0.51	0.08	1.88	0.68	0.08	1.28	0.77	0.08	0.89	0.43	0.08	2.25	0.42	0.08	2.13	0.56	0.09	1.69	0.42	0.08	2.20	0.38	0.10	2.09	0.52	0.10	1.78

			RGMM (Shrunk)			RS-RGMM (Shrunk)			HG-RGMM (Shrunk)			RGMM (OAS)			RS-RGMM (OAS)			HG-RGMM (OAS)			RGMM (LW)			RS-RGMM (LW)			HG-RGMM (LW)		
k	c	d	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)	ARI (avg)	ARI (std)	CI (avg)
20	-0.26	2	0.44	0.03	3.13	0.49	0.02	1.80	0.47	0.03	2.73	0.43	0.02	3.33	0.45	0.04	2.87	0.44	0.03	3.47	0.43	0.02	3.13	0.45	0.03	3.00	0.45	0.03	3.47
20	-0.26	5	0.31	0.03	3.27	0.35	0.03	2.27	0.35	0.04	1.87	0.30	0.04	3.60	0.31	0.05	3.13	0.31	0.05	2.93	0.30	0.04	3.93	0.31	0.05	3.67	0.31	0.05	3.67
20	-0.26	10	0.24	0.03	3.73	0.30	0.05	2.73	0.31	0.05	2.80	0.22	0.04	4.47	0.26	0.06	4.53	0.27	0.07	4.27	0.22	0.05	4.53	0.26	0.06	4.73	0.27	0.05	4.20
20	-0.26	20	0.11	0.03	5.20	0.18	0.04	4.67	0.20	0.04	4.40	0.08	0.02	5.67	0.13	0.03	6.27	0.14	0.04	5.40	0.08	0.02	6.27	0.12	0.03	6.13	0.13	0.03	5.60
20	-0.26	30	0.06	0.02	5.60	0.08	0.02	6.38	0.13	0.03	5.07	0.05	0.01	5.93	0.07	0.02	6.80	0.07	0.01	6.73	0.05	0.01	6.40	0.02	0.01	6.73	0.06	0.04	5.87
20	-0.26	40	0.04	0.01	6.53	0.06	0.02	6.20	0.07	0.02	6.13	0.03	0.01	6.93	0.04	0.02	6.80	0.04	0.02	6.93	0.04	0.02	7.07	0.02	0.01	7.27	0.03	0.02	6.80
20	-0.10	2	0.66	0.05	1.93	0.76	0.04	0.20	0.74	0.04	0.27	0.63	0.05	2.33	0.74	0.04	0.47	0.70	0.06	1.07	0.62	0.06	2.73	0.73	0.06	0.47	0.70	0.06	0.93
20	-0.10	5	0.61	0.04	1.20	0.69	0.04	0.20	0.68	0.04	0.20	0.57	0.05	2.20	0.62	0.06	1.40	0.62	0.06	1.27	0.57	0.05	2.20	0.61	0.07	1.53	0.62	0.06	1.20
20	-0.10	10	0.55	0.06	2.13	0.66	0.05	0.40	0.66	0.05	0.33	0.48	0.07	2.93	0.57	0.07	2.27	0.58	0.07	1.73	0.48	0.05	2.73	0.56	0.08	2.47	0.59	0.06	1.67
20	-0.10	20	0.34	0.04	4.00	0.45	0.06	3.53	0.52	0.07	2.60	0.26	0.05	4.93	0.31	0.06	4.93	0.36	0.05	4.67	0.27	0.05	4.53	0.33	0.04	5.00	0.39	0.06	4.53
20	-0.10	30	0.16	0.03	5.87	0.24	0.03	5.60	0.29	0.04	5.07	0.11	0.02	5.93	0.15	0.03	6.67	0.16	0.04	6.47	0.13	0.03	6.27	0.16	0.04	6.87	0.20	0.04	6.20
20	-0.10	40	0.10	0.02	6.00	0.13	0.03	6.40	0.18	0.03	5.07	0.07	0.02	6.53	0.07	0.02	6.53	0.10	0.03	6.20	0.09	0.03	6.00	0.06	0.05	6.27	0.10	0.05	6.40
20	0.01	2	0.79	0.05	1.60	0.90	0.01	0.00	0.90	0.01	0.00	0.76	0.06	1.87	0.90	0.02	0.00	0.89	0.04	0.13	0.76	0.05	1.93	0.90	0.01	0.00	0.89	0.03	0.07
20	0.01	5	0.80	0.03	1.20	0.87	0.02	0.00	0.87	0.02	0.00	0.77	0.03	1.60	0.87	0.02	0.00	0.87	0.03	0.07	0.76	0.03	1.67	0.86	0.04	0.13	0.86	0.03	0.07
20	0.01	10	0.76	0.04	1.47	0.86	0.02	0.07	0.86	0.02	0.00	0.69	0.04	2.13	0.80	0.03	1.07	0.82	0.05	0.60	0.68	0.05	2.33	0.79	0.03	1.33	0.80	0.03	1.20
20	0.01	20	0.57	0.08	3.33	0.72	0.07	1.80	0.78	0.08	1.00	0.43	0.09	4.67	0.50	0.08	4.00	0.59	0.10	3.40	0.46	0.10	3.33	0.52	0.08	3.67	0.61	0.09	3.00
20	0.01	30	0.33	0.05	4.47	0.45	0.04	4.60	0.54	0.05	3.33	0.23	0.05	5.33	0.26	0.03	6.13	0.35	0.05	4.73	0.25	0.06	5.27	0.29	0.05	5.47	0.36	0.05	5.13
20	0.01	40	0.20	0.05	5.00	0.24	0.04	5.40	0.33	0.04	4.20	0.15	0.04	5.47	0.12	0.02	6.47	0.19	0.04	5.60	0.16	0.04	5.27	0.15	0.06	6.67	0.24	0.04	5.73
20	0.21	2	0.86	0.04	2.13	0.99	0.00	0.00	0.99	0.00	0.00	0.84	0.06	2.20	0.99	0.00	0.00	0.99	0.00	0.00	0.84	0.06	2.20	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	5	0.92	0.04	1.20	0.99	0.00	0.00	0.99	0.00	0.00	0.91	0.03	1.27	0.99	0.00	0.00	0.99	0.00	0.00	0.91	0.03	1.27	0.99	0.00	0.00	0.99	0.00	0.00
20	0.21	10	0.94	0.05	0.93	0.99	0.00	0.00	0.99	0.00	0.00	0.90	0.03	1.33	0.98	0.02	0.27	0.99	0.00	0.00	0.90	0.03	1.27	0.99	0.01	0.07	0.99	0.00	0.00
20	0.21	20	0.85	0.06	1.80	0.96	0.04	0.40	0.99	0.00	0.00	0.73	0.05	3.07	0.82	0.06	1.73	0.93	0.04	0.87	0.72	0.06	3.07	0.81	0.04	2.13	0.94	0.05	0.73
20	0.21	30	0.66	0.05	3.07	0.80	0.04	1.87	0.91	0.03	0.80	0.54	0.06	4.00	0.53	0.07	4.47	0.68	0.07	2.87	0.54	0.05	3.93	0.50	0.06	4.33	0.66	0.05	3.20
20	0.21	40	0.50	0.04	3.73	0.56	0.05	3.47	0.69	0.03	2.33	0.35	0.03	5.33	0.32	0.03	5.40	0.42	0.06	4.13	0.37	0.03	5.00	0.30	0.04	4.73	0.42	0.05	4.07