THÈSE de DOCTORAT

Opérée au sein de :

l'Université de Lille

École doctorale : MADIS-631

Spécialité de doctorat : Mathématiques Appliquées

Thèse préparée et soutenue publiquement le 30/09/2022, pour obtenir le grade de Docteur en Statistiques par :

Filippo Antonazzo

Unsupervised learning of huge data sets with limited computed resources

Apprentissage non supervisé pour données extrêmement volumineuses en situation de ressources informatiques arbitrairement limitées

---

Devant le jury composé de :

Président du jury	Mustapha Lebbah	Maître de conférences Université Sorbonne Paris Nord
Rapportrice	Cinzia Viroli	Full professor Università di Bologna
Rapporteur	Allou Samé	Directeur de recherche Université Gustave Eiffel
Examinatrice	Cathy Maugis-Rabusseau	Maître de conférences INSA Toulouse
Directeur de thèse	Christophe Biernacki	Professeur Université de Lille
Co-Directrice de thèse	Christine Keribin	Maître de conférences Université Paris-Saclay

Abstract

Clustering reveals all its interest when the data set size considerably increases, since there is the opportunity to discover tiny but possibly high value clusters, which can not be detected with moderate sample sizes. However, the clustering of such high data volumes encounters computational limitations, requiring extremely high memory and computational resources. Thus, current clustering algorithms need frugal implementations, also demanded by institutions and industries to accomplish today's eco-friendly policies. In this context, Gaussian model-based clustering, a popular clustering technique based on Gaussian mixtures, has required frugal adaptations to overcome these computational limitations and to report, even in the huge data case, the same good performance achieved in moderate size analyses. Such implementations are essentially based on subsampling strategies, which manage to be frugal, but they are expected to heavily failed in highly imbalanced cluster case. Thus, in this work, we propose a frugal technique, based on a so-called bin-marginal data-compression, to perform Gaussian model-based clustering on huge and imbalanced data sets. After a preliminary analysis on simple univariate settings revealing the potential of our solution (here, based on univariate binned data), we extend our proposal to multivariate data sets, where bin-marginal data are employed to perform a drastic reduction of the data volume. Despite this extreme loss of information, we prove identifiability property for the diagonal mixture model and we also introduce a specific EM-like algorithm associated to a composite likelihood approach guaranteeing frugality. Numerical experiments highlight that the proposed method outperforms subsampling both in controlled simulations and in various real applications where imbalanced clusters may typically appear, such as image segmentation, hazardous asteroids recognition and fraud detection. Then, additional topics regarding model choice, the problem of local maxima and the impact of our data-compression on clustering are dealt with a pure experimental point of view. Finally, through a collaboration with a company specialized in predictive maintanance, a practical application of anomaly detection on real time series is shown, in order to extend the potential application domains of the proposal.

Résumé

Par nature, le clustering révèle tout son intérêt lorsque le volume des jeux de données augmente considérablement, parce qu'il y a ainsi l'opportunité de découvrir des classes potentiellement petites mais inconnues jusqu'alors puisque indétectables avec des tailles d'échantillons plus réduits. L'intérêt de telles classes peut être en outre inversement proportionnel à leur taille, signe de phénomènes atypiques mais à forte valeur comme des anomalies, des fraudes, etc. Toutefois, classifier de tels volumes de données peut facilement rencontrer des limitations informatiques fortes, demandant en effet potentiellement d'énormes quantités de mémoire vive et d'autres ressources informatiques substantielles (calcul, énergie, flux). Par conséquent, si l'on souhaite effectivement mettre en oeuvre des algorithmes de classification sur de très grands jeux de données tout en limitant les ressources informatiques à mobiliser (pour des raisons de coût ou d'écologie), il est nécessaire d'envisager des approches beaucoup plus frugales que les approches actuelles, tout en garantissant des résultats d'estimation de haute qualité. La classification sur modèle de mélange gaussien étant certainement l'approche la plus populaire (ne serait-ce par son lien structurel avec les méthodes de k-means), ce travail de thèse explore prioritairement la frugalité du clustering dans ce cadre. Il est à noter que des stratégies fondées sur de l'échantillonnage, bien qu'ayant de bonnes propriétés de frugalité, doivent être écartées car elles s'avèrent incapables de détecter des partitions extrêmement déséquilibrées, ce qui est un prérequis essentiel dans notre contexte. Par conséquent, dans cette thèse, on adopte une stratégie frugale alternative qui repose sur une compression des données à la fois par axes et par intervalles (on parle alors de "bin-marginal"). Après une analyse préliminaire en situation simplifiée (univarié avec bins) qui révèle le potentiel de notre proposition, nous abordons le cas multivarié (combinant cette fois bins et marginalisation) qui sera le coeur de ce travail. Malgré la réduction extrême des données permise par le "bin-marginal", nous montrons que cette perte drastique d'information n'est pas préjudiciable à l'objectif de clustering par mélanges gaussiens dans le cas diagonal. Dans un premier temps, nous montrons l'identifiabilité de ces mélanges diagonaux et nous introduisons un algorithme spécifique similaire à EM mais associé à une approche basée sur une vraisemblance composite qui s'appuie sur une garantie de consistance des estimateurs. Des expériences numériques illustrent que notre méthode est beaucoup plus performante que le sous-échantillonnage soit dans des simulations, soit dans des applications réelles où les classes sont fortement déséquilibrées par nature, comme la segmentation d'images, la reconnaissance d'astéroïdes dangereux ou la détection de fraudes. Ensuite, des sujets supplémentaires concernant le choix de modèle, la problématique des maxima

locaux et l'impact de notre compression sur le clustering sont traités avec un point de vue plus expérimental. Finalement, une application pratique de détection d'anomalies sur des séries temporelles (potentiellement très volumineuse), et réalisée dans le cadre d'un partenariat avec une petite entreprise spécialisée en maintenance prédictive, est menée pour évaluer la potentialité de notre approche dans un domaine d'application connexe.

Contents

Introduction

Clustering and all other analyses known as unsupervised learning aim to distinguish homogeneous classes among data, when data labels are not provided. Nowadays, these techniques are intensively studied, as, in many contexts, it is impossible to do a data labeling, especially in those cases where data size is prohibitive (millions or billions of records). Actually, huge data sets are really common (Sagiroglu and Sinanc, 2013), thanks to the technological development of the last decades. In addition to the impossibility of data labeling, the analysis of such enormous statistical information is complex with traditional methods, as classical clustering algorithms require too many computational resources (time, memory and energy). This is also in contrast with the current eco-friendly policies of many national governments and industries, which are searching for methods able to do good statistical analyses without employing complex and wasteful technologies. Thus, in this thesis, we focus on methods able to perform clustering on huge data sets *frugally*, i.e., exploiting only the limited computational resources of a standard laptop.

The aim of this thesis is to propose a frugal approach for *model-based clustering*. This is a way of clustering which has become popular because it allows a well-posed mathematical definition of the clusters, thanks to the use of Gaussian mixture models, for instance. Model-based clustering has proved to be successful in case of moderate size data sets, but its common frugal specifications can be inefficient, especially if the data set to analyze is *imbalanced*. These kinds of data, where there is one class composed by very few elements in comparison to the others, can be collected in different fields, such as anomaly or fraud detection. In general, imbalanced data appear in all contexts where very few "abnormal" objects have to be recognised among a large amount of "normal" ones. Thus, in this thesis, we provide a frugal method for model-based clustering which can be applied even on such huge imbalanced data sets.

This work is organised as follows. In Chapter 1, we define our framework, reviewing current clustering algorithms with particular emphasis on ideas to make clustering frugal and on model-based clustering. We also provide the mathematical tools that are essential for the rest of the work. Then, we specify the motivations of the thesis and its main contributions. It is in this part that we introduce binned data, whose use is crucial for our frugality purposes. Indeed, the *artificial* construction of binned data let us to obtain a heavily-reduced data set which can be clustered frugally. In Chapter 2, we apply this idea in two simple practical situations. Firstly, we estimate a single univariate Gaussian, investigating both the theoretical properties of the estimators in presence of binned data and the influence of the binning grids on the estimation. Then, we pass to the

mixture case (still univariate), illustrating a suitable algorithm of estimation which lets us appreciate for the very first time the benefits of our proposal in terms of efficiency and frugality. In Chapter 3, we present our main contribution, that is a technique to cluster multidimensional huge data sets (mainly imbalanced). Here, we discuss why a simple multivariate extension of the binned solution provided in Chapter 2 is not possible and how we can cope with it using our principal contribution. We also discuss some theoretical aspects of the models involved, including identifiability. Finally, extensive numerical simulations and real applications are provided in order to quantify the frugality of the method and its advantages with relation to concurrency, especially in presence of cluster imbalance. In Chapter 4, further topics regarding local maxima, model and grid choice are dealt with, essentially from a numerical point for view. In Chapter 5, we complete the work showing a possible application of our proposal on real cases of anomaly detection in time series. In the final chapter, we briefly summarize the main results contained in the thesis and its perspectives for future research.

List of contributions The content of this thesis is based on the following contributions listed below:

- Chapter 2 is based on the work "Estimation of univariate Gaussian mixtures for huge raw data sets by using binned data sets", F. Antonazzo, C. Biernacki and C. Keribin, submitted to JDS 2020-52ème Journées de Statistiques de la Société Française de Statistique.

- Chapter 3 is based on "Frugal Gaussian clustering of huge imbalanced data sets through a bin-marginal approach", F. Antonazzo, C. Biernacki and C. Keribin, submitted to Statistics and Computing (under revision).

Chapter 1

Background and motivations

In this chapter, we illustrate the state of the art in clustering for very large data sets. In particular, we discuss the main ideas to make clustering frugal, with particular emphasis on our framework of reference, the model-based clustering. In the same chapter, we introduce binned data, giving both definition and formal notation, on which we base the techniques proposed in the next chapters. We conclude with a brief presentation of the contributions contained in the thesis, explaining why they are introduced.

1.1 Clustering

Clustering is a very common statistical technique consisting in dividing a data set $\mathcal{D}$ into a partition $\{\mathcal{C}_k \subset \mathcal{D}, k = 1, \ldots, K\}$, where $\mathcal{C}_k, k = 1, \ldots, K$ are K groups that are homogenous according to a certain criterion of similarity between the elements. Different approaches for clustering can be identified: the most common are partitioning, hierarchical, density-based, model-based. Despite a wider description of each clustering methods could be interesting, this is not the aim of the chapter. We prefer giving in the next paragraphs an illustration of the best-known algorithms for each approach in order to provide their distinctive notation. This is useful to describe the frugal methods of the following sections. Furthermore, we reserve for model-based clustering a stand-alone section, because this thesis is mainly based on this approach.

1.1.1 Partitioning clustering

In this kind of clustering we divide the initial data set into a partition of K subgroups that minimizes a certain cost function. This partitioning is usually conducted iteratively until its stabilization and the number K is a-priori fixed. In this section, we describe two famous basic algorithms: K-means (MacQueen et al., 1967) and K-medoids (Kaufman and Rousseeuw, 1990).

K-means K-means is recognised as one of the oldest clustering algorithms. Its main idea is quite simple: it associates each point to the cluster with the nearest mean point

(or *center*) respectively to the Euclidean distance $d_2(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||_2$, where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$. Denoting the n data points as $\boldsymbol{x}_i \in \mathbb{R}^D$, $i = 1, \ldots, n$, and with $\mathcal{C}_k$, $k = 1, \ldots, K$ the K clusters, K-means consists in these steps:

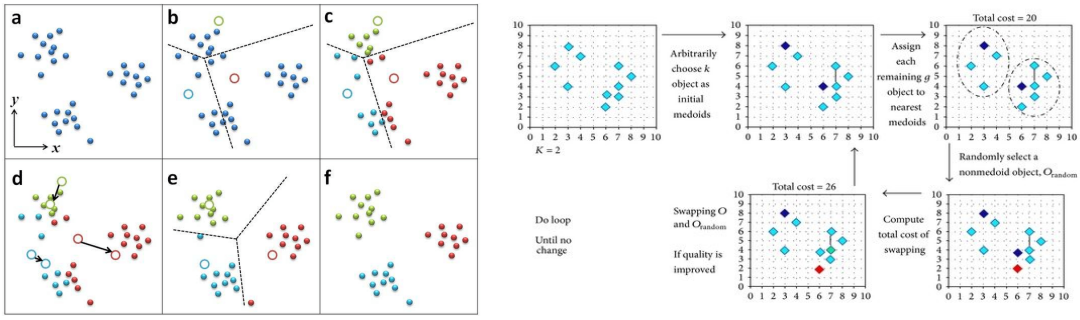
1. **Initialization**: fix a number K of clusters and select randomly K points $\boldsymbol{c}_k \in \mathbb{R}^D$, $k = 1, \ldots, K$. These points are considered the means of the initial clusters.
2. **Clusters formation**: associate each point to the cluster with the nearest mean according to the distance $d_2(\cdot, \cdot)$.
3. **Means update**: update $\boldsymbol{c}_k$, $k = 1, \ldots, K$ with the means of the objects constituting the new clusters:
$$\boldsymbol{c}_k = \frac{\sum_{i:\boldsymbol{x}_i \in \mathcal{C}_k} \boldsymbol{x}_i}{|\mathcal{C}_k|},$$
where $|\mathcal{C}_k|$ is the cardinality of $\mathcal{C}_k$.
4. Repeat 2-3 until no point changes cluster.

In Figure 1.1a an example of K-means clustering is depicted: the first two pictures (a-b) refer to the initialization phase; the third picture (c) shows the initial clusters formation; in the fourth picture (d) means are updated leading to the new clusters formation of picture (e). The last picture (f) finally shows the recovered partition.

It is evident that K-means can not be used when categorical variables occur, as in defining an Euclidean distance between categorical data points is not appropriate. It is for this reason that K-medoids algorithm was introduced.

K-medoids A *medoid* of a cluster is a representative object of this cluster for which the total distance or dissimilarity to all the objects of the cluster is minimal. Medoids are similar in concept to means, but medoids are always restricted to be members of the data set (Boehmke and Greenwell, 2019). A method of partitioning clustering based on the search of medoids for K clusters is the K-medoids. A common implementation of K-medoids is the PAM (*Partitioning around medoids*) algorithm (Kaufman and Rousseeuw, 1990), which we describe in the following. It starts selecting K initial medoids among the data points and choosing a distance or dissimilarity function $d(\cdot, \cdot)$. It is possible to choose any kind of function, so suitable distances (dissimiliarities) to analyze categorical variables can be selected. Each medoid is denoted as $\boldsymbol{m}_k$, $k = 1, \ldots, K$, and their set as $\mathcal{M}$. Given $\mathcal{M}$, each data point $\boldsymbol{x}_i \notin \mathcal{M}$ is assigned to the cluster $\mathcal{C}_k$ whose medoid minimizes the distance function, in the same fashion of K-means. Given a set of medoids $\mathcal{M}$ and a partition $\mathcal{C}_1, \ldots, \mathcal{C}_K$, at each step a data point $\boldsymbol{x}_i \notin \mathcal{M}$ is chosen to substitute $\boldsymbol{m}_k$ as new possible medoid. Considering $\boldsymbol{x}_i$ as a medoid induces a new set $\mathcal{M}'$ and a new partition $\mathcal{C}'_1, \ldots, \mathcal{C}'_K$. The idea is to substitute $\boldsymbol{m}_k$ with $\boldsymbol{x}_i$ if the following cost function

$$Cost(\boldsymbol{x}_i, \boldsymbol{m}_k) = \sum_{l:\boldsymbol{x}_l \notin \mathcal{M}'} \sum_{k':\boldsymbol{x}_l \in \mathcal{C}'_{k'}} d(\boldsymbol{x}_l, \boldsymbol{m}'_{k'}) - \sum_{l:\boldsymbol{x}_l \notin \mathcal{M}} \sum_{k':\boldsymbol{x}_l \in \mathcal{C}_{k'}} d(\boldsymbol{x}_l, \boldsymbol{m}_{k'})$$



(a) (b)

Figure 1.1: Flowcharts of K-means (Chen and Lai, 2016) (a) and K-medoids (Choi and Kwon, 2015) (b) applied on two bi-dimensional data sets.

is negative. We are ready to formalize the steps of PAM:

1. **Initialization**: Select K initial medoids among the data points.
2. **Cluster formation**: associate each point to the cluster with the nearest medoid according to the distance $d(\cdot, \cdot)$.
3. **Optimal new medoid search**: for each medoid $\boldsymbol{m}_k$ and non-medoid point $\boldsymbol{x}_i$ calculate $Cost(\boldsymbol{x}_i, \boldsymbol{m}_k)$ and retain the couple $(\boldsymbol{x}_{i^*}, \boldsymbol{m}_{k^*})$ with the minimum cost.
4. **Swap**: if $Cost(\boldsymbol{x}_{i^*}, \boldsymbol{m}_{k^*}) < 0$ swap $\boldsymbol{m}_{k^*}$ with $\boldsymbol{x}_{i^*}$ and return to 2. Stop otherwise.

Figure 1.1b shows of a generic iteration of K-medoids: in this picture, all its characteristic phases are depicted starting from the first two pictures on the top-left (initialization) and finishing with the phase of swap represented in the central picture on the bottom.

Advantages and disadvantages Partitioning clustering algorithms have the advantage of being very intuitive and very easy to implement (Tomar and Agarwal, 2013). However, they encounter difficulties in detecting non-convex clusters and they are very sensitive to outliers and initialization. In addition, the choice of the number of clusters K has to be a-priori defined (Xu and Tian, 2015).

1.1.2 Hierarchical clustering

Hierarchical clustering (Murtagh and Contreras, 2012) orders data points in a tree structure, named *dendogram*. In this context it is important, once selected a distance function $d(\cdot, \cdot)$ between two points, to define a distance between two clusters $\mathcal{C}_k$ and $\mathcal{C}_{k'}$, with cardinalities $|\mathcal{C}_k|$ and $|\mathcal{C}_{k'}|$, respectively. Examples of cluster distances are:

- **Single link:** $d_S(\mathcal{C}_k, \mathcal{C}_{k'}) = \min_{\boldsymbol{x}_i \in \mathcal{C}_k, \boldsymbol{x}_l \in \mathcal{C}_{k'}} d(\boldsymbol{x}_i, \boldsymbol{x}_l)$
- **Complete link:** $d_C(\mathcal{C}_k, \mathcal{C}_{k'}) = \max_{\boldsymbol{x}_i \in \mathcal{C}_k, \boldsymbol{x}_l \in \mathcal{C}_{k'}} d(\boldsymbol{x}_i, \boldsymbol{x}_l)$
- **Average link:** $d_A(\mathcal{C}_k, \mathcal{C}_{k'}) = \frac{\sum_{\boldsymbol{x}_i \in \mathcal{C}_k, \boldsymbol{x}_l \in \mathcal{C}_{k'}} d(\boldsymbol{x}_i, \boldsymbol{x}_l)}{|\mathcal{C}_k||\mathcal{C}_{k'}|}$

According to the way of constructing clusters we distinguish between *agglomerative* and *divisive* methods.

Agglomerative clustering Every algorithm starts considering each data point as a single cluster, thus $K = n$. At each step the two points (or two clusters in successive iterations) minimizing the chosen distance criterion are merged together. This process continues until all the points are grouped into the same cluster (K=1). In this way, a tree structure (the dendrogram) is obtained, because each cluster could be considered as the parent node of the two clusters by which it is composed. The criterion of choice of the best clustering is the following: it is recommended to select the partition corresponding to the merging stage where the decrease in total cost function (sum of distance function within of points) becomes negligible respectively to a certain threshold. A particular case of agglomerative clustering is represented by the Ward's method (Ward, 1963). In this technique clusters are merged in order to minimize the increase of the *intra-class inertia*:

$$I_a = \frac{1}{n} \sum_{k=1}^{K} \sum_{i:\boldsymbol{x_i} \in \boldsymbol{\mathcal{C}_k}} d(\boldsymbol{x}_i, \boldsymbol{c}_k),$$

where each $\boldsymbol{c}_k$ is the mean point of the cluster $\mathcal{C}_k$. This is also equivalent to maximize the increase of the *inter-class inertia*:

$$I_e = \frac{1}{n} \sum_{k=1}^{K} d(\boldsymbol{c}, \boldsymbol{c}_k),$$

where $\boldsymbol{c}$ is the mean point of all data.

Divisive clustering This approach is the exact opposite of the agglomerative one: it starts with a single cluster grouping all observations and at each step the cluster is divided into two new clusters aiming to minimize a cost function based on a cluster distance. Figure 1.2 synthesizes both agglomerative and divisive approach, representing the dendrogram obtained for a clustering of seven objects. Despite it seems as intuitive as the agglomerative approach, divisive clustering has not encountered the same success in common statistical applications.

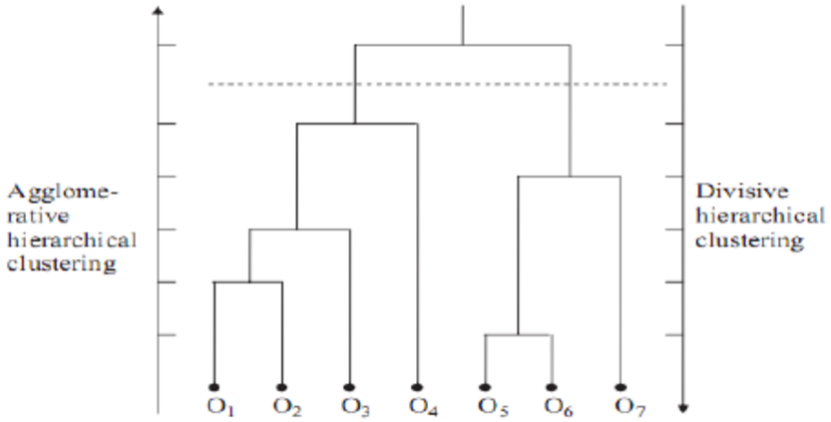


Figure 1.2: Example of hierarchical clustering (both agglomerative and divisive) on seven objects (Sembiring et al., 2011).

Advantages and disadvantages An advantage of this kind of clustering is that it is not necessary to specify a-priori the number of clusters K, which is on the contrary mandatory in partitional algorithms. Moreover, hierarchical algorithms can detect various shapes of clusters (Xu and Tian, 2015; Tomar and Agarwal, 2013). However easy their implementation may be, these procedures have a high time and memory complexities ($O(n^3)$ and $O(n^2)$, respectively (Pandove et al., 2018)) and they are sensitive to outliers.

1.1.3 Density-based clustering

The algorithms belonging to this macro-group share the idea of considering clusters as those areas of the observational space where density of points is higher. The formulation of a density-based clustering procedure requires some initial definitions. We introduce them in the context of DBSCAN (*Density-Based Spatial Clustering of Applications with Noise* (Ester et al., 1996), which is the most popular density-based algorithm.

Before giving these definitions, we have to fix the two tuning parameters of DBSCAN: a real number ϵ and a natural number M, whose meaning is specified in the following. We also select a point distance $d(\cdot, \cdot)$, that could be, for example, the Euclidean distance.

Given a set of n data points $\boldsymbol{x}_i, i = 1, \ldots, n$, we present the first fundamental definition:

Definition 1.1.1. *Given a point $\boldsymbol{x}_i$, the ϵ-neighborhood of $\boldsymbol{x}_i$ is the set of points $N_\epsilon(\boldsymbol{x}_i) = \{\boldsymbol{x}_l : d(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq \epsilon\}$.*

This definition is necessary to operate a first distinction between the data points. Indeed, all points $\boldsymbol{x}_i$ such that $|N_\epsilon(\boldsymbol{x}_i)| \geq M$ are named *core points* and the remaining ones are named *border points*. Another definition is:

Definition 1.1.2. *A point $\boldsymbol{x}_l$ is directly density-reachable from a point $\boldsymbol{x}_i$ w.r.t. ϵ and M if:*

1. $\boldsymbol{x}_l \in N_\epsilon(\boldsymbol{x}_i)$;

2. $\boldsymbol{x}_i$ *is a core point.*

This definition is generalized by the following one that involves chains of "reachable" points.

Definition 1.1.3. *A point $\boldsymbol{x}_l$ is density-reachable from a point $\boldsymbol{x}_i$ w.r.t. ϵ and M if it exists a succession of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_S$ with $\boldsymbol{x}_1 = \boldsymbol{x}_i$ and $\boldsymbol{x}_S = \boldsymbol{x}_l$, such that each point $\boldsymbol{x}_{s+1}$ is directly density-reachable from $\boldsymbol{x}_s$ for $s = 1, \ldots, S-1$.*

It is clear that the relation of density-reachability is not symmetric. To cover this case, a notion of density-connectivity is introduced.

Definition 1.1.4. *A point $\boldsymbol{x}_l$ is density-connected to a point $\boldsymbol{x}_i$ w.r.t. ϵ and M if it exists a point $\boldsymbol{x}_s$ such that both $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ are density-reachable from $\boldsymbol{x}_s$.*

Thanks to all of these definitions, it is possible to define what is a cluster in the density approach. A cluster can be interpreted as a set of density-connected points which is maximal respectively to the relation of density-reachability. In more formal terms:

Definition 1.1.5. *Given a set of data $\mathcal{D}$, a cluster $\mathcal{C}$ w.r.t. ϵ and M is a non-empty subset of $\mathcal{D}$ satisfying the following conditions:*

1. $\forall \boldsymbol{x}_i, \boldsymbol{x}_l$ *with* $i \neq l$ *if* $\boldsymbol{x}_i \in \mathcal{C}$ *and* $\boldsymbol{x}_l$ *is density-reachable from* $\boldsymbol{x}_i$ *then* $\boldsymbol{x}_l \in \mathcal{C}$ *(maximality condition).*

2. $\forall \boldsymbol{x}_i, \boldsymbol{x}_l \in \mathcal{C}$ *with* $i \neq l$, $\boldsymbol{x}_i$ *is density-connected to* $\boldsymbol{x}_l$ *w.r.t.* ϵ *and* M *(connectivity condition).*

Once defined what is a cluster it is possible to name *noise* every point not belonging to a cluster. Figure 1.3 represents all the key-components of DBSCAN: in picture (a) a cluster is depicted; picture (b) shows a core point (in blue), while a border point is depicted in picture (c) (in yellow); in picture (d) a set of density-reachable points is represented.

Now, we describe briefly the main steps of DBSCAN:

1. Select arbitrarily a data point $\boldsymbol{x}_i$.

2. Find the list of points density-reachable from $\boldsymbol{x}_i$.

3. If $\boldsymbol{x}_i$ is a core point, then this list of points forms a cluster, otherwise it is a noise and we have to pass to the next point.

4. Repeat 1-3 until all points are processed.

In the original article of DBSCAN it is provided also an heuristic to choose the best values for ϵ and M, the parameters which indicate how much "dense" the clusters should be.

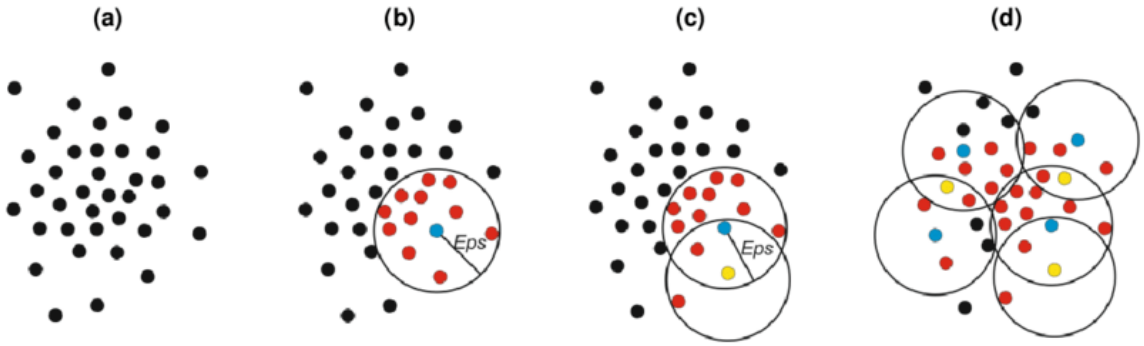


Figure 1.3: An illustration of DBSCAN on a bi-dimensional data set (Entezami et al., 2020).

Advantages and disadvantages DBSCAN does not require to specify at the beginning the number of clusters K and it can also detect clusters of any shapes. It can also manage well the presence of noise. However, it is highly dependent on the choice of the distance $d(\cdot;\cdot)$ and it requires a huge amount of memory if the data set is very large (the memory complexity is $O(n^2)$) (Pandove et al., 2018; Tomar and Agarwal, 2013; Xu and Tian, 2015).

1.1.4 Spectral clustering

Spectral clustering (Von Luxburg, 2007) has its roots in graph theory. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be considered as a set of *vertices* $\mathcal{V} = \{v_1, \dots, v_n\}$ linked by a certain set of *edges* $\mathcal{E} = \{e_1, \dots, e_m\}$. In general, an edge between two vertices v_i and v_l has a non-zero weight w_{il}, if this edge exists. Thus, it is possible construct a matrix $\boldsymbol{W} = (w_{il})_{i,j=1,\dots,n}$ named *adjacency matrix* containing all weights of the graph. If $w_{il} = w_{li}$ for all $i, l = 1, \dots, n$ the graph is said to be *undirected*. It is also useful to define the diagonal *matrix of degree* $\boldsymbol{D} = (d_i)_{i=1,\dots,n}$, where $d_i = \sum_{l=1}^{n} w_{il}$. Given a set of vertices $\mathcal{V}$, the set of subsets $\mathcal{A}_1, \dots, \mathcal{A}_P$ is a *partition* if $\mathcal{A}_p \subset \mathcal{V}, p = 1, \dots, P$, $\mathcal{A}_1 \cup \dots \cup \mathcal{A}_P = \mathcal{V}$ and $\mathcal{A}_i \cap \mathcal{A}_l = \emptyset$, if $i \neq l$.

The idea of spectral clustering is to report the problem of finding homogeneous groups among data to how to recover a partition in a graph. A graph partition has these characteristics: the weights of edges between points belonging to the same $\mathcal{A}_p$ are low and those ones between points of two different $\mathcal{A}_p$ and $\mathcal{A}_{p'}$ are high. The first step to do is to transform a set of data points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ into a graph. It is supposed that a similarity function $s(\boldsymbol{x}_i, \boldsymbol{x}_l)$ (or a distance $d(\boldsymbol{x}_i, \boldsymbol{x}_l)$) has already been defined in order to associate to every couple of points $(\boldsymbol{x}_i, \boldsymbol{x}_l)$. Given this information, a graph can be constructed in different ways. We highlight the most used.

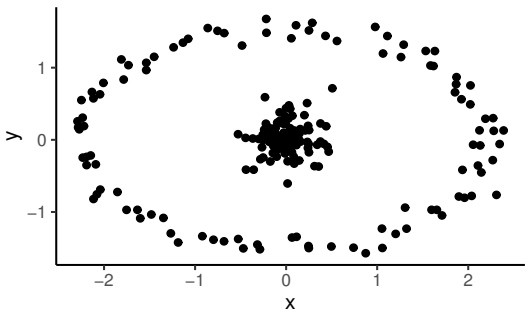
- **ϵ-neighbourhood graph** Two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ are considered as connected, if their similarity (or distance) is less or equal to a certain threshold ϵ. In this case, each edge has the same weight, 1 for instance.
- **K-nearest neighbour graph** In this case $\boldsymbol{x}_i$ is connected to $\boldsymbol{x}_l$ if $\boldsymbol{x}_i$ is among the K-nearest neighbours of $\boldsymbol{x}_l$ (and/or vice versa). Then, the weight of this edge is equal to the similarity $s(\boldsymbol{x}_i, \boldsymbol{x}_l)$ (or distance $d(\boldsymbol{x}_i, \boldsymbol{x}_l)$).
- **Fully-connected graph** All points are connected and each weight $\boldsymbol{w}_{il}$ is equal to their similarity $s(\boldsymbol{x}_i, \boldsymbol{x}_l)$ (or distance $d(\boldsymbol{x}_i, \boldsymbol{x}_l)$).

Once obtained a graph representation of data and the matrices $\boldsymbol{W}$ and $\boldsymbol{D}$, it is possible to obtain a P-partition using the eigenvectors corresponding to the first P non-zero eigenvalues of the matrix $\boldsymbol{L} = \boldsymbol{W} - \boldsymbol{D}$, called *Laplacian* of the graph. Then, in order to obtain a clustering partition, a clustering algorithm is performed on a data matrix of size $n \times P$, where the P columns are the P selected eigenvectors. Here is a brief formulation of the simpler spectral clustering algorithm (Von Luxburg, 2007):

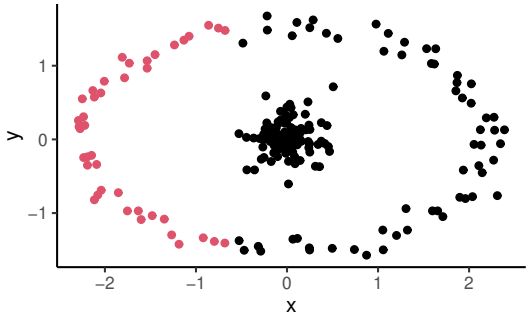
1. Define a similarity function $s(\boldsymbol{x}_i, \boldsymbol{x}_l)$ or a distance $d(\boldsymbol{x}_i, \boldsymbol{x}_l)$.
2. Build a graph representation of data points using one of the approach described before.
3. Calculate the eigenvalues of the Laplacian matrix $\boldsymbol{L}$ and select the eigenvectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_P$ corresponding to the first P ones.
4. Build the $n \times P$ matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_P)$.
5. Perform a clustering algorithm with the number of groups equal to K on the data contained in $\boldsymbol{A}$. A partition $\mathcal{C}_1^*, \ldots, \mathcal{C}_K^*$ is obtained.
6. Associate each point $\boldsymbol{x}_i$ to the cluster $\mathcal{C}_k$ if the point corresponding to the i-th row of A belongs to $\mathcal{C}_k^*$.

Figure 1.4 shows an example of spectral clustering applied to data depicted in Figure 1.4a, where a naive K-means fails (Figure 1.4b). In the first step, the adjacency matrix $\boldsymbol{W}$ is calculated (Figure 1.4c). Then, the first P eigenvectors of the Laplacian $\boldsymbol{L} = \boldsymbol{W} - \boldsymbol{D}$ are extracted and stored in a matrix $\boldsymbol{A}$ (Figure 1.4d). Finally, a good clustering partition is recovered by applying a clustering algorithm (here, a K means) on $\boldsymbol{A}$ (Figure 1.4e).

Advantages and disadvantages Spectral clustering can be used to detect clusters of any shapes and it can deal with categorical variables (it can be based on similarities) and with outliers. On the contrary, its time and memory complexities are high ($O(n^3)$ and $O(n^2)$, respectively) and it highly depends on the choice of $d(\cdot, \cdot)$ (or $s(\cdot, \cdot)$) and on the number of selected eigenvectors (Pandove et al., 2018; Xu and Tian, 2015).



(a)



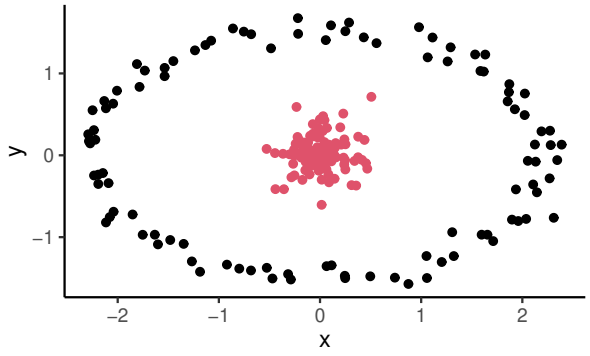
(b)

	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	0	1
4	1	1	0	1	1
5	1	1	1	1	1

(c)

	1	2	3	4	5
1	−0.071	−0.07	−0.004	−0.003	0.006
2	−0.071	−0.07	−0.004	−0.003	0.006
3	−0.071	−0.07	−0.004	−0.003	0.006
4	−0.071	−0.07	−0.004	−0.003	0.006
5	−0.071	−0.07	−0.004	−0.003	0.006

(d)



(e)

Figure 1.4: An example of spectral clustering on a bi-dimensional set. (a) The original data; (b) A bad partition obtained with K-means; (c) An extract of the adjacency matrix $\boldsymbol{W}$; (d) An extract of the eigenvectors of the Laplacian $\boldsymbol{L}$; (e) The recovered partition.

1.2 Model-based clustering

In the previous sections we described clustering methods principally based on geometrical heuristics, such as distances between the data points. Another way of clustering, the *model-based clustering*, has become popular because it allows a well-posed mathematical definition of the clusters. Indeed, it is principally based on maximum likelihood estimation of finite mixture models (McLachlan and Peel, 2004), flexible models that are used in several areas, including density-estimation and robustness analysis.

1.2.1 Models

Finite mixtures assume data come from K different sub-populations with different densities $f_k(\cdot, \boldsymbol{\theta}_k), k = 1, \ldots, K$ of the same shape, indexed by a vector of parameters $\boldsymbol{\theta}_k$. Let consider a set of n observations with D variables. It is supposed that the observations $\mathbf{x} = \{\boldsymbol{x}_i \in \mathbb{R}^D, i = 1, \ldots, n\}$ are i.i.d. and generated according to a D-dimensional mixture with K components, whose probability density function is:

$$\begin{aligned} f(\boldsymbol{x}; \boldsymbol{\psi}) &= \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k) \\ \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k > 0 \quad (k = 1, \ldots, K), \end{aligned} \tag{1.1}$$

where $\boldsymbol{\psi} = (\pi_1, \ldots, \pi_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ contains all the parameters of the mixtures and it belongs to a real space $\boldsymbol{\Psi}$. The set of all possible vectors of proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ is denoted as $\boldsymbol{\Pi}_K$. The parameters $\pi_k, k = 1, \ldots, K$ are called *weights*, while each $f_k(\cdot; \boldsymbol{\theta}_k)$ is a *component* of the mixture.

In (1.1) the shape of the components is not specified. Typically, this choice depends on the nature of data. In statistical and clustering literature, particular importance is given to the *Gaussian mixture models (GMM)*, which assume that each density component $f_k(\cdot; \boldsymbol{\theta}_k)$ has a Gaussian shape. In addition, each Gaussian density is indexed by the mean $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kD})$ and the covariance matrix $\boldsymbol{\Sigma}_k$, which diagonal $(\sigma^2_{k1}, \ldots, \sigma^2_{kD})$ and, thus, it is denoted by $\phi(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Consequently, the probability density function of a Gaussian mixture model is given by a specialization of (1.1), where $f_k(\boldsymbol{x}; \boldsymbol{\theta}_k) = \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for each $k = 1, \ldots, K$. Thus, the vector containing all parameters of a Gaussian mixture model is $\boldsymbol{\psi} = (\pi_1, \ldots, \pi_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$. Due to its prominence in model-based clustering literature and its importance for the rest of the work, we will principally focus on model-based clustering with GMM.

1.2.2 Partition recovery

In model-based clustering the partition is recovered by first providing a good estimate of $\boldsymbol{\psi}$, let say $\widehat{\boldsymbol{\psi}}$, and, then, by applying a decision rule based on it. The estimate $\widehat{\boldsymbol{\psi}}$ is typically calculated by maximizing the likelihood of the model $L(\boldsymbol{\psi}; \mathbf{x}) = \prod_{i=1}^{n} f(\boldsymbol{x}_i; \boldsymbol{\psi})$ or its log-likelihood $\ell(\boldsymbol{\psi}; \mathbf{x}) = \log L(\boldsymbol{\psi}; \mathbf{x})$. For mixture models, log-likelihood maximization is not trivial and it requires special numerical routines as the Expectation-Maximization (EM) algorithm, which is described in the next paragraph. Regarding the decision rule, typically a maximum a posteriori (MAP) rule is chosen to assign observations to the K groups. In the Gaussian case, it means that the estimated labels $\widehat{\boldsymbol{z}} = (\widehat{z}_1, \ldots, \widehat{z}_n)$, where $\widehat{z}_i = k$ if $\boldsymbol{x}_i$ is assigned to $\mathcal{C}_k$, are given by:

$$\widehat{z}_i = \underset{1 \leq k \leq K}{\operatorname{argmax}} \, \widehat{\pi}_k \phi(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k) \quad i = 1, \ldots, n.$$

1.2.3 EM algorithm

In the case of Gaussian mixtures, log-likelihood maximization is hard with common algebraic tools. For this reason, it is usual to recover it through numerical algorithms. The most popular one in statistical literature is the EM algorithm, an iterative routine consisting in maximizing the objective log-likelihood through the maximization of an "easier" function which could be seen as a "surrogate" of the original one. EM algorithm (Dempster et al., 1977) is historically employed to estimate those models when data information is somewhat hidden. Mixture models are part of this category, because it is not known which group each observation $\boldsymbol{x}_i$ comes from. Data membership information, if known, can be resumed in a $n \times k$ matrix $\mathbf{z}$ whose generic element z_{ik} is 1 if $\boldsymbol{x}_i$ belongs to the k-th group, 0 otherwise. An hypothetical knowledge of this hidden information could simplify estimation, as the log-likelihood

$$\ell^c(\boldsymbol{\psi}; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log[\pi_k \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

is very easy to maximize. This log-likelihood is called the *complete log-likelihood*.

Let consider an initial guess for $\boldsymbol{\psi}$, let say $\boldsymbol{\psi}^{(0)}$ and the conditional density $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\psi}^{(0)})$. This conditional density is so defined:

$$p(\mathbf{z}|\mathbf{x}; \boldsymbol{\psi}^{(0)}) \propto \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\frac{\pi_k^{(0)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)})}{f(\boldsymbol{x_i}; \boldsymbol{\psi}^{(0)})} \right)^{z_{ik}}. \tag{1.2}$$

It is shown in Dempster et al. (1977) that the maximization, with respect to $\boldsymbol{\psi}$, of $Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(0)}) = \mathbb{E}_{\boldsymbol{\psi}^{(0)}}[\ell^c(\boldsymbol{\psi}; \mathbf{X}, \mathbf{Z})|\mathbf{X} = \mathbf{x}]$, where $\mathbf{X}$ and $\mathbf{Z}$ are the random variables generating $\mathbf{x}$ and $\mathbf{z}$, regularly increases the objective (or *incomplete*) log-likelihood $\ell(\boldsymbol{\psi}; \mathbf{x})$. This can be iterated for $j \geq 0$ iterations until convergence of a chosen criterion typically

based on absolute or relative log-likelihood increase. Algorithm 1 presents an exhaustive EM algorithm formulation in the case of a Gaussian mixture model. It is worth noting that this algorithm produces closed-form update formulas, which are, thus, easy to apply.

Algorithm 1 EM algorithm for Gaussian mixtures models

1. **Initialization phase**: provide an initial guess $\boldsymbol{\psi}^{(0)}$ and a threshold $\epsilon > 0$.
2. For $j \geq 0$:
   - **E Step:** Given the estimate $\boldsymbol{\psi}^{(j)}$, calculate $Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)}) = \mathbb{E}_{\boldsymbol{\psi}^{(j)}}[\ell^c(\boldsymbol{\psi}; \mathbf{X}, \mathbf{Z})|\mathbf{x}]$.
   - **M Step:** Obtain the new estimate $\boldsymbol{\psi}^{(j+1)} = \text{argmax}_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$. This maximization leads to:

     For $k = 1, \ldots, K$

$$\begin{aligned}
\tau_{ik}^{(j)} &= \frac{\pi_k \phi(\boldsymbol{x}_i; \mu_k^{(j)}, \Sigma_k^{(j)})}{f(\boldsymbol{x}_i; \boldsymbol{\psi}^{(j)})}, \quad i = 1, \ldots, n, \\
\pi_k^{(j+1)} &= \frac{\sum_{i=1}^n \tau_{ik}^{(j)}}{n}, \\
\mu_k^{(j+1)} &= \frac{\sum_{i=1}^n \boldsymbol{x}_i \tau_{ik}^{(j)}}{\sum_{i=1}^n \tau_{ik}^{(j)}}, \\
\Sigma_k^{(j+1)} &= \frac{\sum_{i=1}^n (\boldsymbol{x}_i - \mu_k^{(j+1)})(\boldsymbol{x}_i - \mu_k^{(j+1)})^t \tau_{ik}^{(j)}}{\sum_{i=1}^n \tau_{ik}^{(j)}}.
\end{aligned}$$

   - **Stopping rule**: if $\left| \frac{\ell(\boldsymbol{\psi}^{(j+1)};\mathbf{x}) - \ell(\boldsymbol{\psi}^{(j)};\mathbf{x})}{\ell(\boldsymbol{\psi}^{(j)};\mathbf{x})} \right| < \epsilon$ is verified, continue otherwise.

1.2.4 Advantages and disadvantages

Model-based clustering with GMM can be seen as a generalization of K-means (Fraley and Raftery, 2002). Indeed, suitable specifications of the underlying GMM can help in describing data adequately (Xu and Tian, 2015). As K-means, the main drawback is related to the number of clusters K which has to be a-priori chosen and it is also highly dependent on the specification of the model. But, the fact on being based on a statistical model allows the use of well-posed choice criteria to select both of them (further details will be given in Chapter 4). This is one of the reasons justifying model-based clustering as our approach of reference.

1.3 Frugal clustering for huge data sets

In this thesis we want to propose a *frugal* clustering method able to work with very large data sets (contemporary data sets have millions or billions records (Rajaraman, 2016; Sagiroglu and Sinanc, 2013)), without employing too many resources. Previously, we have described for each clustering approach its advantages and disadvantages taking into account the performances, in terms of accuracy of an algorithm or its ability to discover clusters of any shape. In order to reach frugality, we have to quantify and discuss their complexity. Typically, the analysis of the complexity refers to the asymptotic time demanded by the whole algorithm, which is usually a function of the sample dimension n, the dimensionality of data D and the number of clusters K. Although lowering time complexity is often the main objective of research, we have also to reduce the complexity in terms of memory required by the algorithm, because we suppose to work with limited memory space, too. It is a reliable restriction which is present for example in *edge computing* (Hassan et al., 2019), where machine learning algorithm are executed by front-end sensors and devices with limited memory (Zhang et al., 2019). In Table 1.1 we provide time and space complexity of the techniques presented in the last sections (Pandove et al., 2018).

Algorithm	Time	Memory
K-means	$O(nKD)$	$O(n(D+K))$
PAM	$O(K(n-K)^2)$	$O(n^2)$
Hierarchical	$O(n^3)$	$O(n^2)$
DBSCAN	$O(nlog(n))$	$O(n)$
Spectral	$O(n^3)$	$O(n^2)$
Model-based	$O(nKD)$	$O(nD)$

Table 1.1: Complexity of algorithms presented in Sections 1.1 and 1.2.

Given these asymptotic complexities we try to quantify them practically. Concerning time amount, if n is very large (order of billions), even with a linear algorithm we have to execute billions of elementary operations (unit of time used in complexity theory) that a standard laptop is able to do in hours or in a couple of days. Regarding memory we can say that, using an R implementation (R Core Team, 2021), a vector of 10 billions data (today this is realistic, see, for instance, the UCI Machine Learning repository (Dua and Graff, 2017)) is large around 80 Gb. A such large memory is not available for a common PC. So, to sum up, we have to develop an algorithm sub-linear respectively to n or which does not depend on n. Similar conclusions could be done for the dimensionality D and the number of clusters K, but here we focus on those data sets where n is considerably greater than D and K.

In the last years, researchers have tried to reduce the complexity of traditional techniques in order to allow the analysis of larger and larger data sets, developing algorithms

to perform the so-called *frugal clustering*. However useful they may be for their ideas, some of them remain too burdensome for us, because they are not able to properly deal with time and memory at the same time. There are also different methods using very advanced technologies (such as parallel computer architectures) to be frugal: this could be considered outside the domain of the thesis, because we have decided to work with the limited computational resources provided by a standard laptop.

In the following, we describe different algorithms for frugal clustering for huge data sets, which were extensively reviewed in Pandove et al. (2018). We group these techniques according to their main idea to allow frugality in order to highlight the tools to leverage complexity in clustering.

Thus, we can identify five groups of methods:

1. Data-reduction;
2. Operations reduction;
3. Clustering on transformed space;
4. Subspace clustering;
5. Advanced technologies for clustering.

Before describing each of these groups, we highlight that different algorithms use actually a combination of several techniques to enhance their performances.

1.3.1 Data-reduction

This is probably the biggest group of methods and it contains several algorithms with different approaches to the same main idea: reducing the size of the original data set to make easier and faster the analysis with traditional methods. Firstly, we summarize the most common techniques:

- Subsampling;
- Data summarization;
- Grid-based approach.

Subsampling All the algorithms presented in the following paragraph share the idea of extracting a random subsample of dimension m from the initial data set (with dimension n), such that $m \ll n$. The statistical motivation is quite easy: as n is very big, we can suppose that a subsample of a more reasonable dimension m can provide acceptable results. The length m is usually a tuning parameter and its choice is given by a thread-off

between efficiency and quality of estimation, but, in some cases, m is given by limitation on memory or available resources.

Subsampling is a distinct feature of CLARA (Kaufman and Rousseeuw, 1990), where the classical K-medoids algorithm is performed on a subsample, reducing the quadratical complexity of the original method (PAM). CLARA consists in performing a PAM algorithm on 5 random samples of length $40 + 2K$, retaining the best result. In this way, the total time complexity of CLARA is $O(K(40 + K)^2 + K(n - K))$.

Actually, the subsampling is a very general-purpose method used by several other algorithms present in literature to speed-up execution, even if this is not the main characteristic of them. An example is CURE (Guha et al., 1998). This algorithm performs a scalable agglomerative hierarchical clustering, where the key-idea is to use only c representative points of each cluster to calculate the distance between them (classic hierarchical approach in Section 1.1.2 uses *all* points). These ones are calculated in this way: the best scattered points of a cluster according to a distance criterion are selected and, then, shrunk toward the cluster mean by a factor α. This helps to prevent the presence of highly-influential outliers. Despite the use of particular data structure, such as heaps and trees, to store data in a linear memory space and to simplify the algorithm, CURE has a quadratic time complexity that in the worst case can reach even $O(n^2 \log n)$. Thus, the authors provided two methods to manage very large data sets: subsampling and partitioning. In case of partitioning, the data set is divided into multiple subsets and a double-stage clustering is performed: a first one on each subset, and the second one on the total of clusters found. Using a combination of these two methodologies, CURE is proved to be frugal with a complexity depending only on the dimension of the subsample and/or partition.

Data summarization In this case, data set dimension is reduced by representing a large number of data with a vector of quantities, typically sufficient statistics. The key-idea is that these statistics are able to convey enough information about the groups with an evident gain in terms of complexity and, so, there is no need to store the entire set of data. Once summarized data, each vector of statistics is treated as a single sub-cluster for which an easy generalization of classical clustering algorithms is demanded.

An example of this approach is BIRCH (Tian et al., 1996). In this algorithm, data are scanned once and stored in a tree structure where each node corresponds to a vector of sufficient statistics summarizing a group of data (means, for example), named *clustering feature*. The dimension of the tree is given by memory limits. Once obtained it, an agglomerative clustering is performed on the leaf vectors. Its time complexity is linear but it has some performance limits, because it works well only with spherical clusters. In the original work there are also two optional phases: in the first one it is possible to re-build a smaller tree, removing outliers and merging subclusters into larger ones, while, in the second one, an ulterior refinement of the clustering is performed, using the original

clusters as seeds and redistributing the data points to the closest seeds. In this way, new clusters are obtained.

BFR (Bradley et al., 1998a) uses data summarization for an iterative compressed version of K-means method, where clusters are represented by summarizing statistics (means, sum of squares and dimensions). At each phase, a random subsample of dimension given by memory limitations is extracted from the original data set. This subsample is used to update the current clustering model (an extended version of K-means able to work with summarizing statistics) and, then, data in main memory are divided into these three sets:

1. **Retain set**: these points are retained in the main memory, as they do not belong to any cluster.

2. **Compression set**: points to retain in the main memory, but after a compression in summarizing statistics.

3. **Discard set**: points that can be discarded, as they are represented by statistics contained in the compression set.

The algorithm is then stopped when no point changes cluster, as in classic K-means.

Grid-based approach It is very important to analyze this set of methods because they are similar to the one proposed in this thesis. They consist in imposing a grid on the original sample space and grouping all observations that lie in the same region or cell. In this type of clustering, each cell is considered as a single unit and methods proceed aggregating them using some criteria based on density or entropy. Concerning the grid, it can be regular or adaptive, but it is needed that the number of cells remains inside the limit of the computational resources.

The first example of such algorithms is CLIQUE (Agrawal et al., 1998), which we now briefly describe. Considering a D-dimensional data set whose variables are $(X_1, \ldots, X_D)$, CLIQUE divides the space of each variable $X_d, d = 1, \ldots, D$ into several units of length ϵ (which is, thus, a tuning parameter), obtaining a list of one-dimensional units. The final units are built as product of the one-dimensional ones and they are considered as *dense* if the number of points inside them is over a certain threshold τ. Then, similar to the density-clustering approach, a cluster is defined as a set of *connected* units, i.e., units that share a common face.

As discovering all connected units in all subspaces is infeasible, CLIQUE employs some heuristics to reduce complexity. Firstly, it reduces the number of clusters to analyze, because it can be proved that if some units form a cluster in a subspace of dimension D^*, they are part of the same cluster projected in a subspace of dimension $D^* - 1$. So, if some units do not form a cluster in a subspace of dimension $D^* - 1$, they are discarded them when clusters in upper dimensions are searched. Finally, it is also possible to reduce the

number of clusters, considering only those ones in subspaces with a high *coverage*, which is defined as the fraction of points located in the dense units.

The algorithm ends recovering for each cluster a minimal and non-redundant description as sum of rectangles. Totally, the time complexity of CLIQUE is linear with respect to n and quadratically with respect to D, as shown by several simulations in the original paper.

There are other methods could be considered as slight variation of CLIQUE schema: in ENCLUE (Cheng et al., 1999) a new entropy criterion is added to correct some bad behaviours of CLIQUE, while in MAFIA (Goil et al., 1999) regular grids are substituted by adaptive grids, varying according the difference in density between two consecutive cells (if it is too tight, those cells are merged, making the grid coarser).

1.3.2 Operation reduction

While the main idea in the last subsection was to reduce the size of data set, here we will focus on how to build clustering algorithm with a minimum number of operations. Some bottlenecks of traditional algorithms are usually multiple scanning of the data set and the analysis of clusters that are not significant in the context of reference. In order to solve the first issue, some authors developed techniques able to scan all data few times, building also particular data structure such as tree or graph. We have already mentioned BIRCH and CLARA, where such structures are present. An improvement of CLARA, named CLARANS (Ng and Han, 2002), uses a graph architecture to reduce the number of comparisons to do. Indeed, it is possible to build a graph $\mathcal{G}_{n,K}$ (for a K-medoids algorithm applied on a sample with n observations), such that each node S is a set of medoids $\mathcal{M}$. Then, two nodes S_1 and S_2 are neighbors (i.e. connected by an arc) if the two sets of medoids $\mathcal{M}_1$ and $\mathcal{M}_2$ differ just for one medoid. And, if each node S is associated to the dissimilarity cost of its induced partition, its clear that the cost difference between two neighbors is equal to the cost usually minimized in a K-medoids algorithm. Thus, the clustering problem can be viewed as a search for a minimum in a graph $\mathcal{G}_{n,k}$.

In this new view, PAM can be seen as a search for a minimum in the complete graph $\mathcal{G}_{n,k}$, which is an optimal but complex process, while CLARA is a sub-optimal search in a simpler graph $\mathcal{G}_{n,K}$. Despite this, both algorithms consist in repetitively selecting an arbitrary node on the graph and then comparing its cost to those ones of its neighbors, until all the nodes are explored. In order to reach both optimality and simplicity, CLARANS works on the original graph $\mathcal{G}_{n,K}$, but instead of checking all neighbors of a random selected node it controls only a prefixed number of them, after a random selection. Moreover, it is also possible to fix a maximum number of iterations for this operation, as it is not demanded to exhaust the list of nodes.

It is worth also mentioning an example of spectral clustering, the USPEC method (Huang et al., 2019). In this technique, in addition to an hybrid strategy of subsampling and K-means to select p representative points, it is possible to build an adjacency matrix

of dimension $n \times p$ where each point x_i is linked only to its K-nearest representative points. This has important advantages because the obtained graph is bi-partite and it has structural properties allowing a remarkable simplification in eigenvalues calculus. In this way, the whole time complexity of the algorithm is $O(nD\sqrt{p})$ and the memory cost $O(nK)$ (or $O(n\sqrt{p})$, depending on the implementation).

For the second problem, several heuristics to prune irrelevant clusters are available. In fact, different algorithms employ criteria based on density (CLIQUE) or other measures, like correlation (ENCLUE), to select only the most relevant clusters and delete the others, decreasing the complexity. The application of these heuristics is strongly related to the subspace clustering (Section 1.3.3), because they help to exclude all the clusters formed in lower on higher dimensions by a cluster not respecting them.

1.3.3 Clustering on transformed space/subspace

We choose to regroup those two ideas because they both refer to the more general intuition of working in spaces where clustering is easier. The difference between them is that in the former clusters are usually searched into a transformation of the original space and then re-transformed in clusters to obtain an interpretable result, while in the latter the research is restricted on small subspaces of the complete space in order to avoid the problem of the dimensionality.

WaveCluster (Sheikholeslami et al., 1998) uses a wavelet transform over the feature space. This algorithm starts with the construction of units similar to the grid approach (in this case, this step is called *quantization*), then these ones are transformed by a wavelet function. It is possible to re-apply it many other times in order to obtain a multi-resolution representation of the transformed space. Then, at each different level, connected units are recognised and mapped back to the clusters in the original sample space. The time complexity of WaveCluster is $O(n)$ and, furthermore, it is possible to reduce noise thanks to the application of the wavelet transformation.

Clustering on subspaces is well suited to analyze data sets with a high number of variables. As clusters in CLIQUE and MAFIA are connected units inside particular subspaces, these algorithms belong to this group. We can also mention SUBCLU (Kailing et al., 2004), which can be considered a subspace version of DBSCAN. Indeed, it starts generating all clusters in each one-dimensional subspaces using DBSCAN. Then, knowing the clusters in subspaces of dimension D^*, it generates iteratively the ones in subspaces of dimension D^*+1, where clusters are searched through DBSCAN. This algorithm, in order to alleviate computational burden, exploits monotonicity between connected subspaces to conclude that ,if a subspace of dimension D^* does not include a cluster in any of its subspaces of dimension lower than D^*, it can not include clusters. Moreover, another heuristic, based on the minimum number of object contained in a cluster, is used to select only the best subspace among those of a fixed dimension. DUSC (Assent et al., 2007) is an improvement of SUBCLU which introduces a new density measure, motivated by

the fact that two densities in two different subspaces are not comparable (*dimensionality bias*).

The method known as nCluster (Liu et al., 2007) introduces new heuristics to be used in a subspace clustering context. Given a set of objects $\mathcal{O}$ and a set of attributes $\mathcal{A}$ and a threshold δ, two objects $x, y \in \mathcal{O}$ are *neighbors* in a subset of attributes $\mathcal{H} \subset \mathcal{A}$ if, for each $a \in \mathcal{H}$ (a continuous), $|v_{xa} - v_{ya}| < \delta R_a$, where v_{xa}, v_{ya} are the values of the attribute a for object x and y and R_a is the range of values for a. If a is a categorical attribute, it is demanded to have $v_{xa} = v_{ya}$. With this statement, *δ-nClusters* are defined by the couple $(\mathcal{T}, \mathcal{H})$, where $\mathcal{T} \subset \mathcal{O}$ and $\mathcal{H} \subset \mathcal{A}$, such that for each attribute $a \in \mathcal{H}$ any two objects $x, y \in \mathcal{T}$ are neighbors. Moreover, if for a δ-nClusters $(\mathcal{T}, \mathcal{H})$ it does not exist another δ-nClusters $(\mathcal{T}', \mathcal{H}')$ such that $\mathcal{T} \subset \mathcal{T}'$, $\mathcal{H} \subset \mathcal{H}'$, then $(\mathcal{T}, \mathcal{H})$ is called *maximal δ-nClusters*. In effect, nCluster searches for these particular clusters, selecting those with a minimum number of observations and attributes. Similarly to the previous algorithms, this method starts from discovering one-dimensional clusters, then it generates clusters in upper dimensions using monotonic properties similar to the previous ones to prune subspaces useless to analyze.

1.3.4 Advanced technologies for clustering

Big data arose due to the technological development of the last decades. It is also true that this phenomenon has also brought new technologies to be used in statistics and in computer science. In particular, we refer to those instruments able to heavily reduce time execution of every computing task enabling parallelization. Thanks to the advances in communications, now it is possible to build frameworks of several computing machines that are able to execute any task collaboratively at the same time, according to the paradigm *divide et impera*. Among these ones, we mention MapReduce (Dean and Ghemawat, 2008) and Spark (Zaharia et al., 2012), which are intensively used in clustering and they help to obtain results from classical clustering algorithms quickly. Indeed, enhanced versions of K-means, as PKMeans (Zhao et al., 2009) and SOKM (Zayani et al., 2016), have been proposed. Briefly speaking, they use an architecture called master-slave, where a machine is a master node and the others slaves. Generally, the huge amount of data is divided into several little data sets, which are analyzed singularly by each slave node. The master node assigns the tasks to each slave one, it coordinates their activities, and, at the end, it joins together the results. It is straightforward that these methods are really powerful, but it is also obvious they request lots of machines and the availability of infrastructures allowing the communication between each node of the architecture and data set partition. Thus, we consider all the algorithms employing them outside the domain of the thesis.

1.4 Frugal model-based clustering

At the end of Section 1.2, we specified model-based clustering would have been our approach of reference. But, the time complexity of EM algorithm is linearly related to the sample size n as reported in Table 1.1, making it not directly applicable to huge data sets. Thus, motivated by the good results of model-based clustering for moderate size data, researchers have designed frugal implementations for model-based clustering when very massive data sets have to be analyzed. The strategies used are the same we have described in the previous section.

Subsampling This technique is widely employed to make model-based clustering frugal (Fraley and Raftery, 2002; Banfield and Raftery, 1993; Tsapanos et al., 2016). It is appreciated for its simplicity and speed as it consists in applying EM algorithm on a randomly selected subsample. However, this method is generally prone to inaccuracy and variability (DuMouchel et al., 1999). In addition, this approach becomes critical in presence of very small classes as the random subsample could not contain any representatives of this class: this is usual when the analyzed data set is *imbalanced*, as shown in Section 1.5.1.

Data summarization In this category we find EMADS (Jin et al., 2005) and the algorithm described in Moore (1998), where sufficient statistics (means and covariances, in particular) are used to represent group of points. The main criticality of these methods is in the structures they use to store summaries, which are multidimensional grids and multiresolution KD-trees. Indeed, the dimension of them (and, thus, the corresponding memory occupancy) can explode even if dimension D is moderate and also the speed-up of the algorithms with respect to classic EM declines rapidly if D increases (see, for example, Moore (1998)). Another similar approach is contained in Bradley et al. (1998b), where a key-role is also played by subsampling, so it inherits all its drawbacks.

Advanced technologies EM algorithm has also been parallelized and adapted to Map-Reduce applications (Wolfe et al., 2008). As written in Section 1.3.4, these paradigms are considered outside the domain of the thesis, because they require particular technological infrastructures.

1.5 Imbalanced data sets

In the previous sections, we have talked about the challenges of clustering very huge data sets. Here, we introduce a second category of data collection, which contributes to completely define the context of our thesis: *imbalanced* data sets. The main characteristic of these data is the presence of at least one class which is very tiny with respect to the total. This is usual in several fields, such as credit card fraud detection (Chan and Stolfo,

1998), cancer recognition (Yu et al., 2012), fraudulent calls (Fawcett and Provost, 1997), where typically very few anomalies have to be distinguished from normal events.

Imbalanced data sets are usually analyzed in classification settings, where class labels are known. The most employed techniques consist in the creation of an artificial balanced data set in a pre-preprocessing stage, by oversampling the minority class (Chawla et al., 2002), or undersampling (Tahir et al., 2009) the majority one. In this thesis, we focus on solving the corresponding clustering problem, motivated by the fact that labelling records could be sometimes difficult, especially when sample size is large. Our purpose has been ultimately strengthened by the explosion of Big Data, which has made possible the availability of very large data sets, mostly imbalanced (Leevy et al., 2018; Fernández et al., 2017).

1.5.1 Subsampling with imbalanced data sets

In Section 1.4, we have written that subsampling becomes critical in presence of tiny classes. This is because, if the small class proportion is very low and the subsample size S is small, a random subsample could not contain any representatives of this class. This is realistic, as our memory constraints are strong. Consequently, designing a frugal clustering technique based on subsampling is not advised if the data set to be analyzed is imbalanced. In this section, we empirically illustrate this fact showing how many different subsamples "miss" the small class of an imbalanced data set under strong memory constraints. This experience considers 200 subsamples of various size $S = 50, 100, 200, 400$ extracted from three simulated data sets with 10^6 records generated according to three 2-class univariate mixtures with densities

$$
\begin{aligned}
f_1(x;\boldsymbol{\psi}) &= 10^{-2}\phi(x;-4,1) + (1-10^{-2})\phi(x;4,1)\\
f_2(x;\boldsymbol{\psi}) &= 10^{-3}\phi(x;-4,1) + (1-10^{-3})\phi(x;4,1)\\
f_3(x;\boldsymbol{\psi}) &= 10^{-4}\phi(x;-4,1) + (1-10^{-4})\phi(x;4,1)
\end{aligned}
$$

Thus, what it changes between the three mixtures is the proportion of the small class π_1, which is decreasing. Consequently, imbalance is increasing. In Figure 1.5 we shows the percentage of subsamples containing representatives of the small class for each scenario and for each subsample size. This experience confirms our first intuition about subsampling in case of highly imbalanced data sets and strong memory constraints: the probability of "missing" the small class is not negligible (especially if the maximum sample size S is small) and it is increasing with the imbalance. Consequently, a new data-reduction approach is needed: our proposal is based on binned data, which are introduced in the next section.

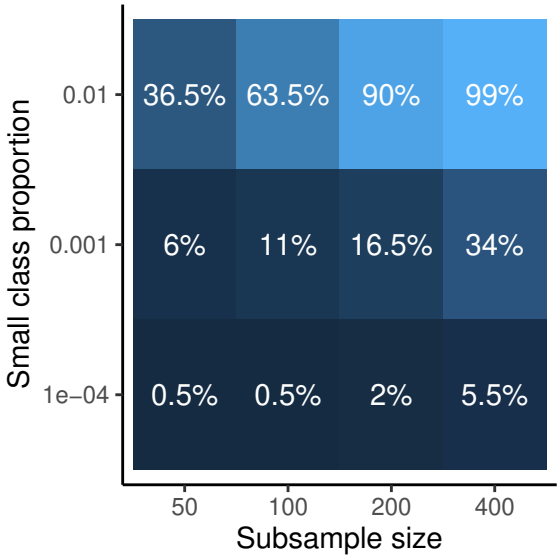


Figure 1.5: Percentage of subsamples (out of 200 samples) containing representatives of the small class.

1.6 Binned data

In this section we introduce binned data, which correspond to the counts of raw data in given regions of the sample space (McLachlan and Jones, 1988). They typically arise when it is impossible to collect data with infinite precision and some phenomena, like truncation or rounding, can happen. In this work, we employ them for frugality purposes. Let consider a sample $\mathbf{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ composed by n observations belonging to a real D-dimensional space $\mathcal{X} \subset \mathbb{R}^D$ and a partition $\{\mathcal{B}_b \subset \mathbb{R}^D, b = 1, \ldots, B\} \subset \mathcal{X}$. Binned data are defined as the vector $\boldsymbol{n} = (n_1, \ldots, n_B)$, where each element n_b is the number of observations lying inside the region $\mathcal{B}_b$. Thus, $n_b = \#\{\boldsymbol{x}_i \in \mathcal{B}_b\}$.

In this work, we suppose binning regions to be D-dimensional real intervals. In this case, we can hypothesize that binning regions are delimited by a D-dimensional Cartesian grid, which can be named *binning grid*. Formally, we can assume that the grid $G = G_1 \times \ldots \times G_D$ is the Cartesian product between D univariate grids G_d with $R_d + 2$ cut points $(a_{d0}, \ldots, a_{d(R_d+1)})$, where $a_{d0} = -\infty$ and $a_{d(R_d+1)} = \infty$. This grid, whose *refinement* is defined as $R = \prod_{d=1}^{D} R_d$, divides the sample space into $B = \prod_{d=1}^{D}(R_d + 1)$ real intervals of dimension D. Each region (or *bin*) is defined as $\mathcal{B}_b = \bigotimes_{d=1}^{D}[a_{d(b_d-1)}, a_{db_d})$, where $(b_1, \ldots, b_D)$ is a vector of indices satisfying

$$b = b_1 + \sum_{d=2}^{D}(b_d - 1)\prod_{d'=1}^{d-1}(R_{d'} + 1), \tag{1.3}$$

with $b_d \in \{1, \ldots, R_d + 1\}$, for each $d = 1, \ldots, D$.

Let assume that observations $\boldsymbol{x}_i$ are i.i.d realizations of a random real variable $\mathbf{X}$ with parametric density $f(\boldsymbol{x};\boldsymbol{\psi})$ indexed by the vector of parameters $\boldsymbol{\psi}$, which has to be estimated. The whole set of parameter is denoted as $\boldsymbol{\Psi}$ and it is typically assumed to be an Euclidean set. Under these assumptions, binned data $\boldsymbol{n}$ are modelled by the multinomial density:

$$p(\boldsymbol{n};\boldsymbol{\psi}) \propto \prod_{b=1}^{B} \Big(\int_{B_b} f(\boldsymbol{x};\boldsymbol{\psi}) d\boldsymbol{x} \Big)^{n_b}. \tag{1.4}$$

This means that the only knowledge of binned data has important consequences in all the algorithms which aim to estimate $\boldsymbol{\psi}$, as we have to treat the model given by (1.4), instead of the raw model with density $f(\boldsymbol{x};\boldsymbol{\psi})$.

1.6.1 Gaussian mixture models with binned data

If the random variable $\mathbf{X}$ follows a Gaussian mixture model with density $f(\boldsymbol{x};\boldsymbol{\psi})$ given by $\sum_{k=1}^{K} \pi_k \phi(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$, binned data density specializes in

$$p(\boldsymbol{n};\boldsymbol{\psi}) \propto \prod_{b=1}^{B} \Big(\sum_{k=1}^{K} \pi_k \int_{B_b} \phi(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) d\boldsymbol{x} \Big)^{n_b}. \tag{1.5}$$

As mentioned in the previous section, the usage of binned data changes model definition and thus its estimation. Consequently, the EM algorithm described in Algorithm 1 can not be directly applied to binned mixtures. For this reason, a suitable EM algorithm was introduced in McLachlan and Jones (1988); Cadez et al. (2002). They proposed to consider the couple composed by *raw* data $\mathbf{x}$ and labels $\mathbf{z}$ as hidden information sources and, thus, use the EM machinery on the complete log-likelihood

$$\ell^c(\boldsymbol{\psi};\mathbf{x},\mathbf{z}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log(\pi_k \phi(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)).$$

In the initial iteration, once fixed the initial guess $\boldsymbol{\psi}^{(0)}$, they propose to maximize the quantity $Q(\boldsymbol{\psi},\boldsymbol{\psi}^{(0)}) = \mathbb{E}_{\boldsymbol{\psi}^{(0)}}[\ell^c(\boldsymbol{\psi};\mathbf{X},\mathbf{Z})|\mathbf{n}]$, which is calculated with respect to the conditional density $p(\mathbf{x},\mathbf{z}|\mathbf{n};\boldsymbol{\psi}^{(0)})$. This maximization is then repeated at each iteration $j \geq 0$, until the algorithm is stopped due to the convergence of a chosen criterion based on log-likelihood relative or absolute increasing. Algorithm 2 shows briefly the complete procedure associated to the binned EM algorithm for multivariate Gaussian mixtures.

Algorithm 2 Bin-EM algorithm for multivariate Gaussian mixtures models

1. **Initialization phase**: provide an initial guess $\boldsymbol{\psi}^{(0)}$ and a threshold $\epsilon > 0$.
2. For $j \geq 0$:
   - **E-Step:** Given the estimate $\boldsymbol{\psi}^{(j)}$, calculate $Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)}) = \mathbb{E}_{\boldsymbol{\psi}^{(j)}}[\ell^c(\boldsymbol{\psi}; \mathbf{X}, \mathbf{Z})|\boldsymbol{n}]$.
   - **M-Step:** Obtain the new estimate $\boldsymbol{\psi}^{(j+1)} = \text{argmax}_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$. This maximization leads to:

     For $b = 1, \ldots, B$

$$g_b(\boldsymbol{x}) = \frac{f(\boldsymbol{x}, \boldsymbol{\psi}^{(j)})}{\int_{\mathcal{B}_b} f(\boldsymbol{x}, \boldsymbol{\psi}^{(j)}) d\boldsymbol{x}}$$

     For $k = 1, \ldots, K$

$$\begin{aligned}
\tau_k^{(j)}(\boldsymbol{x}) &= \frac{\hat{\pi}_k \phi(\boldsymbol{x}, \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)})}{f(\boldsymbol{x}, \boldsymbol{\psi}^{(j)})} \\
\pi_k^{(j+1)} &= \frac{\sum_{b=1}^{B} n_b \int_{\mathcal{B}_b} \tau_k^{(j)}(\boldsymbol{x}) g_b(\boldsymbol{x}) d\boldsymbol{x}}{n} \\
\boldsymbol{\mu}_k^{(j+1)} &= \frac{\sum_{b=1}^{B} n_b \int_{\mathcal{B}_b} \boldsymbol{x} \tau_k^{(j)}(\boldsymbol{x}) g_b(\boldsymbol{x}) d\boldsymbol{x}}{\sum_{b=1}^{B} n_b \int_{\mathcal{B}_b} \tau_k^{(j)}(\boldsymbol{x}) g_b(\boldsymbol{x}) d\boldsymbol{x}} \\
\boldsymbol{\Sigma}_k^{(j+1)} &= \frac{\sum_{b=1}^{B} n_b \int_{\mathcal{B}_b} (\boldsymbol{x} - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x} - \boldsymbol{\mu}_k^{(j+1)})^t \tau_k^{(j)}(\boldsymbol{x}) g_b(\boldsymbol{x}) d\boldsymbol{x}}{\sum_{b=1}^{B} n_b \int_{\mathcal{B}_b} \tau_k^{(j)}(\boldsymbol{x}) g_b(\boldsymbol{x}) d\boldsymbol{x}}
\end{aligned}$$

   - **Stopping rule**: Stop if $\left| \frac{\ell(\boldsymbol{\psi}^{(j+1)}; \boldsymbol{n}) - \ell(\boldsymbol{\psi}^{(j)}; \boldsymbol{n})}{\ell(\boldsymbol{\psi}^{(j)}; \boldsymbol{n})} \right| < \epsilon$ is verified, continue otherwise.

1.7 Contribution of the thesis

In describing the frugal approaches to model-based clustering in Section 1.4, we can note that the only technique we can employ frugally with very strong computational constraints is the random subsampling. But, it has some disadvantages: on the one hand, it is prone to high variability depending on the quality of the selected subsample; on the other hand, if the data set to analyze is imbalanced and the maximum possible size for the subsample is really small, there is a high possibility that a tiny but important class (which characterizes imbalanced data) is not represented in the extracted subsample,

as shown in Section 1.5.1. This means that we could never detect the presence of this relevant cluster. For this reason, we propose a frugal model-based clustering method overperforming subsampling in presence of imbalanced data sets and strong computational constraints. This technique is principally based on reducing the size of the original raw data into a new highly compressed data set composed by a collection of binned data, which are artificially built in a way specified in the following chapters.

In this thesis, we present two main contributions. In Chapter 2, we present the idea to build artificially binned data through a binning grid in order to reduce the dimensionality of the problem. Then, we first apply this method to univariate Gaussian mixture estimations, providing numerical simulations. A remarkable theoretical contribution is given in the same chapter, as we demonstrate the identifiability of univariate binned Gaussian mixtures. In Chapter 3, we illustrate our contribution for multivariate diagonal Gaussian mixtures. Our solution relies on the combination between binned data (used in a marginal fashion) and composite likelihood (Lindsay, 1988). Marginal binned data avoid to store D-variate binned data in our limited memory, as their size rapidly explodes even if D is moderately high ($D = 3$ or $D = 4$). Composite likelihood is used to circumvent a computational problem given by a naive full likelihood estimation of the marginal binned Gaussian mixture arising from our bin-marginal data reduction. The introduction of both of these tools will be motivated and discussed in Chapter 3. Further topics that are complementary to our multivariate contribution are discussed in Chapter 4. In particular, we discuss experimentally local maxima, a typical issue in Gaussian mixture estimation, showing how it occurs in our case. A similar approach is used to illustrate the influence of the binning grid on the estimation. In the same chapter, we also propose some guidelines to choose the number of components of the mixture if this is unknown. Chapter 5 contains a real-data application of the proposed method on anomaly detection in time series.

Chapter 2

Frugal univariate Gaussian mixtures with binned data

At the end of the previous chapter we have introduced binned data, defining them as the result of an incomplete and imprecise observational process. At a first sight, the only knowledge of binned data seems to complicate estimation process, as raw data information is not available. But, they can help in our search of frugality. Indeed, if the raw sample $\mathbf{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ has length n and the binned data vector $\boldsymbol{n}$ has length B and $B \ll n$, it is straighforward to note that storing binned data is much more frugal than saving all raw data, even if the corresponding grid structure (its *cut points*) must be also counted. Thus, our first key idea is to group raw data in order to obtain *artificially* binned data, that are used in estimation tasks instead of raw ones. In our case, the binning operation is done with the help of the Cartesian binning grid introduced in Section 1.6. In the following paragraphs, we apply this idea to two univariate settings, where data binning reveals all of its potential in simple situations. As we work in a univariate context, we can simplify notation to facilitate reading. Indeed, a univariate grid G of refinement R is defined by $R + 2$ cut points $(a_0, \ldots, a_{R+1})$, where $a_0 = -\infty$ and $a_{R+1} = \infty$. Accordingly, binned data vector $\boldsymbol{n}$ will have length equal to $B = R + 1$ and each generic element n_b is so defined:

$$n_b = \#\{x_i : a_{b-1} \leq x_i < a_b\}, \quad b = 1, \ldots, B.$$

In the following, we will employ a special class of regular grids assuming equidistant cut-points. This assumption facilitates our investigation about the influence of the binning process on the quality of estimation, in dependence of the only degree of refinement R. Returning to the contents of the chapter, we first deeply analyze the estimation of a Gaussian with unknown mean. This preliminary work highlights good theoretical properties of the binned maximum likelihood estimator and its dependence in function of the chosen binning grid. Finally, we focus on how to estimate univariate Gaussian mixtures in presence of binned data. We will employ a univariate version of Algorithm 1 to experimentally show the computational savings given by our binned strategy. In addition, an important result about identifiability of univariate Gaussian mixtures in presence of binned data is provided.

2.1 Preliminary work: a single univariate Gaussian

In the specific case of this section, we suppose that the sample $\mathbf{x} = \{x_1, \ldots, x_n\}$ arises from n i.i.d. univariate Gaussian $N(\mu, \sigma^2)$ outcomes with density $\phi(\cdot; \mu, \sigma^2)$, where μ denotes the mean and σ^2 the variance. Here, we aim to deal with three important questions regarding the identifiability of the model when using binned data (Section 2.1.1), the properties of the maximum likelihood estimator (Section 2.1.2) and the influence of the binning grid on statistical accuracy, proposing also a choice criterion between two grids (Section 2.1.3).

In Section 2.1.2 and 2.1.3 we also make two additional hypotheses. First, the variance σ^2 is known and equal to 1. Second, the grids considered are equispaced and symmetric around μ. With these last regularity assumptions, the grids are simply indexed by two parameters which are the number of points R and the "starting" point a_1. Consequently, in these sections we denote each grid with $G(a_1, R)$.

2.1.1 Identifiability

We are interested by a fundamental probabilistic property of a model which is the identifiability. A parametric model is identifiable if there is a one-to-one relationship between each density and each parameter. Here is a formal definition of the concept.

Definition 2.1.1. *A parametric model $\mathcal{M} = \{f(\cdot; \boldsymbol{\psi}) : \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$ is said to be identifiable if*

$$\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi} \quad f(\cdot; \boldsymbol{\psi}) = f(\cdot; \boldsymbol{\psi}') \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}'. \tag{2.1}$$

In case of a model in presence of binned data, this definition has to be specialized due to the presence of the grid G. We can reasonably argue that identifiability must hold whatever the grid may be.

Definition 2.1.2. *In presence of binned data, a parametric model $\mathcal{P} = \{p(\cdot; \boldsymbol{\psi}) : \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$ is said to be identifiable if*

$$\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi} : \ p(\boldsymbol{n}; \boldsymbol{\psi}) = p(\boldsymbol{n}; \boldsymbol{\psi}') \ \forall G, \boldsymbol{n} \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}'. \tag{2.2}$$

In case of a single Gaussian distribution, we have to prove that each binned multinomial density $p(\boldsymbol{n}; \mu, \sigma^2) \propto \prod_{b=1}^{B} \left(\int_{a_{b-1}}^{a_b} \phi(x; \mu, \sigma^2) dx \right)^{n_b}$ is indexed by only one couple (μ, σ^2). This is true only for specific grids as the following proposition states.

Proposition 2.1.1. *Binned univariate normal models are identifiable for $R \geq 2$.*

Proof. Considering Definition 2.1.2 and denoting with $\Phi(\cdot)$ the cumulative density function (c.d.f.) of a standard Gaussian, if the grid G has R finite cut points $(a_1, \ldots, a_R)$ then

it is sufficient to prove that the system

$$\begin{cases} \Phi(\frac{a_1-\mu}{\sigma}) = \Phi(\frac{a_1-\mu'}{\sigma'}) \\ \vdots \\ \Phi(\frac{a_R-\mu}{\sigma}) = \Phi(\frac{a_R-\mu'}{\sigma'}) \end{cases}$$

has only the trivial solution $\boldsymbol{\psi} = \boldsymbol{\psi}'$ whatever the grid is. Due to the monotonicity of the Gaussian c.d.f., it is equivalent to:

$$\begin{cases} \frac{a_1-\mu}{\sigma} = \frac{a_1-\mu'}{\sigma'} \\ \vdots \\ \frac{a_R-\mu}{\sigma} = \frac{a_R-\mu'}{\sigma'} \end{cases}$$

It is straighforward to prove that if $R \geq 2$, $\boldsymbol{\psi} = \boldsymbol{\psi}'$ is the only solution and, thus, identifiability is achieved. □

2.1.2 Estimators properties

In this section, we aim to analyze the statistical properties of the binned maximum likelihood estimator (MLE) of μ obtained from the binned data set $\boldsymbol{n}$. In particular, we study the bias and the variance of this estimator. To simplify, we use the same notation to denote both the estimator (which is a random variable) and the estimate, which is a particular realization of the estimator. Specifically, in this section the general estimate is denoted as $\widehat{\mu}^b_{a_1,R}$ to highlight the dependence on the equispaced grid $G(a_1, R)$ symmetric around μ.

The following proposition assures that the binned estimator is asymptotically unbiased and that its asymptotic variance converges to the asymptotic variance of the raw MLE $\widehat{\mu}^{MLE}$ as the binning grid becomes wider and finer:

Proposition 2.1.2. *$\widehat{\mu}^b_{a_1,R}$ is asymptotically unbiased and* $\lim_{\substack{a_1\to-\infty \\ R\to+\infty}} Var(\widehat{\mu}^b_{a_1,R}) = Var(\widehat{\mu}^{MLE})$.

Proof. The estimator $\widehat{\mu}^b_{a_1,R}$ is asymptotically unbiased as it is a maximum likelihood estimator (regularity conditions provided by Rao (1957) are satisfied). Given the model log-likelihood

$$\ell(\mu; \boldsymbol{n}) = \sum_{b=1}^{B} n_b \log\Big[\Phi\Big(\frac{a_b-\mu}{\sigma}\Big) - \Phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big)\Big],$$

it follows from maximum likelihood theory that the asymptotic variance $Var(\widehat{\mu}^b_{a_1,R})$ is equal to the reverse of the expected information matrix

$$\mathcal{I}(\mu) = \mathbb{E}[-\ell''(\mu)] = \frac{n}{\sigma^2} \sum_{b=1}^{B} \frac{(\phi(\frac{a_b-\mu}{\sigma}) - \phi(\frac{a_{b-1}-\mu}{\sigma}))^2}{\Phi(\frac{a_b-\mu}{\sigma}) - \Phi(\frac{a_{b-1}-\mu}{\sigma})}.$$

Therefore the asymptotic variance of $\widehat{\mu}_b^2$ is

$$\mathrm{Var}[\widehat{\mu}_b^2] = \frac{1}{\mathcal{I}(\mu)} = \frac{\sigma^2}{n\sum_{b=1}^{B}\frac{(\phi(\frac{a_b-\mu}{\sigma})-\phi(\frac{a_{b-1}-\mu}{\sigma}))^2}{\Phi(\frac{a_b-\mu}{\sigma})-\Phi(\frac{a_{b-1}-\mu}{\sigma})}}.$$

As $Var(\widehat{\mu}^{MLE}) = \frac{\sigma^2}{n}$, the quotient of the two variances is equal to:

$$\sum_{b=1}^{B}\frac{(\phi(\frac{a_b-\mu}{\sigma})-\phi(\frac{a_{b-1}-\mu}{\sigma}))^2}{\Phi(\frac{a_b-\mu}{\sigma})-\Phi(\frac{a_{b-1}-\mu}{\sigma})}.$$

When R is big enough, we can introduce the following approximations for $2 \leq b \leq B-1$:

$$\begin{aligned} a_b &\approx a_{b-1} + h; \\ \phi\Big(\frac{a_b-\mu}{\sigma}\Big) - \phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big) &\approx -\frac{h}{\sigma}\frac{a_{b-1}-\mu}{\sigma}\phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big); \\ \Phi\Big(\frac{a_b-\mu}{\sigma}\Big) - \Phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big) &\approx \frac{h}{\sigma}\phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big), \end{aligned}$$

with $h \to 0$. Using these approximations and the symmetry of the Gaussian density, it is possible to write:

$$\begin{aligned} \sum_{b=1}^{B}\frac{(\phi(\frac{a_b-\mu}{\sigma})-\phi(\frac{a_{b-1}-\mu}{\sigma}))^2}{\Phi(\frac{a_b-\mu}{\sigma})-\Phi(\frac{a_{b-1}-\mu}{\sigma})} &= \frac{2\phi^2(\frac{a_1-\mu}{\sigma})}{\Phi(\frac{a_1-\mu}{\sigma})} + \sum_{b=2}^{B-1}\frac{(\phi(\frac{a_b-\mu}{\sigma})-\phi(\frac{a_{b-1}-\mu}{\sigma}))^2}{\Phi(\frac{a_b-\mu}{\sigma})-\Phi(\frac{a_{b-1}-\mu}{\sigma})} \\ &\approx \frac{2\phi^2(\frac{a_1-\mu}{\sigma})}{\Phi(\frac{a_1-\mu}{\sigma})} + \sum_{b=2}^{B-1}\Big(\frac{a_{b-1}-\mu}{\sigma}\Big)^2\phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big)\frac{h}{\sigma^2}. \end{aligned}$$

It is straightforward to prove that

$$\lim_{a_1,R\to\infty}\frac{2\phi^2(\frac{a_1-\mu}{\sigma})}{\Phi(\frac{a_1-\mu}{\sigma})} + \sum_{b=2}^{B-1}\Big(\frac{a_{b-1}-\mu}{\sigma}\Big)^2\phi\Big(\frac{a_{b-1}-\mu}{\sigma}\Big)\frac{h}{\sigma^2} = \frac{1}{\sigma^2}\int_{-\infty}^{+\infty}\frac{(x-\mu)^2}{\sigma}\phi(\frac{x-\mu}{\sigma})dx = 1.$$

This completes the proof. □

2.1.3 Grid selection

A further point of interest is the selection of an optimal grid. In this section, we consider as "optimal" that grid whose estimator have minimal variance among the class of equispaced and symmetric grids. In Proposition 2.1.2, we have seen that an infinitely wide and fine grid produces estimators whose variance approaches the MLE variance, which is known to be optimal. Once fixed the refinement R, it is interesting to know how much

wide the optimal grid must be, i.e., which is the optimal value of a_1. Thanks to a numerical computation, we have seen that a_1 may decrease at least at logarithmic rate with regards to R. Figure 2.1 illustrates the succession of optimal a_1 in the case where $\mu = 0$, showing this logarithmic behaviour. This experimental result motivates the following (experimental) conjecture for which a theoretical proof is still required.

Conjecture 2.1.3. *The sequence* $a_1^{(R)} = \max_{a_1<\mu} Var(\widehat{\mu}^{MLE})/Var(\widehat{\mu}^b_{R,a_1})$ *is bounded below by the sequence* $a^{(R)} = -2\log R + \mu$.

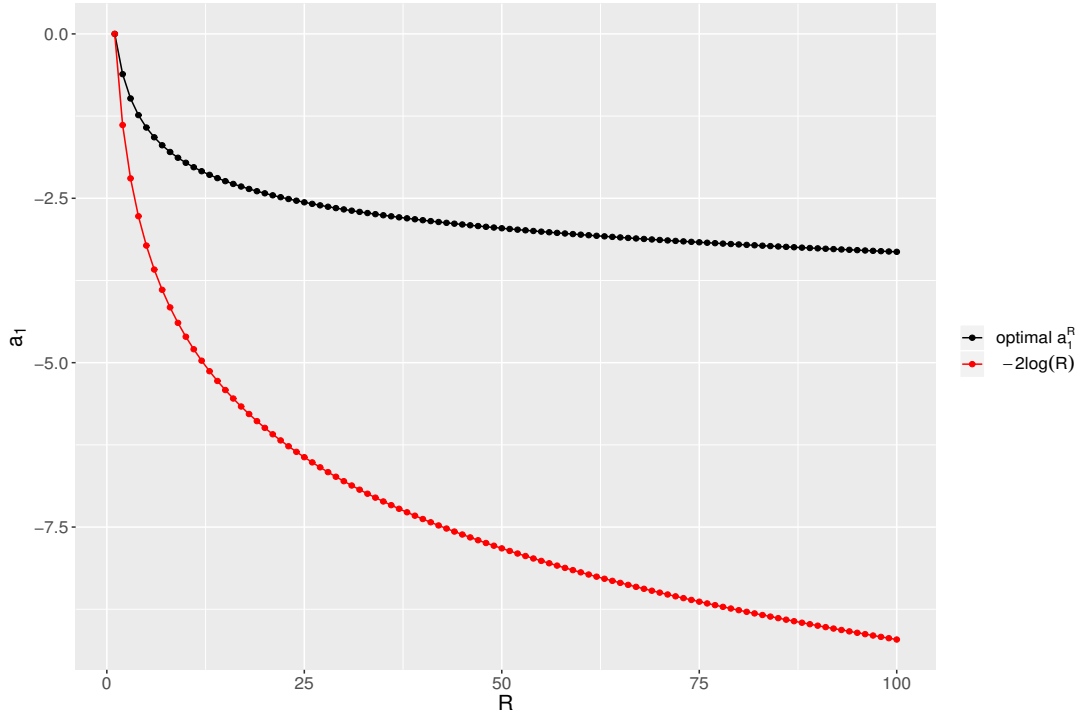


Figure 2.1: Lower bound for the sequence $a_1^{(R)}$ when $\mu = 0$.

The previous conjecture would be useful to foresee how much wide an optimal grid would be, if its refinement R is known. In the following, we propose a criterion to select an optimal grid among all equispaced grids $G(a_1, R)$ symmetric around the point $(\min(\mathbf{x}) + \max(\mathbf{x}))/2$ (which is asymptotically equal to μ). The selected grid is optimal as the corresponding provided estimator $\widehat{\mu}^b_{a_1,R}$ has minimum variance. This criterion, named GVC (*Grid variance criterion*), consists in maximizing, w.r.t. a_1 and R, the quantity:

$$\text{GVC} = \sum_{i=0}^{R} \frac{(\phi(a_i, \widehat{\mu}^b_{R,a_1}, 1) - \phi(a_{i-1}, \widehat{\mu}^b_{R,a_1}, 1))^2}{\Phi(a_i, \widehat{\mu}^b_{R,a_1}, 1) - \Phi(a_{i-1}, \widehat{\mu}^b_{R,a_1}, 1)}.$$

GVC reveals to be consistent, as the following proposition states.

Proposition 2.1.4. *GVC criterion is consistent, i.e., the probability of selecting the grid $G(a_1, R)$ providing the estimator with minimum variance tends to 1 when $n \to \infty$.*

Before proceeding to the proof of this proposition, we have to demonstrate two lemmas.

Lemma 2.1.5. *If a_n and b_n are two non-negative successions and $a_n + b_n \to 0$, then $a_n \to 0$ and $b_n \to 0$.*

Proof. The limit $a_n + b_n \to 0$ is equivalent to write

$$\forall \epsilon > 0 \ \exists \nu > 0 : \ \forall n > \nu \ |a_n + b_n| < \epsilon$$

and, using the positiveness of a_n and b_n,

$$\forall \epsilon > 0 \ \exists \nu > 0 : \ \forall n > \nu \ a_n + b_n < \epsilon$$

We can prove that $a_n \to 0$. In fact having fixed $\epsilon > 0$, it exists a $\nu > 0$ so that $|a_n| = a_n \leq a_n + b_n < \epsilon$ for all $n > \nu$. Similarly we can prove that $b_n \to 0$. □

Lemma 2.1.6. *If $X_n \xrightarrow{p} c$ and $Y_n \xrightarrow{p} d$ with $c < d$ then $P(X_n - Y_n < 0) \to 1$.*

Proof. It is well-known that if $X_n \xrightarrow{p} c$ and $Y_n \xrightarrow{p} d$ then $X_n - Y_n \xrightarrow{p} c - d = e < 0$. It is equivalent to write

$$\forall \epsilon > 0 \ \lim_{n \to \infty} P(|X_n - Y_n - e| > \epsilon) = 0.$$

Fixing $\epsilon = -e$ we have that

$$\lim_{n \to \infty} P(|X_n - Y_n - e| > -e) = \lim_{n \to \infty} P(X_n - Y_n > 0) + P(X_n - Y_n < -2e) = 0.$$

As both the successions $P(X_n - Y_n > 0)$ and $P(X_n - Y_n < -2e)$ are non-negative, we can use the previous lemma to demonstrate that $P(X_n - Y_n > 0) \to 0$, which is equivalent to the thesis. □

We are now ready for the proof of Proposition 2.1.4.

Proof of Proposition 2.1.4. Let $G' = G(a_1', R')$ and $G'' = G(a_1'', R'')$ be two grids on the same data. We define an oracle criterion GVCO, consisting in mazimizing w.r.t. a_1, R:

$$\text{GVCO} = \sum_{i=0}^{R} \frac{(\phi(a_i, \mu, 1) - \phi(a_{i-1}, \mu, 1))^2}{\Phi(a_i, \mu, 1) - \Phi(a_{i-1}, \mu, 1)},$$

where μ is the true mean. We suppose that the grid G' is better than G'' according to the oracle criterion $GVCO$, i.e. $GVCO(G'') < GVCO(G')$. What we need to demonstrate is that $\lim_{n \to \infty} P(GVC(G'') - GVC(G') < 0) = 1$.

We know that the $\widehat{\mu}^b_{R,a_1} \xrightarrow{p} \mu$ and $\widehat{\mu}^b_{R,a_1'} \xrightarrow{p} \mu$, from properties of maximum likelihood estimator. Consequently $GVC(G') \xrightarrow{p} GVCO(G')$ and $GVC(G'') \xrightarrow{p} GVCO(G'')$, because our criteria are based on the same continuous function. So, the hypotheses of the previous lemma are fulfilled, and the claim follows immediately from this result. □

2.2 Univariate Gaussian mixtures

In this section, we focus on univariate Gaussian mixture models with K classes. We show how the usage of binned data requires a particular version of the EM algorithm to be estimated (McLachlan and Jones, 1988). This procedure reveals to be scalable and frugal, guaranteeing a good compromise between clustering quality and time and memory consumption. This fact is shown experimentally, making comparisons with the original raw algorithm, which will be overtaken especially with $R \ll n$. Previously, we assess the generic identifiability of univariate Gaussian mixtures in presence of binned data. We can specialize the notation presented in Section 1.2, assuming for that the observations x_i are generated according to the density:

$$f(x; \boldsymbol{\psi}) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \sigma_k^2), \;\; \pi_k > 0, \;\; \sum_{k=1}^{K} \pi_k = 1,$$

in which μ_k denotes the mean of the k-th component, σ_k^2 is its variance and $\boldsymbol{\theta}$ is the vector that contains all the parameters, thus $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2)$. Moreover, as the observations have real values like in the previous cases, we can adopt the same notation for the grids considered. If we build binned data $\boldsymbol{n}$ with a grid of refinement degree equal to R, then $\boldsymbol{n}$ will have length $B = R + 1$ and density:

$$p(\boldsymbol{n}; \boldsymbol{\psi}) \propto \prod_{b=1}^{B} \Big(\int_{a_{b-1}}^{a_b} \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \sigma_k^2) dx \Big)^{n_b}. \tag{2.3}$$

2.2.1 Generic identifiability of univariate binned Gaussian mixtures

In Section 2.1.1 we have defined the notion of identifiability and prove it for binned Gaussian models. Gaussian mixtures with K components are known to be generically identifiable (Yakowitz and Spragins, 1968), which means that strict identifiability as defined in Definition 2.1.1 holds only almost everywhere in $\boldsymbol{\Psi}$. Here is a formal definition of that:

Definition 2.2.1. *A parametric model $\mathcal{M} = \{f(\cdot; \boldsymbol{\psi}) : \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$ is said to be generically identifiable if it exists a null measure set $\boldsymbol{\Psi}'$ such that*

$$\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi} \setminus \boldsymbol{\Psi}' \quad f(\cdot; \boldsymbol{\psi}) = f(\cdot; \boldsymbol{\psi}') \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}'. \tag{2.4}$$

Mixture models with K components are only generically identifiable because identifiability holds up to labels permutation. Now, our aim is to assess the validity of the same property for binned mixture models. As in Section 2.1.1, we specialize the definition of generic identifiability in presence of binned data.

Definition 2.2.2. *In presence of binned data, a parametric model* $\mathcal{P} = \{p(\cdot;\boldsymbol{\psi}) : \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$ *is said to be generically identifiable if it exists a null measure set* $\boldsymbol{\Psi}'$ *such that*

$$\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi} \setminus \boldsymbol{\Psi}' \;\; p(\boldsymbol{n};\boldsymbol{\psi}) = p(\boldsymbol{n};\boldsymbol{\psi}') \;\; \forall G, \boldsymbol{n} \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}'. \tag{2.5}$$

Typically, identifiabity is considered to be a prerequisite for a good estimation. However this, to the best of our knowledge, there is no reference to Gaussian mixtures identifiability with binned data, neither in the seminal works of McLachlan and Jones (1988) and Cadez et al. (2002), which pass directly to the estimation phase. In this section, we cover partially this lack, giving some conditions on the grid assuring identifiability in the univariate case. This analysis continues in the next chapters, where multivariate models will be debated.

In the univariate setting, we are able to define a sufficient condition that assures generic identifiability. This is a consequence of the following proposition which is contained in Valiant (2012).

Proposition 2.2.1 (**Proposition 11.5** in Valiant (2012))**.** *Given the linear combination of* K *univariate Gaussian densities* $f(x) = \sum_{k=1}^{K} \pi_k \phi(x;\mu_k,\sigma_k^2)$*, such that either* $\mu_{k_1} \neq \mu_{k_2}$ *or* $\sigma_{k_1}^2 \neq \sigma_{k_2}^2$ *for* $k_1 \neq k_2$ *and for all* k $\pi_k \in \mathbb{R}^*$*, the number of solutions to* $f(x) = 0$ *is at most* $2(K-1)$*.*

We are ready to enounce and prove our proposition for the generic identifiability of univariate binned Gaussian mixtures.

Proposition 2.2.2. *Binned univariate mixtures of* K *Gaussian distributions are identifiable if the binning grid has* $R > 4K - 3$ *cut points.*

Proof. If $\mathcal{X} = \mathbb{R}$, the considered probability mass functions reduces to $p(\boldsymbol{n},\boldsymbol{\psi})$, thus it is needed to demonstrate that statement

$$\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi} : \;\; p(\boldsymbol{n};\boldsymbol{\psi}) = p(\boldsymbol{n};\boldsymbol{\psi}') \, \forall G, \boldsymbol{n} \;\Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}' \tag{2.6}$$

hold almost everywhere (up to label permutation) except for a set whose Lebesgue's measure is zero, respectively to the dimension of the original space.

Denoting with $\Phi(\cdot)$ the cumulative density function of a standard Gaussian, if G has R cut points $(a_1,\ldots,a_R)$ then it is sufficient to prove that the system

$$\begin{cases} \sum_{k=1}^{K} \pi_k \Phi(\frac{a_1-\mu_k}{\sigma_k}) = \sum_{k=1}^{K} \pi'_k \Phi(\frac{a_1-\mu'_k}{\sigma'_k}) \\ \sum_{k=1}^{K} \pi_k \Phi(\frac{a_2-\mu_k}{\sigma_k}) = \sum_{k=1}^{K'} \pi'_k \Phi(\frac{a_2-\mu'_k}{\sigma'_k}) \\ \vdots \\ \sum_{k=1}^{K} \pi_k \Phi(\frac{a_R-\mu_k}{\sigma_k}) = \sum_{k=1}^{K} \pi'_k \Phi(\frac{a_R-\mu'_k}{\sigma'_k}) \end{cases}$$

has only the trivial solution $\boldsymbol{\psi} = \boldsymbol{\psi}'$ whatever the grid is. Hence, the non-zero subset of non identifiability is the one of the possible permutation of $\boldsymbol{\psi}$.

It is also equivalent to discover how many zeros can have the difference between the cumulative density functions of two different Gaussian mixtures. If this number is a certain Z, identifiability is assured for $R > Z$.

Again, considering the difference between two cumulative functions with Z zeros, namely $h(x)$, for continuity and for the fact that $\lim_{x\to-\infty} h(x) = \lim_{x\to+\infty} h(x) = 0$, it is necessary that this function has at least $Z + 1$ critical points, i.e. the difference of the two respective density functions has at least $Z + 1$ zeros. So it is possible to formulate the problem in the terms of maximum number of zeros of the difference between the densities of two different mixtures.

Valiant's theorem states that this maximum number is $4K - 2$. Thus, if $R > 4K - 3$, identifiability holds. □

2.2.2 Binned EM algorithm for univariate mixture models

The usage of binned data changes model definition and thus its estimation. We have already presented the EM algorithm used to estimate a general multivariate mixture models in presence of binned data (Algorithm 1). We can therefore specify the previous algorithm for a univariate mixture adopting our notation. As before, the couple composed by raw data $\mathbf{x}$ and labels $\mathbf{z}$ is considered as hidden information sources and, thus, used the EM machinery on the complete log-likelihood

$$\ell^c(\boldsymbol{\psi}; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log(\pi_k \phi(x_i; \mu_k, \sigma_k^2)).$$

In the binned data case, once fixed an initial guess $\boldsymbol{\psi}^{(0)}$, the quantity $Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(0)}) = \mathbb{E}_{\boldsymbol{\psi}^{(0)}}[\ell^c(\boldsymbol{\psi}; \mathbf{X}, \mathbf{Z})|\mathbf{n}]$, calculated with respect to the conditional density $p(\mathbf{x}, \mathbf{z}|\mathbf{n}; \boldsymbol{\psi}^{(0)})$, is maximized. This maximization is repeated at each iteration $j \geq 0$, until the algorithm is stopped due to the convergence of a chosen criterion. Algorithm 3 briefly shows the complete procedure associated to the binned EM algorithm for univariate Gaussian mixtures.

2.2.3 Experimental analysis: binned data in action

In order to motivate our proposed binned strategy, we furnish a numerical simulation illustrating the gain that could be expected in comparison to the classical subsampling strategy. As seen in Section 1.3.1, subsampling is usually used for reducing the data size and, thus, alleviate computational cost. In this simulation a sample of $n = 10^6$ raw data i.i.d. arises from a univariate Gaussian mixture of three components, with true density

$$f(x; \boldsymbol{\psi}^*) = 0.6\phi(x; -1, 2) + 0.3\phi(x; 1, 1) + 0.1\phi(x; 0, 0.5).$$

Algorithm 3 Bin-EM algorithm for univariate Gaussian mixtures models

1. **Initialization phase**: provide an initial guess $\boldsymbol{\psi}^{(0)}$ and a threshold $\epsilon > 0$
2. For $j \geq 0$:
   - **E Step:** Given the estimate $\boldsymbol{\psi}^{(j)}$, calculate $Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)}) = \mathbb{E}_{\boldsymbol{\psi}^{(j)}}[\ell^c(\boldsymbol{\psi}; \mathbf{X}, \mathbf{Z})|\boldsymbol{n}]$;
   - **M Step:** Obtain the new estimate $\boldsymbol{\psi}^{(j+1)} = \text{argmax}_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} Q(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$. This maximization leads to:

     For $b = 1, \ldots, B$

$$g_b(x) = \frac{f(x, \boldsymbol{\psi}^{(j)})}{\int_{a_{b-1}}^{a_b} f(x, \boldsymbol{\psi}^{(j)})dx}$$

     For $k = 1, \ldots, K$

$$\tau_k^{(j)}(x) = \frac{\pi_k \phi(x, \mu_k^{(j)}, \sigma_k^{2(j)})}{f(x, \boldsymbol{\psi}^{(j)})}$$

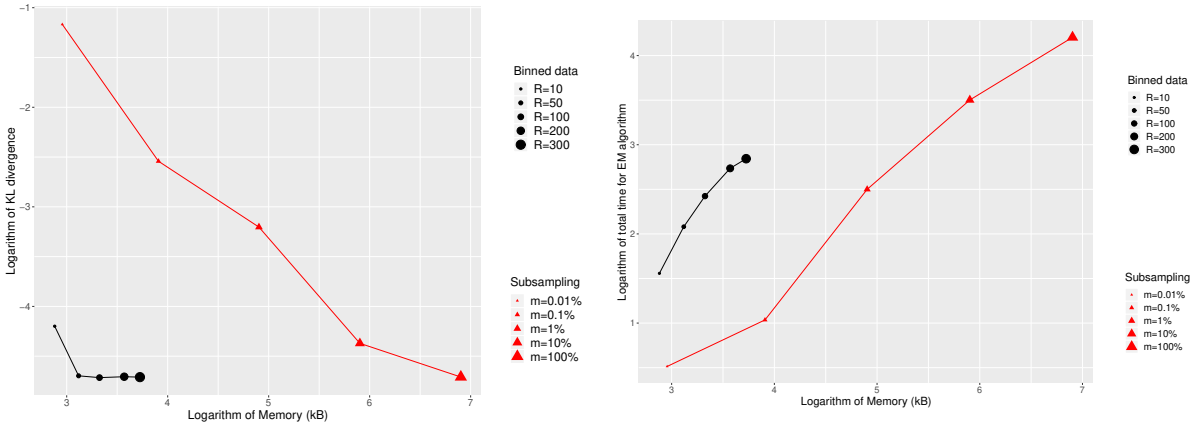
$$\pi_k^{(j+1)} = \frac{\sum_{b=1}^{B} n_b \int_{a_{b-1}}^{a_b} \tau_k^{(j)}(x) g_b(x) dx}{n}$$

$$\mu_k^{(j+1)} = \frac{\sum_{b=1}^{B} n_b \int_{a_{b-1}}^{a_b} x\tau_k^{(j)}(x) g_b(x) dx}{\sum_{b=1}^{B} n_b \int_{a_{b-1}}^{a_b} \tau_k^{(j)}(x) g_b(x) dx}$$

$$\sigma_k^{2(j+1)} = \frac{\sum_{b=1}^{B} n_b \int_{a_{b-1}}^{a_b} (x - \mu_k^{(j+1)})^2 \tau_k^{(j)}(x) g_b(x) dx}{\sum_{b=1}^{B} n_b \int_{a_{b-1}}^{a_b} \tau_k^{(j)}(x) g_b(x) dx}$$

   - **Stopping rule**: Stop if $\left|\frac{\ell(\boldsymbol{\psi}^{(j+1)};\boldsymbol{n}) - \ell(\boldsymbol{\psi}^{(j)};\boldsymbol{n})}{\ell(\boldsymbol{\psi}^{(j)};\boldsymbol{n})}\right| < \epsilon$ is verified, continue otherwise.

As specified before, binned data are created through a grid with refinement parameter R. We considered different values of R (thus different candidate binned data sets) and different values of m (thus different candidate subsampled data sets) both to compare binning and subsampling strategies and also to observe the influence of the grid refinement on the estimation procedure. For each value of R and m we performed EM algorithm (binned and subsampled version, respectively). Then, we measured time and memory requested by each algorithm execution. In addition, to quantify the loss of information induced by binning or subsampling, we calculate the Kullback-Leibler divergence (Kullback and



(a) (b)

Figure 2.2: (a) Logarithm of Kullback-Leibler divergence from the true parameters for different values of R and m in function of the required computer memory (logarithmic scale); (b) Logarithm of total time requested by EM algorithm for different values of R (binned version) and m (subsamplig version) in function of the required computer memory (logarithmic scale).

Leibler, 1951) between each estimated density and $f(\cdot; \boldsymbol{\psi}^*)$. Figures 2.2a-2.2b confirm that binning is more convenient than subsampling. Indeed, it is possible to note that the loss of information (measured by the Kullback-Leibler divergence) induced by binning is much lower than that obtained with subsampling, even negligible if we use a grid moderately dense. This is in addition accompanied by an evident gain in terms of computer memory and computational time.

2.3 Conclusion

In this chapter we have formalized our idea of data-reduction based on an building artificially binned data, where a key-role is played by a regular binning grid. We have seen that binned MLE for univariate Gaussian mixtures preserves the good properties of its raw counterpart and that the related EM algorithm is frugal and more efficient than subsampling.

These preliminary results are very promising for an extensive application of binned data in D-variate context, with $D > 1$, which are presented in the next chapter. This is the main contribution of the thesis and it allows to perform model-based clustering frugally on huge imbalanced multivariate data sets. As it consists in a marginal use of binned data, we have named our proposed technique *bin-marginal*.

Chapter 3

Frugal multivariate Gaussian mixture models with binned data: bin-marginal approach

In this chapter we describe the main contribution of the thesis: a bin-marginal method to frugally cluster huge and imbalanced D-variate data using Gaussian mixture models. This approach is motivated by the fact that a specific version of the curse of dimensionality (Bellman, 1961) affects the naive multivariate extension of the binned methodology described in Chapter 2, as shown in Section 3.1. In order to cope with this difficulty, in Section 3.2.1 we propose further reducing our initial data set, working with *marginal counts* or bin-marginal data, i.e., a collection of univariate binned data generated by each separate variable of the raw data set. This further data-reduction given by marginalization involves the definition of a new bin-marginal model, whose identifiability is discussed in Section 3.2.2, after a preliminary discussion on the identifiability of multivariate binned Gaussian mixtures. In Section 3.2.3, we formulate the EM algorithm to optimize the related bin-marginal likelihood, but this procedure turns out to be computationally unfeasible. Therefore, in Section 3.3 we define a *composite likelihood* (Lindsay, 1988) approach to estimate the bin-marginal model, providing a feasible EM-like algorithm which maximizes the *bin-marginal composite likelihood*. This final step finally defines our frugal Gaussian clustering proposal based on the bin-marginal approach. Previously, in Section 3.3.3, we give some theoretical remarks on the bin-marginal composite likelihood. The method is developed under the hypothesis of diagonal covariance matrices, due to the theoretical impossibility of estimating covariance parameters. This diagonal restriction is in fact common in literature, as it is employed in some popular clustering methods, as K-means (MacQueen et al., 1967), or in the so-called parsimonious Gaussian mixture models (McNicholas and Murphy, 2008). Then, in Section 3.4, the proposed method is tested on several numerical simulations involving imbalanced and huge data sets, comparing it to the subsampling strategy and to the full data set analysis. In particular our method and the subsampling are compared under identical computational constraints, assuring they will use the same amount of computer memory. Full data result is used as a benchmark,

even if it is far from being competitively frugal. The same settings are maintained in Section 3.5, where we test the proposed technique on real data sets coming from various domains of application, such as image segmentation, hazardous asteroids detection and frauds recognition.

3.1 Curse of dimensionality for binned data

In Section 1.6.1, we presented multivariate binned Gaussian mixture models and the EM algorithm to estimate them through the maximization of the log-likelihood

$$\ell(\boldsymbol{\psi}; \boldsymbol{n}) = \sum_{b=1}^{B} n_b \log\Big(\sum_{k=1}^{K} \pi_k \int_{\mathcal{B}_b} \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) d\boldsymbol{x}\Big),$$

where all quantities have already been defined in the same section.

The use of multivariate binned data can be seen as the natural extension of the univariate binned methodology developed in the previous chapter. We have shown that this strategy works well if $B = R + 1 \ll n$, where R is the refinement of the only grid considered.

The same technique could be adopted in this multivariate setting, but we have to point out the arising of some issues when D increases. Indeed, as the number of non-empty bins depends exponentially on the dimension D (Figure 3.1), the amount of binned data does not allow to stick to our frugal memory constraints. Thus, in the D-dimensional context, a classical approach with binned data vanishes any kind of gain.

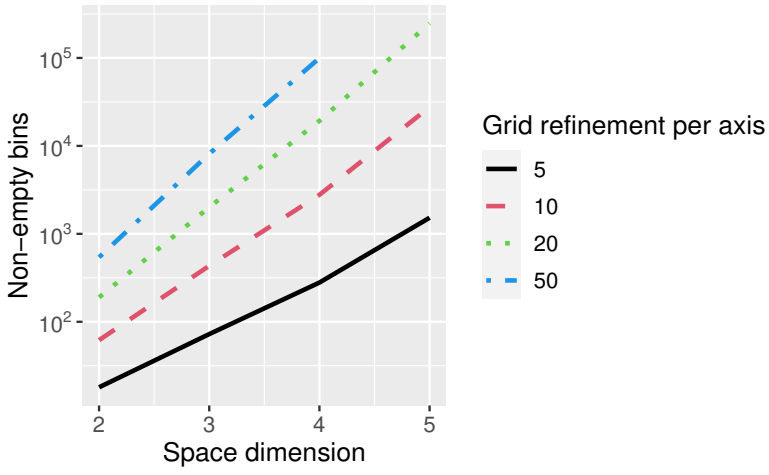


Figure 3.1: Number of non-empty bins depending on both space dimension D and grid refinement (per axis) generated by a single D-variate standard Gaussian.

In order to avoid this particular version of *curse of dimensionality* for binned data, non-trivial multivariate extensions of the binned methodology of Chapter 2 are demanded.

In the following sections, we present our solution, based on a marginal use of binned data and on the related *bin-marginal* model.

3.2 Bin-marginal model

3.2.1 Compressed binned data: bin-marginal solution

In the previous section we pointed out the storage issues linked to a classical use of binned data. Our first idea consists in using what we call *marginal counts*, that are the collection of binned data obtained on each dimension *separately*. In the present section we illustrate a full likelihood estimation of the model generating marginal counts, highlighting its complexity, which motivates completely our final proposal in Section 3.3 based on an alternative composite likelihood approach.

Let define $\boldsymbol{m} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_D\}$, where $\boldsymbol{m}_d$ is the binned data vector referring to the projection on the axis d of the observations $\boldsymbol{x}_i$ after imposing the grid G_d, which produces $B_d = R_d + 1$ bins. It means that, for each $d = 1, \ldots, D$, $\boldsymbol{m}_d = (m_{d1}, \ldots, m_{dB_d})$, where each component is defined as $m_{db_d} = \#\{x_{id} : a_{d(b_d-1)} \leq x_{id} < a_{db_d}\}$ and x_{id} is the d-th component of $\boldsymbol{x}_i$. Thus, the collection $\boldsymbol{m}$ contains the *marginal counts* of $\boldsymbol{n}$. To facilitate the comprehension of the specific data compression mechanism and its related notation, a simple bivariate situation is depicted in Figure 3.2. Here, a 3×3 grid overlaps 20 raw individuals $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{20})$ and both the bivariate binned data $\boldsymbol{n}$ and marginal counts $\boldsymbol{m}$ are highlighted.

The introduction of marginal counts makes resource savings possible: in fact, it is clear that storing them instead of the full grid is convenient for computer memory, as we have to save at most $\sum_{d=1}^{D} B_d$ elements instead of $\prod_{d=1}^{D} B_d$ ones. So, a first attempt could be the estimation of the *bin-marginal* model whose probability mass function is:

$$p_m(\boldsymbol{m}; \boldsymbol{\psi}) = \sum_{\boldsymbol{n}' \in \mathcal{F}_{\boldsymbol{m}}} p(\boldsymbol{n}'; \boldsymbol{\psi}), \tag{3.1}$$

where $\mathcal{F}_{\boldsymbol{m}}$ is the set of tables $\boldsymbol{n}'$ sharing the same marginals $\boldsymbol{m}$. Formally:

$$\mathcal{F}_{\boldsymbol{m}} = \{\boldsymbol{n}' : \boldsymbol{m}' = \boldsymbol{m}\},$$

where $\boldsymbol{m}'$ are the marginal counts of each table $\boldsymbol{n}'$.

But now we need to assess three important issues before proposing this model as a useful frugal method:

- *Identifiability of the model.* We wonder if different parameters index different bin-marginal probability mass functions. This question will be treated in Section 3.2.2.

- *Mathematical complexity of the log-likelihood* $\ell_m(\boldsymbol{\psi}; \boldsymbol{m}) = \log p_m(\boldsymbol{m}; \boldsymbol{\psi})$. From (3.1) we note that the computation of this log-likelihood is intractable, because we need to calculate a considerable number of complete tables. Section 3.3 will be dedicated to overcome this specific issue.
- *Optimization of the likelihood.* In Section 3.2.3 we give a version of the EM algorithm to do this task. We will show it does not solve all the issues appeared in 3.1 and, again, Section 3.3 will propose a specific solution.

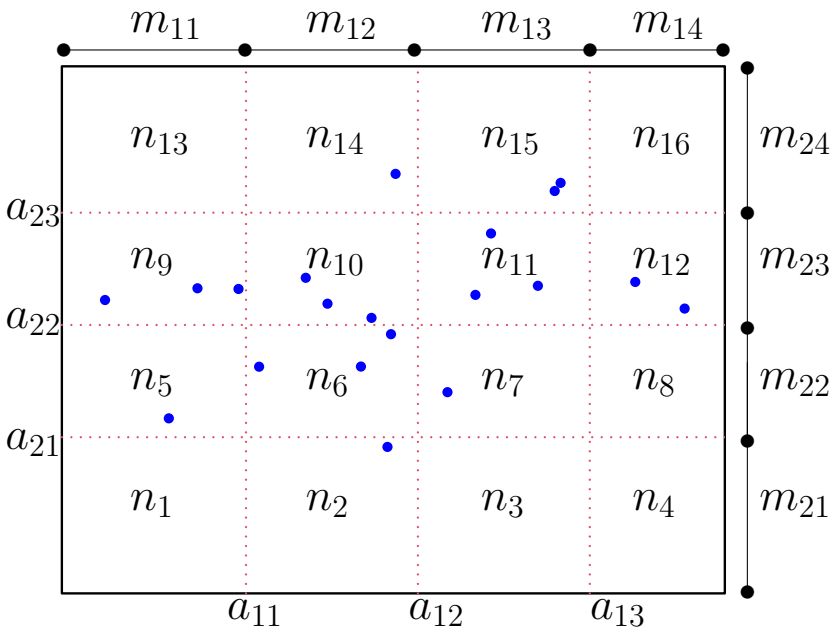


Figure 3.2: Bivariate representation of a 3×3 grid (red dotted lines) superposing on 20 points $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{20})$ (in blue). Bivariate binned data are $\boldsymbol{n} = (n_1, \ldots, n_{16})$, while marginal counts are $\boldsymbol{m} = \{\boldsymbol{m}_1, \boldsymbol{m}_2\}$, where $\boldsymbol{m}_1 = (m_{11}, \ldots, m_{14})$ and $\boldsymbol{m}_2 = (m_{21}, \ldots, m_{24})$.

3.2.2 Requirements for identifiability

Typically, before proceeding with the estimation of any statistical model $\mathcal{P} = \{p(\boldsymbol{x}; \boldsymbol{\psi}), \boldsymbol{x} \in \mathcal{X}, \boldsymbol{\psi} \in \boldsymbol{\Psi}\}$, statisticians are interested in knowing if it is *identifiable*, i.e. if any different

value of the model parameter $\boldsymbol{\psi}$ indexes different elements in $\mathcal{P}$. In case of continuous model, these elements are densities, while they are probability mass functions if the model is discrete, as in our binned data case. In this section, we discuss the identifiability of bin-marginal Gaussian mixtures models, knowing that Gaussian mixtures with raw data are identifiable up to a labelling permutation (Yakowitz and Spragins, 1968). As pointed out in Section 2.2.1, there is no reference to Gaussian mixtures identifiability with binned data in the multivariate case, to the best of our knowledge. Our investigation in the bin-marginal case also allows us to cover partially this lack, as our result on the bin-marginal case is based on a preliminary statement about full binned identifiability. These two results provide sufficient conditions regarding the binning grids and the parameter space, under hypothesis of diagonal covariance matrices (in the following, $\boldsymbol{\Psi}$ is the space containing only diagonal Gaussian mixtures). This apparent restriction does not affect our proposal, because this assumption is common in several clustering approaches, even for the raw data case, as K-means (MacQueen et al., 1967) and parsimonious Gaussian mixture models (Celeux and Govaert, 1995), and because, in Section 3.3, our proposal will be presented under these conditions.

Preliminary result: identifiability of binned Gaussian diagonal mixtures Sufficient conditions for the identifiability of diagonal D-variate binned mixture models are provided by the following proposition. These conditions regard the refinement degree of the binning grid and the mixture components, that can not share the projection on the same axis or having the same proportion. Thus, the parametric space is restricted to $\boldsymbol{\Psi}/\boldsymbol{\Psi}^{\dagger}$. The set $\boldsymbol{\Psi}^{\dagger}$ is defined as $\boldsymbol{\Psi}^{\dagger} = (\boldsymbol{\Pi}_K^{\ddagger} \times \mathbb{R}^{DK} \times \mathbb{R}^{+DK}) \cup \boldsymbol{\Psi}^{\ddagger}$, where

$$\begin{aligned}\boldsymbol{\Psi}^{\ddagger} &= \{\boldsymbol{\psi} \in \boldsymbol{\Psi} : \exists k, k', d : \mu_{kd} = \mu_{k'd}, \sigma_{kd}^2 = \sigma_{k'd}^2\}\\ \boldsymbol{\Pi}_K^{\ddagger} &= \{\boldsymbol{\pi} \in \boldsymbol{\Pi}_K : \exists i, j : \pi_i = \pi_j\}.\end{aligned}$$

As $\boldsymbol{\Psi}^{\dagger}$ is a null-measure set, we can also say that binned Gaussian diagonal mixtures are *generically identifiable* (Allman et al., 2009).

Proposition 3.2.1 (**Binned Gaussian diagonal mixtures**)**.** *Under hypothesis of diagonal covariance matrices, binned D-variate mixtures of K components are identifiable if $R_d > 4K - 3$, $d = 1, \ldots, D$ and $\boldsymbol{\psi} \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^{\dagger}$, up to label permutations.*

Proof. The statement to prove is:

$$\begin{aligned}\forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^{\dagger} : \;& p(\boldsymbol{n}; \boldsymbol{\psi}) = p(\boldsymbol{n}; \boldsymbol{\psi}') \;\; \forall G, \boldsymbol{n} \\ & \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}' \;\; \text{(up to label permutation)}.\end{aligned} \tag{3.2}$$

We have to prove statement (3.2) for a binned mixture of dimension D. Considering a grid with R_d cut points on each dimension $d = 1, \ldots, D$ and $\prod_{d=1}^{D}(R_d+1)$ bins, statement

(3.2) holds if the system

$$\begin{cases} \sum_{k=1}^{K} \pi_k \int_{\mathcal{B}_b} \phi(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) d\boldsymbol{x} \\ = \sum_{k=1}^{K} \pi'_k \int_{\mathcal{B}_b} \phi(\boldsymbol{x}; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k) d\boldsymbol{x} \\ \\ b = 1, \ldots, B \end{cases} \tag{3.3}$$

has only the trivial solutions $\boldsymbol{\psi} = \boldsymbol{\psi}'$, up to a label permutation. Each D-dimensional bin is the Cartesian product of certain 1-dimensional bins, so every one-dimensional projection of a D-dimension bin $\mathcal{B}_b$, namely $\mathcal{B}_b^d$, coincides with a certain $\mathcal{B}_{b_d}^d$, which is a bin on the d-th dimension. Thus, under hypothesis of diagonal covariance matrices, the system (3.3) can be rewritten as:

$$\begin{cases} \sum_{k=1}^{K} \pi_k \int_{\mathcal{B}_{b_1}^1} \phi(x_1; \mu_{k1}, \sigma_{k1}^2) dx_1 \\ \times \ldots \times \int_{\mathcal{B}_{b_D}^D} \phi(x_D; \mu_{kD}, \sigma_{kD}^2) dx_D \\ = \sum_{k=1}^{K} \pi'_k \int_{\mathcal{B}_{b_1}^1} \phi(x_1; \mu'_{k1}, \sigma_{k1}^{2\prime}) dx_1 \\ \times \ldots \times \int_{\mathcal{B}_{b_D}^D} \phi(x_D; \mu'_{kD}, \sigma_{kD}^{2\prime}) dx_D \\ \\ b_d = 1, \ldots, B_d, \quad d = 1, \ldots, D. \end{cases} \tag{3.4}$$

To simplify notation, we define the vector of indices $\boldsymbol{b} = (b_1, \ldots, b_D)$ and the set $\bar{\mathcal{B}}$ containing all $\boldsymbol{b}$. Furthermore, we resume with $p_{\boldsymbol{b}}(\boldsymbol{\psi}) = p_{\boldsymbol{b}}(\boldsymbol{\psi}')$ each equation in the system (3.4), which can be concisely written as

$$\begin{cases} p_{\boldsymbol{b}}(\boldsymbol{\psi}) = p_{\boldsymbol{b}}(\boldsymbol{\psi}') \\ \\ \boldsymbol{b} \in \bar{\mathcal{B}}. \end{cases} \tag{3.5}$$

Let consider all equations involving integrals on the same set $\mathcal{B}_{b_1}^1$ and sum them for every $\mathcal{B}_{b_d}^d, b_d = 1, \ldots, B_d, d = 2, \ldots, D$. Iterate this procedure for $b_1 = 2, \ldots, B_1$ to obtain:

$$\sum_{k=1}^{K} \pi_k \int_{\mathcal{B}_{b_1}^1} \phi(x_1; \mu_{k1}, \sigma_{k1}^2) dx_1 = \sum_{k=1}^{K} \pi'_k \int_{\mathcal{B}_{b_1}^1} \phi(x_1; \mu'_{k1}, \sigma_{k1}^{2\prime}) dx_1.$$

These equations define the system of identifiability for a univariate binned mixture with K components. Therefore, from Proposition 2.2.2, it exists a permutation $\rho_1(\cdot)$, such that, for each $k = 1, \ldots, K$:

$$\pi'_k = \pi_{\rho_1(k)} \;\; \mu'_{k1} = \mu_{\rho_1(k)1} \;\; \sigma^{2\prime}_{k1} = \sigma^2_{\rho_1(k)1}.$$

The permutation $\rho_1(\cdot)$ is also unique, as proportions are different from the fact that $\boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$. We can use the rest of equations in system (3.5) to iterate the same procedure for all the D axes, finding D (unique) permutations $\rho_d(\cdot)$ such that, for each $k = 1, \ldots, K$ and $d = 1, \ldots, D$:

$$\pi'_k = \pi_{\rho_d(k)} \;\; \mu'_{kd} = \mu_{\rho_d(k)d} \;\; \sigma^{2\prime}_{kd} = \sigma^2_{\rho_d(k)d}. \tag{3.6}$$

If all permutations $\rho_d(\cdot)$ are equal, identifiability is achieved. Let assume that two permutations $\rho_{d'}(\cdot)$ and $\rho_{d''}(\cdot)$ are different for $d' \neq d''$. It means that there is at least a value k_1 such that

$$\begin{cases} \rho_{d'}(k_1) = k_2 \\ \rho_{d''}(k_1) = k_3, \end{cases}$$

with $k_2 \neq k_3$.

From (3.6) we have $\pi'_{k_1} = \pi_{k_2}$ and $\pi'_{k_1} = \pi_{k_3}$. Thus, $\pi_{k_2} = \pi_{k_3}$. It means that, if $\rho_{d'}(\cdot)$ and $\rho_{d''}(\cdot)$ are different, at least two of the K proportions are equal. This is absurd, as $\boldsymbol{\psi} \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$, and the two permutations must be the same. This completes the proof. □

Identifiability of bin-marginal Gaussian diagonal mixtures Proposition 3.2.1 is crucial to prove identifiability of bin-marginal Gaussian mixtures themselves. Indeed, Proposition 3.2.2 establishes below that bin-marginal mixtures are identifiable if binned mixtures are identifiable. Thus, under the same conditions as Proposition 3.2.1, bin-marginal Gaussian diagonal mixtures are identifiable in $\boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$. Since $\boldsymbol{\Psi}^\dagger$ is a null-measure set, bin-marginal Gaussian diagonal mixtures are, thus, generically identifiable. This result is of central interest in this work, since we will consider only the bin-marginal data in order to preserve computer memory.

Proposition 3.2.2. *Bin-marginal D-variate mixtures of K components are identifiable if binned D-variate mixtures are identifiable. So, under diagonal covariance matrices hypothesis, identifiability is achieved if $R_d > 4K - 3$, $d = 1, \ldots, D$ and $\boldsymbol{\psi} \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$, up to label permutation.*

Proof. Let consider two probability mass functions $p_m(\boldsymbol{m}; \boldsymbol{\psi})$ and $p_m(\boldsymbol{m}; \boldsymbol{\psi}')$. Our aim is to demonstrate

$$\begin{aligned} \forall \boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger : \;\; & p_m(\boldsymbol{m}; \boldsymbol{\psi}) = p_m(\boldsymbol{m}; \boldsymbol{\psi}') \;\; \forall G, \boldsymbol{m} \\ & \Rightarrow \boldsymbol{\psi} = \boldsymbol{\psi}' \;\; \text{(up to label permutation)}. \end{aligned}$$

We can consider a grid of dimension $R_1 \times \ldots \times R_D$ as defined in Section 3.2.1 and the vectors $\boldsymbol{m}_{\boldsymbol{b}} = (\boldsymbol{m}^1_{b_1}, \ldots, \boldsymbol{m}^D_{b_D})$, where $\boldsymbol{b} = (b_1, \ldots, b_D) \in \prod_{d=1}^D \{1, \ldots, B_d\}$. Each vector

$\boldsymbol{m}^d_{b_d}$ is defined as

$$\boldsymbol{m}^d_{b_d} = \begin{cases} n & \text{for an index } b_d \in \{1, \dots, B_d\} \\ 0 & \text{otherwise.} \end{cases}$$

So each $\boldsymbol{m}^d_{b_d}$ is a vector of counts representing the situation in which observations are concentrated in the b_d-th bin on the d-th dimension. Moreover, for each possible $\boldsymbol{m_b}$ we have:

$$p_m(\boldsymbol{m_b}; \boldsymbol{\psi}) = \sum_{\boldsymbol{n'} \in \mathcal{F}_{\boldsymbol{m_b}}} p(\boldsymbol{n'}; \boldsymbol{\psi}) = k(\boldsymbol{m_b}) P_{\boldsymbol{b}}$$
$$p_m(\boldsymbol{m_b}; \boldsymbol{\psi'}) = \sum_{\boldsymbol{n'} \in \mathcal{F}_{\boldsymbol{m_b}}} p(\boldsymbol{n'}; \boldsymbol{\psi'}) = k(\boldsymbol{m_b}) P'_{\boldsymbol{b}}$$

where $k(\boldsymbol{m_b})$ is a constant and $P_{\boldsymbol{b}}$ (and $P'_{\boldsymbol{b}}$) is the probability for the bin whose marginal bin on the d-th is indexed by the d-th element of $\boldsymbol{b}$. Choosing every possible value for $\boldsymbol{b}$ we obtain the same system of identifiability equation for a multivariate binned mixture model. There are no other equation to satisfy because the other probabilities for other vectors $\boldsymbol{m}$ are combinations of $P_{\boldsymbol{b}}$ (or $P'_{\boldsymbol{b}}$). Thus if multivariate binned mixture models are identifiable the binned marginal-conjoint model is identifiable. Moreover, under the hypothesis of Proposition 3.2.1 diagonal binned conjoint-marginal multivariate mixtures are identifiable. □

Remarks and necessary conditions for identifiability Previous propositions state sufficient conditions on the parametric space guaranteeing identifiability. Actually, we think that binned Gaussian mixtures could be identifiable everywhere in $\boldsymbol{\Psi}$, but this is not the aim of this work. Indeed, this preliminary result is sufficient to assess identifiability in the bin-marginal case, which is the main objective of our analysis. Furthermore, the same parametric restrictions will be considered again in Section 3.3.3 to cope with further issues related to our particular estimation strategy that will be presented in the next sections.

We have already pointed out that these propositions contain only sufficient conditions assuring identifiability. Thus, they are not sharp and they may become too rough if K increases. For this reason it is interesting to discuss necessary conditions for identifiability. Following the same approach as Ranalli and Rocci (2017b), in which similar topics are discussed, a necessary condition is that the number of bins in D-dimensional grid must be equal or greater that the number of parameters of a full binned D-variate diagonal Gaussian mixture with K components, as this model can be viewed as a $\prod_{d=1}^{D}(R_d + 1)$

contingency table. It means that:

$$\prod_{d=1}^{D}(R_d+1)-1 \geq 2DK+K-1. \tag{3.7}$$

3.2.3 EM algorithm

It is possible to formulate a specific EM algorithm in order to maximize the bin-marginal log-likelihood $\ell_m(\boldsymbol{m};\boldsymbol{\psi}) = \log p_m(\boldsymbol{m};\boldsymbol{\psi})$ associated to the bin-marginal data set. Therefore, we introduce the *complete log-likelihood*

$$\ell^c(\boldsymbol{\psi};\mathbf{x},\mathbf{z}) = \sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\log(\pi_k\phi(\boldsymbol{x}_i,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)),$$

where $\mathbf{z}$ is an $n \times K$ matrix whose generic element z_{ik} is equal to 1 if $\boldsymbol{x}_i$ belongs to population k, it is 0 otherwise. Thus, $\mathbf{z}$ contains the hidden class memberships of the raw data $\mathbf{x} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$. More precisely, at each iteration $j \geq 0$, given the current estimate $\boldsymbol{\psi}^{(j)}$, the complete log-likehood is used in the so-called E-step, where the following quantity is calculated

$$Q_m(\boldsymbol{\psi},\boldsymbol{\psi}^{(j)}) = \mathbb{E}_{\boldsymbol{\psi}^{(j)}}[\ell^c(\boldsymbol{\psi};\mathbf{X},\mathbf{Z})|\boldsymbol{m}], \tag{3.8}$$

taking the expectation with respect to $p(\mathbf{x},\mathbf{z}|\boldsymbol{m};\boldsymbol{\psi}^{(j)})$. Note that $\mathbf{X}$ and $\mathbf{Z}$ denote, respectively, the random variables generating $\mathbf{x}$ and $\mathbf{z}$.

Let rewrite (3.8):

$$\begin{aligned} &Q_m(\boldsymbol{\psi},\boldsymbol{\psi}^{(j)}) \\ &= \sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} p(\boldsymbol{n}|\boldsymbol{m};\boldsymbol{\psi}^{(j)})\mathbb{E}_{\boldsymbol{\psi}^{(j)}}[\ell^c(\boldsymbol{\psi};\mathbf{X},\mathbf{Z})|\boldsymbol{n}] \end{aligned}$$

where

$$p(\boldsymbol{n}|\boldsymbol{m};\boldsymbol{\psi}^{(j)}) = \frac{p(\boldsymbol{n};\boldsymbol{\psi}^{(j)})}{\sum_{\boldsymbol{n}'\in\mathcal{F}_{\boldsymbol{m}}} p(\boldsymbol{n}';\boldsymbol{\psi}^{(j)})}\mathbb{1}_{\{\boldsymbol{n}\in\mathcal{F}_m\}}. \tag{3.9}$$

After some calculus, this expression reduces to:

$$\begin{aligned} &Q_m(\boldsymbol{\psi},\boldsymbol{\psi}^{(j)}) \\ &= \sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} p(\boldsymbol{n}|\boldsymbol{m};\boldsymbol{\psi}^{(j)}) \\ &\times \sum_{k=1}^{K}\sum_{b=1}^{B} n_b\mathbb{E}_b[\tau_k^{(j)}(\mathbf{X})\log[\pi_k\phi(\mathbf{X};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)]] \end{aligned}$$

where $\mathbb{E}_b$ refers to the expectation with respect to the density $g_b^{(j)}(\boldsymbol{x}) = \frac{f(\boldsymbol{x};\boldsymbol{\psi}^{(j)})}{\int_{\mathcal{B}_b} f(\boldsymbol{y};\boldsymbol{\psi}^{(j)})d\boldsymbol{y}}$ and $\tau_k^{(j)}(\boldsymbol{x}) = \frac{\pi_k^{(j)}\phi(\boldsymbol{x};\boldsymbol{\mu}_k^{(j)},\boldsymbol{\Sigma}_k^{(j)})}{f(\boldsymbol{x};\boldsymbol{\psi}^{(j)})}$. Before proceeding with the M-step, we introduce the following quantities to simplify the notations:

$$\begin{aligned}
\alpha^{(j)}(\boldsymbol{n}) &= p(\boldsymbol{n}|\boldsymbol{m};\boldsymbol{\psi}^{(j)}) \\
A_{kb}^{(j)} &= \int_{\mathcal{B}_b} \tau_k^{(j)}(\boldsymbol{x}) g_b^{(j)}(\boldsymbol{x}) d\boldsymbol{x} \\
B_{kb}^{(j)} &= \int_{\mathcal{B}_b} \boldsymbol{x}\tau_k^{(j)}(\boldsymbol{x}) g_b^{(j)}(\boldsymbol{x}) d\boldsymbol{x} \\
C_{kb}^{(j)} &= \int_{\mathcal{B}_b} (\boldsymbol{x}-\boldsymbol{\mu}_k)(\boldsymbol{x}-\boldsymbol{\mu}_k)^t \tau_k^{(j)}(\boldsymbol{x}) g_b^{(j)}(\boldsymbol{x}) d\boldsymbol{x}.
\end{aligned}$$

Then, in the M-step we maximize $Q_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$, obtaining the following update formulas for each component $k = 1, \ldots, K$:

$$\begin{aligned}
\pi_k^{(j+1)} &= \frac{1}{n} \sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} \alpha^{(j)}(\boldsymbol{n}) \sum_{b=1}^{B} n_b A_{kb}^{(j)} \\
\boldsymbol{\mu}_k^{(j+1)} &= \frac{\sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} \alpha^{(j)}(\boldsymbol{n}) \sum_{b=1}^{B} n_b B_{kb}^{(j)}}{\sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} \alpha^{(j)}(\boldsymbol{n}) \sum_{b=1}^{B} n_b A_{kb}^{(j)}} \\
\boldsymbol{\Sigma}_k^{(j+1)} &= \frac{\sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} \alpha^{(j)}(\boldsymbol{n}) \sum_{b=1}^{B} n_b C_{kb}^{(j)}}{\sum_{\boldsymbol{n}\in\mathcal{F}_{\boldsymbol{m}}} \alpha^{(j)}(\boldsymbol{n}) \sum_{b=1}^{B} n_b A_{kb}^{(j)}}.
\end{aligned}$$

Unfortunately, both previous E and M steps involve the computation of all "crossed" tables $\mathcal{F}_m$ sharing the same marginals, coming back to a memory issue (and also a time computation one). Therefore, an estimation based on the full likelihood of the bin-marginal model is not numerically tractable under our strong computational constraints. For this very reason we will provide in the following section estimates following a composite likelihood approach, after having given a brief introduction of this concept.

3.3 Estimation strategy

In this section we present the estimation part of our contribution, working with diagonal Gaussian mixtures (i.e., matrices $\boldsymbol{\Sigma}_k$ in (1.1) are diagonal). Before, it is necessary to briefly introduce the *marginal composite likelihood*, on which our estimation proposal is based.

3.3.1 Marginal composite likelihood

Marginal composite likelihood is a pseudo-likelihood used to obtain asymptotically consistent estimates (see Varin et al. (2011) for instance) when the optimization of the full likelihood is too

burdensome. The marginal composite likelihood relies only on univariate marginal likelihoods and it is a special case of *composite likelihood* (Lindsay, 1988), where more general multivariate marginal likelihoods can be taken into account.

Let $\mathbf{x}$ be a D-dimensional sample with n observations $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iD})$, $i = 1, \ldots, n$, generated by a Gaussian diagonal mixture model with parameter $\boldsymbol{\psi} \in \Psi$, as in Section 1.6.1. Denoting with $\mathbf{x}_d = (x_{1d}, \ldots, x_{nd})$ the component d of the whole raw data set, $L_d(\boldsymbol{\psi}_d; \mathbf{x}_d)$ is the likelihood of the univariate Gaussian mixture at dimension d with parameter $\boldsymbol{\psi}_d = (\pi_1, \ldots, \pi_K, \mu_{1d}, \ldots, \mu_{Kd}, \sigma^2_{1d}, \ldots, \sigma^2_{Kd})$. Then, the *marginal composite likelihood* is defined as

$$\tilde{L}(\boldsymbol{\psi}; \mathbf{x}) = \prod_{d=1}^{D} L_d(\boldsymbol{\psi}_d; \mathbf{x}_d).$$

Similarly, the *marginal composite log-likelihood* is $\tilde{\ell}(\boldsymbol{\psi}; \mathbf{x}) = \sum_{d=1}^{D} \ell_d(\boldsymbol{\psi}_d; \mathbf{x}_d)$, with $\ell_d(\psi_d; \mathbf{x}_d) = \log L_d(\boldsymbol{\psi}_d; \mathbf{x}_d)$.

The estimator $\tilde{\boldsymbol{\psi}}$ maximizing $\tilde{L}(\boldsymbol{\psi}; \mathbf{x})$ is named *maximum marginal composite likelihood estimator*. It has proved to be consistent and asymptotically normally distributed under very mild conditions about the regularity of the marginal densities (see Molenberghs and Verbeke (2005) for instance).

3.3.2 Bin-marginal composite likelihood

Having given the necessary notation in the previous paragraphs, we can now complete our proposal, in which we will combine the memory reduction offered by bin-marginal data with the computational advantages of marginal composite likelihood. Actually, more general but less frugal versions of composite likelihood have already been used in the area of mixture models. Indeed, a formalization of EM algorithm with composite likelihood could be seen in Gao and Song (2011). They also established three fundamental properties of the associated so-called CL-EM algorithm: ascent property, convergence to a stationary point and a quantification of its rate of convergence. Whitaker et al. (2020) proved the consistency of maximum composite likelihood estimators when using binned data, knowing that raw maximum composite likelihood ones are consistent (see Molenberghs and Verbeke (2005) and Lindsay (1988), for instance). An application of composite likelihood on binned data appeared in Ranalli and Rocci (2016a), where these ones arose from a discrete data problem. This is quite similar to the technique we are about to describe, but it is different as it uses bivariate grids and it does not build artificially binned data as a solution for scalability, because they were already given in the problem statement.

Assuming a marginal D-dimensional Cartesian grid G as defined in Section 3.2.1 and diagonal covariance matrices, instead of maximizing the too complex bin-marginal log-likelihood $\ell_m(\boldsymbol{\psi}; \boldsymbol{m})$, we aim to maximize the following bin-marginal composite log-likelihood:

$$\begin{aligned} \tilde{\ell}_m(\boldsymbol{\psi}; \boldsymbol{m}) &= \sum_{d=1}^{D} \ell_d(\boldsymbol{\psi}_d; \boldsymbol{m}_d) \\ &= \sum_{d=1}^{D} \sum_{b_d=1}^{B_d} m_{db_d} \log \Big(\int_{\mathcal{B}^d_{b_d}} f_d(x_d; \boldsymbol{\psi}_d) dx_d \Big). \end{aligned} \tag{3.10}$$

Here, $\ell_d(\boldsymbol{\psi}_d; \boldsymbol{m}_d)$ is the binned log-likelihood for a univariate Gaussian mixture with K components of density $f_d(x_d; \boldsymbol{\psi}_d)$ indexed by the parameter $\boldsymbol{\psi}_d$. The expression of (3.10) motivates why we work with diagonal mixtures: it is impossible to estimate any kind of covariance parameter, since none of them appear in $\boldsymbol{\psi}_d$.

3.3.3 Properties of the bin-marginal composite likelihood

In Section 3.2.2 we have provided sufficient conditions assuring identifiability of both full binned model and bin-marginal one. These conditions restricted the parameter space to $\boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$, where $\boldsymbol{\Psi}^\dagger$ contained all the mixtures where at least two components share either the same proportion or the same projection on at least one axis. We conjectured that, especially for the full binned model, identifiability could hold in a set larger than $\boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$ and that these restrictions would have been considered again to cope with specific issues related to our composite likelihood-based estimation strategy. Indeed, in this section, we show an example of mixture in $\boldsymbol{\Psi}^\dagger$ for which the use of the bin-marginal composite likelihood sets new obstacles impeding a good estimation.

A pathological example Denoting with $\mathcal{N}_2(\cdot, \cdot)$ a normal bivariate distribution, let consider these two bivariate two-classes mixtures:

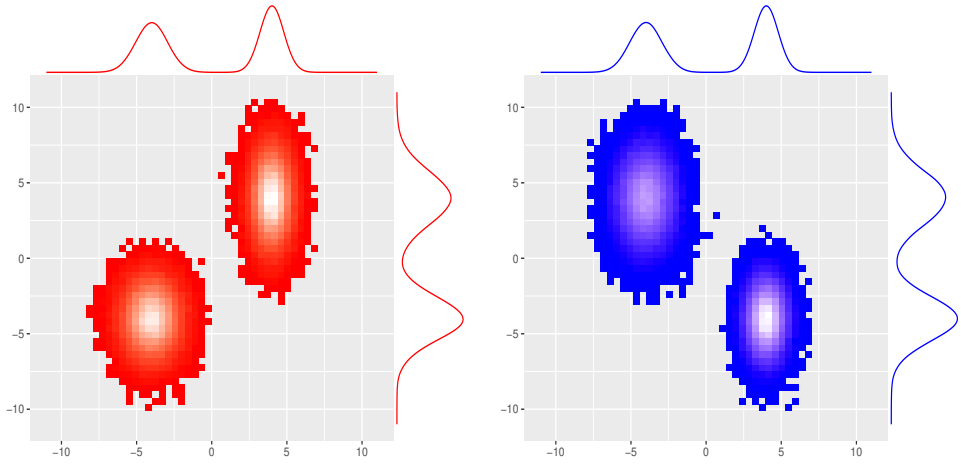
$$0.5\mathcal{N}_2\Big(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}v_1 & 0\\ 0 & v_2\end{pmatrix}\Big) + 0.5\mathcal{N}_2\Big(\begin{pmatrix}\nu_1\\ \nu_2\end{pmatrix}, \begin{pmatrix}w_1 & 0\\ 0 & w_2\end{pmatrix}\Big) \tag{3.11}$$

$$0.5\mathcal{N}_2\Big(\begin{pmatrix}\mu_1\\ \nu_2\end{pmatrix}, \begin{pmatrix}v_1 & 0\\ 0 & w_2\end{pmatrix}\Big) + 0.5\mathcal{N}_2\Big(\begin{pmatrix}\nu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}w_1 & 0\\ 0 & v_2\end{pmatrix}\Big). \tag{3.12}$$

We note that, in both mixtures, proportions are equal, so these two mixtures are in $\boldsymbol{\Psi}^\dagger$. As shown in Figure 3.3, the two mixtures have the same projections on the two axes, although the joint mixtures are different. Therefore, it is not possible to distinguish these two mixtures, knowing only marginal distributions. In this case, thus, maximum bin-marginal composite likelihood estimation is ambiguous.

More specifically, the described example represents a pathological case where two theoretical properties are not matched: the asymptotic identifiability of the optimization criterion and the joint identifiability (i.e., it is not possible to infer the joint D-dimensional mixture, knowing only the D marginal distributions). In the next two paragraphs we show that these two properties are satisfied if $\boldsymbol{\psi} \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$.

Asymptotic identifiability of the optimization criterion The asymptotic identifiability of the optimization criterion for the maximum bin-marginal composite likelihood estimator (i.e., the asymptotic criterion is maximized at the unique value of the true parameter) is a necessary condition to prove its consistency (Wald, 1949; Lindsay, 1988). In this section we prove that this property is fulfilled almost everywhere, except in a null measure set, as the following Proposition 3.3.1 assures. This null measure set of restrictions turns out to be the same $\boldsymbol{\Psi}^\dagger$ defined in Section 3.2.2, which contains constraints regarding projections of components and equality conditions on proportions.



(a) (b)

Figure 3.3: A pathological example where it is impossible to distinguish two bivariate two-classes mixtures from their marginal distributions. Joint distribution and marginal distributions of (a) mixture (3.11) and (b) mixture (3.12).

Proposition 3.3.1. *Assuming the true model is outside the null measure set* $\boldsymbol{\Psi}^\dagger$*, the optimization criterion of the bin-marginal composite log-likelihood, using a grid* $G = G_1 \times \ldots \times G_d$ *with* $\prod_{d=1}^{D} R_d$ *cut points is asymptotically identifiable if* $R_d > 4K - 3$*,* $d = 1, \ldots, D$ *up to label permutation.*

Proof. Let $\mathbf{X} = (X_1, \ldots, X_D)$ be a mixture random variable with pdf $f(\boldsymbol{x}, \boldsymbol{\psi}^*)$ and define the $\sum_d B_d$–dimensional random variable $\mathbf{M}$ with components $(1_{a_{d(b_d-1)} \leq X_d < a_{db_d}})_{d=1,\ldots,D; b_d=1,\ldots B_d}$, margins of the raw observation $\mathbf{X}$ on the D-dimensional grid. Then $\boldsymbol{m}$ is the sum of n outcomes of i.i.d. random variables having $\mathbf{M}$ law. Hence, $\frac{1}{n}\tilde{\ell}_m(\boldsymbol{\psi}; \boldsymbol{m})$ converges in probability to the contrast function $F(\boldsymbol{\psi}) = \mathbb{E}_{\boldsymbol{\psi}^*}[\tilde{\ell}_m(\boldsymbol{\psi}; \mathbf{M})]$ when $n \to \infty$, uniformly in the parameter.

We have to show that the following inequality holds:

$$\mathbb{E}_{\boldsymbol{\psi}^*}[\tilde{\ell}_m(\boldsymbol{\psi}^*; \mathbf{M})] > \mathbb{E}_{\boldsymbol{\psi}^*}[\tilde{\ell}_m(\boldsymbol{\psi}; \mathbf{M})] \quad \forall \boldsymbol{\psi} \neq \boldsymbol{\psi}^*, \tag{3.13}$$

while the corresponding equality holds for $\boldsymbol{\psi} = \boldsymbol{\psi}^*$ (up to label permutation). In this case we will say that there is asymptotic identifiability. From the definition of the bin-marginal composite log-likelihood, we have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\psi}^*}[\tilde{\ell}_m(\boldsymbol{\psi}^*; \mathbf{M})] - \mathbb{E}_{\boldsymbol{\psi}^*}[\tilde{\ell}_m(\boldsymbol{\psi}; \mathbf{M})] \\
&= \mathbb{E}_{\boldsymbol{\psi}_1^*}[\ell_1(\boldsymbol{\psi}_1^*; \mathbf{M}_1)] - \mathbb{E}_{\boldsymbol{\psi}_1^*}[\ell_1(\boldsymbol{\psi}_1; \mathbf{M}_1)] + \ldots \\
&+ \mathbb{E}_{\boldsymbol{\psi}_D^*}[\ell_D(\boldsymbol{\psi}_D^*; \mathbf{M}_D)] - \mathbb{E}_{\boldsymbol{\psi}_D^*}[\ell_D(\boldsymbol{\psi}_D; \mathbf{M}_D)],
\end{aligned}$$

where $\mathbf{M}_d$ is a B_d dimensional random variable with components $(1_{a_d(b_d-1) \leq X_d < a_d b_d})_{b_d=1,\ldots,B_d}$. For all log-likelihoods ℓ_d, $d = 1, \ldots, D$, inequality (3.13) holds. Thus, for all $\boldsymbol{\psi}_1 \neq \boldsymbol{\psi}_1^*, \ldots \boldsymbol{\psi}_D \neq$

$\boldsymbol{\psi}_D^*$:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\psi}_1^*}[\ell_1(\boldsymbol{\psi}_1^*;\mathbf{M}_1)] > \mathbb{E}_{\boldsymbol{\psi}_1^*}[\ell_1(\boldsymbol{\psi}_1;\mathbf{M}_1)]\\
&\vdots\\
&\mathbb{E}_{\boldsymbol{\psi}_D^*}[\ell_D(\boldsymbol{\psi}_D^*;\mathbf{M}_D)] > \mathbb{E}_{\boldsymbol{\psi}_D^*}[\ell_D(\boldsymbol{\psi}_D;\mathbf{M}_D)]
\end{aligned}$$

and we have equality for $\boldsymbol{\psi}_1 = \boldsymbol{\psi}_1^*, \ldots, \boldsymbol{\psi}_D = \boldsymbol{\psi}_D^*$, up to label permutation.

As it is well-known for mixtures, each equality hold up to a permutation: so we can define a set of D permutations named $\rho_1, \ldots, \rho_D$. In the hypothesis of our proposition, which assures that the marginal mixtures have the same number of components of the original ones and different proportions, we can match uniquely the components thanks to proportions matching. Therefore, the D permutations reduce to only one (named ρ) and $\boldsymbol{\psi}^*$ is equal to $\boldsymbol{\psi}$ after ρ. So, in this case, asymptotic identifiability is fulfilled. □

Joint identifiability The use of composite likelihood sets another kind of identifiability issue that has to be considered despite the results obtained in Section 3.2.2. Indeed, we have to show if the joint D-dimensional structure can be uniquely identified looking at only the D marginal distributions. Given two bin-marginal D-variate mixtures $p_m(\boldsymbol{m};\boldsymbol{\psi})$ and $p_m(\boldsymbol{m};\boldsymbol{\psi}')$, we have to prove that $p_m(\boldsymbol{m};\boldsymbol{\psi}) = p_m(\boldsymbol{m};\boldsymbol{\psi}')$, knowing equalities between marginals. The conditions given by Ranalli and Rocci (2017b) in a raw data case and our condition on grid refinement are sufficient to achieve this kind of identifiability even in the bin-marginal case. We point out that these conditions are the same defining the null measure set $\boldsymbol{\Psi}^\dagger$ in the previous propositions, as reported in the following result.

Proposition 3.3.2. *Let $p_m(\boldsymbol{m};\boldsymbol{\psi})$ and $p_m(\boldsymbol{m};\boldsymbol{\psi}')$ two D-variate bin-marginal mixtures with K components such that $p_d(\boldsymbol{m}_d;\boldsymbol{\psi}_d) = p_d(\boldsymbol{m}_d;\boldsymbol{\psi}'_d)$ for all $d = 1, \ldots, D$. Assume that $\boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$ and $R_d > 4K - 3$, $d = 1, \ldots, D$. Then $p_m(\boldsymbol{m};\boldsymbol{\psi}) = p_m(\boldsymbol{m};\boldsymbol{\psi}')$.*

Proof. As $\boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$, marginals $p_d(\boldsymbol{m}_d;\boldsymbol{\psi}_d)$ and $p_d(\boldsymbol{m}_d;\boldsymbol{\psi}'_d)$ for all $d = 1, \ldots, D$ have K components. From $p_d(\boldsymbol{m}_d;\boldsymbol{\psi}_d) = p_d(\boldsymbol{m}_d;\boldsymbol{\psi}'_d)$, the hypotheses on grid refinement $R_d > 4K - 3$ and $\boldsymbol{\psi}, \boldsymbol{\psi}' \in \boldsymbol{\Psi}/\boldsymbol{\Psi}^\dagger$ (in particular that proportions π_k are different), we deduce thanks to Proposition 2.2.2 that $\pi_k = \pi'_k, k = 1, \ldots, K$, and that the labeling order of the components is the same. Using the same hypotheses and Proposition 2.2.2, it follows also that $\mu_{kd} = \mu'_{kd}$ and $\sigma^2_{kd} = \sigma^{2\prime}_{kd}$, for $k = 1, \ldots, K$ and $d = 1, \ldots, D$. This means that $\boldsymbol{\mu}_k = \boldsymbol{\mu}'_k$ and $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}'_k$ for $k = 1, \ldots, K$. This completes the proof. □

Necessary conditions for joint identifiability As in Section 3.2.2, we can also provide a necessary condition. Similarly to what is reported in Ranalli and Rocci (2016a), we can state that a necessary condition to infer the joint structure of a mixture knowing only univariate marginals is

$$\sum_{d=1}^{D} R_d \geq 2DK + K - 1, \tag{3.14}$$

This is because the bin-marginal likelihood is the product of all univariate binned likelihoods and thus the maximum number of estimable parameters is equal to the number of parameters of a main effects log-linear model.

3.3.4 Bin-marginal CL-EM algorithm

We can now maximize (3.10) using an EM-like approach. At each data $\boldsymbol{m}_d, d = 1, \ldots, D$ we associate the *missing* vectors $(\mathbf{x}_d, \mathbf{z}_d), d = 1, \ldots, D$, where $\mathbf{x}_d$ contains the component d of the raw data $\mathbf{x}$ and $\mathbf{z}_d$ is the indicator membership matrix for $\mathbf{x}_d$. Thus, it is an $n \times K$ matrix whose generic element z_{dik} is equal to 1 if $\boldsymbol{x}_{id}$ belongs to population k, 0 otherwise.

To simplify the notation, we set $\tilde{\mathbf{z}} = \{\mathbf{z}_1, \ldots, \mathbf{z}_D\}$: the couple $(\mathbf{x}, \tilde{\mathbf{z}})$ is named *complete* data. Then, we introduce the *complete marginal composite log-likelihood*:

$$\tilde{\ell}^c_m(\boldsymbol{\psi}; \mathbf{x}, \tilde{\mathbf{z}}) = \sum_{d=1}^{D} \ell^c_d(\boldsymbol{\psi}_d; \mathbf{x}_d, \mathbf{z}_d), \tag{3.15}$$

where $\ell^c_d(\boldsymbol{\psi}_d; \mathbf{x}_d, \mathbf{z}_d)$ denotes the complete log-likelihood for the d-th marginal couple of data $(\mathbf{x}_d, \mathbf{z}_d)$.

At iteration $j \geq 0$, $\boldsymbol{\psi}^{(j)}$ denotes the current estimate for $\boldsymbol{\psi}$. Then, denoting respectively with $\mathbf{X}_d$ and $\mathbf{Z}_d$ the random variables generating $\mathbf{x}_d$ and $\mathbf{z}_d$, we now define the quantity:

$$\tilde{Q}_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)}) = \sum_{d=1}^{D} \mathbb{E}_{\boldsymbol{\psi}^{(j)}_d}[\ell^c_d(\boldsymbol{\psi}_d; \mathbf{X}_d, \mathbf{Z}_d) | \boldsymbol{m}_d],$$

where the expectations are taken with respect to the conditional densities $f(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{m}_d; \boldsymbol{\psi}^{(j)}_d)$, $d = 1, \ldots, D$.

Let re-write $\tilde{Q}_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$, indicating with $\mathcal{X}_d \times \mathcal{Z}_d$ the integration domain of $(\mathbf{x}_d, \mathbf{z}_d)$. We have

$$\begin{aligned}\tilde{Q}_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)}) = \sum_{d=1}^{D} \int_{\mathcal{X}_d \times \mathcal{Z}_d} & \ell^c_d(\boldsymbol{\psi}_d; \mathbf{x}_d, \mathbf{z}_d) \\ & \times f(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{m}_d; \boldsymbol{\psi}^{(j)}_d) d\mathbf{x}_d d\mathbf{z}_d.\end{aligned}$$

Now, we can define our bin-marginal CL-EM algorithm, whose fundamental steps are resumed in Algorithm 1. Therein $\mathcal{B}^d_{b_d}$ indicates the b_d-th interval bin on the $d-$th dimension.

Initialization We adopt a uniform random initialization for proportions, means and variances. In particular, for each dimension, means are values extracted uniformly from the range of values of the data and variances are positive uniform values lower than the variance of the data.

Algorithm 1 Bin-marginal CL-EM algorithm for D-dimensional Gaussian diagonal mixtures

1. Initialization phase: provide an initial guess $\boldsymbol{\psi}^{(0)}$.
2. For $j \geq 0$:
   - **Binned CL-E Step:** Given the estimate $\boldsymbol{\psi}^{(j)}$, calculate $\tilde{Q}_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$;
   - **Binned CL-M Step:** Obtain the new estimate $\boldsymbol{\psi}^{(j+1)}$, maximizing $\tilde{Q}_m(\boldsymbol{\psi}, \boldsymbol{\psi}^{(j)})$.

   For $d = 1, \ldots, D$:
   For $b_d = 1, \ldots, B_d$:

$$g_{db_d}^{(j)}(x_d) = \frac{f(x_d; \psi_d^{(j)})}{\int_{\mathcal{B}_{b_d}^d} f(y_d; \psi_d^{(j)}) dy_d}$$

   For $k = 1, \ldots, K$ and $d = 1, \ldots, D$:

$$\begin{aligned}
\tau_{kd}^{(j)}(x_d) &= \frac{\pi_k^{(j)} \phi(x_d; \mu_{kd}^{(j)}, \sigma_{kd}^{2(j)})}{f(x_d; \psi_d^{(j)})} \\
\pi_k^{(j+1)} &= \frac{\sum_{d=1}^{D} \sum_{b_d=1}^{B_d} m_{db_d} \int_{\mathcal{B}_{b_d}^d} \tau_{kd}^{(j)}(x_d) g_{db_d}^{(j)}(x_d) dx_d}{Dn} \\
\mu_{kd}^{(j+1)} &= \frac{\sum_{b_d=1}^{B_d} m_{db_d} \int_{\mathcal{B}_{b_d}^d} x_d \tau_{kd}^{(j)}(x_d) g_{db_d}^{(j)}(x_d) dx_d}{\sum_{b_d=1}^{B_d} m_{db_d} \int_{\mathcal{B}_{b_d}^d} \tau_{kd}^{(j)}(x_d) g_{db_d}^{(j)}(x_d) dx_d} \\
\sigma_{kd}^{2(j+1)} &= \frac{\sum_{b_d=1}^{B_d} m_{db_d} \int_{\mathcal{B}_{b_d}^d} (x_d - \mu_{kd}^{(j)})^2 \tau_{kd}^{(j)}(x_d) g_{db_d}^{(j)}(x_d) dx_d}{\sum_{b_d=1}^{B_d} m_{db_d} \int_{\mathcal{B}_{b_d}^d} \tau_{kd}^{(j)}(x_d) g_{db_d}^{(j)}(x_d) dx_d}
\end{aligned}$$

   Stop if (3.16) is verified, continue otherwise.

Stopping rule Binned CL-EM algorithm stops as soon as

$$\left| \frac{\tilde{\ell}_m(\boldsymbol{\psi}^{(j+1)}; \boldsymbol{m}) - \tilde{\ell}_m(\boldsymbol{\psi}^{(j)}; \boldsymbol{m})}{\tilde{\ell}_m(\boldsymbol{\psi}^{(j)}; \boldsymbol{m})} \right| < \epsilon, \tag{3.16}$$

where ϵ is a chosen threshold.

Obtaining the final clustering partition Once obtained the final estimate of $\boldsymbol{\psi}$ provided by our CL-EM algorithm, namely $\hat{\boldsymbol{\psi}}$, we recover the final clustering partition using a maximum a posteriori probability (MAP) rule. It means that the estimated labels $\hat{\boldsymbol{z}} = (\hat{z}_1, \ldots, \hat{z}_n)$ are given by:

$$\hat{z}_i = \underset{1 \leq k \leq K}{\operatorname{argmax}} \, \hat{\pi}_k \phi(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \quad i = 1, \ldots, n.$$

We highlight this algorithm involves only D binned vectors of dimension $B_d = R_d + 1$, $d = 1, \ldots, D$ and only univariate integrals. Thus, our proposal is able to solve our initial issues linked to storage and complexity.

3.4 Numerical experiences on simulated data

In this section we apply the methodology to different simulated data sets in order to show in controlled frameworks its ability to recognize the minority class.

Table 3.1: Description of the fifteen scenarios. Covariance matrices $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$ are equal to the identity matrix $\boldsymbol{I}_3$ and $\pi_2 = 1 - \pi_1$.

Scenario	Separation	Imbalance	Small class proportion (π_1)	Means
HH	High	High	10^{-4}	$\boldsymbol{\mu}_1 = (-4, -4, -4)$ $\boldsymbol{\mu}_2 = (4, 4, 4)$
HM		Medium	10^{-3}	
HL		Low	10^{-2}	
MH	Medium	High	10^{-4}	$\boldsymbol{\mu}_1 = (-3, -3, -3)$ $\boldsymbol{\mu}_2 = (3, 3, 3)$
MM		Medium	10^{-3}	
ML		Low	10^{-2}	
LH	Low	High	10^{-4}	$\boldsymbol{\mu}_1 = (-2, -2, -2)$ $\boldsymbol{\mu}_2 = (2, 2, 2)$
LM		Medium	10^{-3}	
LL		Low	10^{-2}	
VH	Very low	High	10^{-4}	$\boldsymbol{\mu}_1 = (-1, -1, -1)$ $\boldsymbol{\mu}_2 = (1, 1, 1)$
VM		Medium	10^{-3}	
VL		Low	10^{-2}	
1HH	One separated component	High	10^{-4}	$\boldsymbol{\mu}_1 = (-1, -1, -4)$ $\boldsymbol{\mu}_2 = (1, 1, 4)$
1HM		Medium	10^{-3}	
1HL		Low	10^{-2}	

Our second aim is also to compare it to two possible competitors: classic estimation with the full data set and a subsampling strategy. We will evaluate their performances in terms of clustering quality, measured by the ARI score (Hubert and Arabie, 1985), and also in terms

of both time and memory consumption. In particular, the full data set will be our benchmark in terms of clustering quality, but it will be discarded as it is too much burdensome. The subsampling will prove to cope with our computational constraints, but resulting usually in bad clustering performances or in, even, estimation failures.

In these simulations, and in real applications of Section 3.5 as well, we suppose the true number of components K is known and fixed. Actually, the definition of a criterion to choose the right number of components is needed to complete our analyses. It is true that examples of model choice criteria based on penalized composite likelihood have already been defined (Varin et al., 2011; Ranalli and Rocci, 2016b), but their application on our case requires particular care, due to numerical complexity. Thus, this topic needs further research and, possibly, new criteria to define. It is for this very reason that on this work we prefer to test the potential of our method in simulations where K is fixed. The definition of suitable model choice criteria will be debated in future works.

3.4.1 Experimental settings

Simulation analyses are conducted on data sets with 1 million data generated from several 3-dimensional two classes mixtures, different in proportions assigned to the minority class and also in means, while both covariance matrices remain equal to the identity matrix. These differences are crucial because lowering proportion of the smallest class corresponds to more difficulties in detecting it and changing means helps us in controlling classes separation and, thus, clustering complexity.

We divide our simulations into two main parts: in the first one, cluster separation is equal for all axes, while, in the second one, clusters are well separated only on one axis, while on the other two they are not. This is useful to understand the degree of separation needed by our technique. In particular, in the first part, we gradually increase the small class proportion three times from 10^{-4} to 10^{-2} and we also propose four separation degrees for cluster means, equal to $8, 6, 4, 2$ in terms of absolute difference between them. Their combination results in twelve different scenarios. Each scenario is named by using two letters: the first one (H, M, L, V) refers to the degree of separation of the scenario (respectively: high, medium, low and very low); the second one (H, M, L) refers to the imbalance of the data set (high, medium and low). Three additional scenarios consist in a variation of scenarios HH-HM-HL where the first two dimension have the lowest separation degree, while there is a high separation on the third axis. Their names are 1HH-1HM-1HL, reminding that here high separation is present only on one axis. Table 3.1 details all these fifteen settings.

Regarding the three analyzed methods, we decide to compare subsampling and our bin-marginal proposal under the same memory constraints. Bin marginal uses a grid refinement R, leading to use a $2R$ memory space (binned data itself and grid); hence, subsampling is conducted with a subsample of size $2R$ to be fair. At the same time, we also analyze the influence of the grid refinement on the binned estimation and, consequently, the effect of the subsample size on the subsampling performance. In practice, the refinement can be fixed to $50, 100, 200$ and, consequently, subsample sizes can be $100, 200$ or 400. For each scenario, we simulated 50 different data sets of equal size (1 million) to have consistent results. To evaluate its variability, subsampling performances are evaluated on 100 different subsamples. On the same sample, all

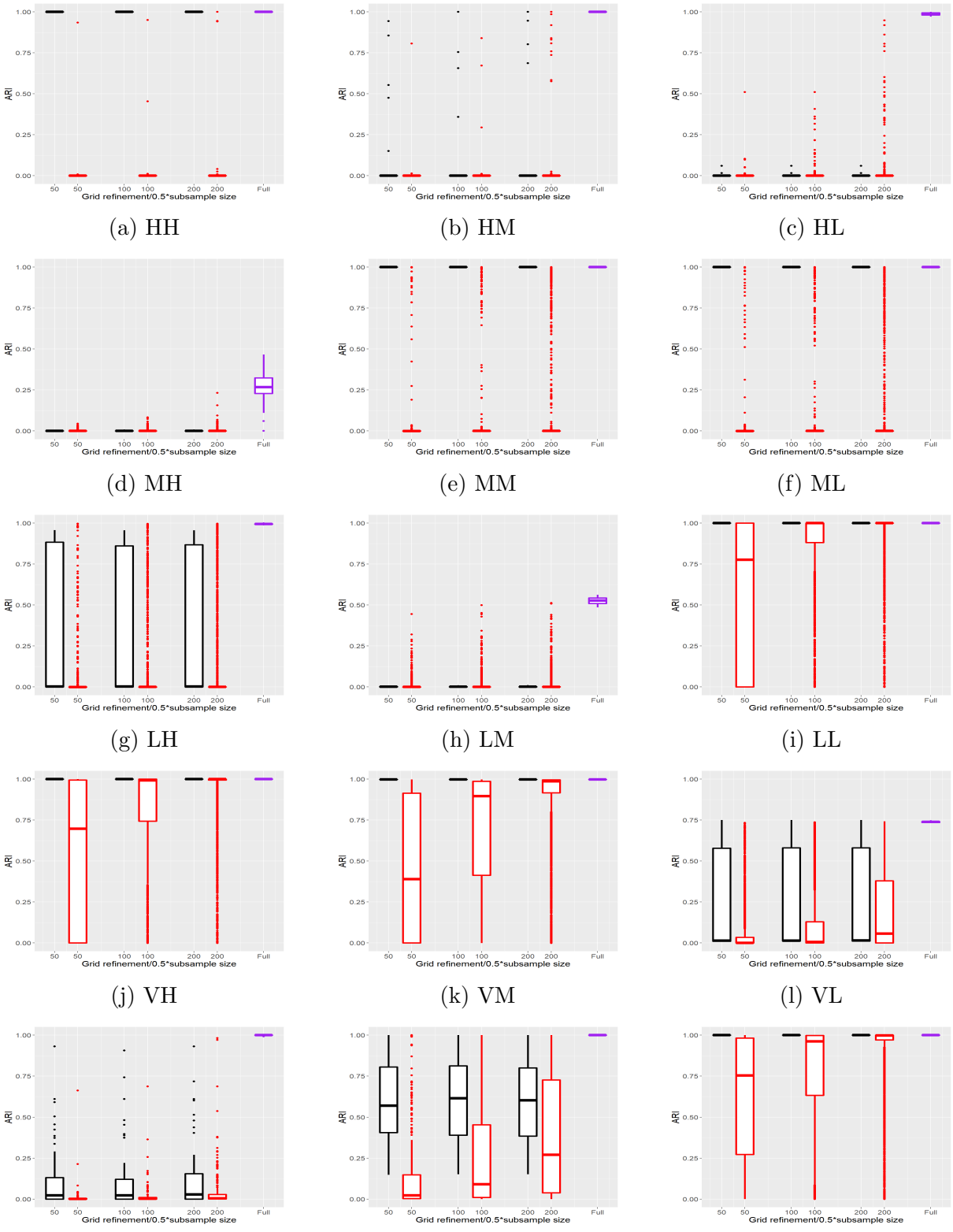
the three algorithms start from the same initialization points, in order to correctly evaluate their performances. Practical implementation, both of simulations and real application as well, was done in the R environment (R Core Team, 2021). More precisely, we used the routines of the R package `mclust` (Fraley et al., 2012) for the two competitors and a self-written code for our bin-marginal technique.

3.4.2 Results

Clustering quality and memory Figures 3.4a-3.4o depict the results of the simulations. Mostly, our proposal outperforms subsampling in all the settings with good performances even with very coarse grids. It encounters some difficulties only in very hard scenarios where separation and proportion are very small. Generally, it approaches with a low consumption the results obtained with the full data set, which, on the contrary, uses a huge amount of memory.

Failures There is another virtue in binned strategy: in fact, subsampling can fail, i.e., the algorithm does not provide any result, as reported in Figure 3.5. It appears the probability of failure increases if separation increases and imbalance ratio decreases. This is quite astonishing, as we expected more failures in a more imbalanced data set, but it is not completely incoherent: in fact, results show that if subsampling does not fail (high imbalancement) it works badly; if on the contrary it can provide good results, it is prone to failures (low imbalancement). In most of scenarios, failures surprisingly increase according to subsample size. At this moment, we are not able to explain exactly the reason of this unusual behaviour, that probably could be resolved by changing the initial settings of EM (implemented in `mclust`). But, fortunately, this does not affect directly our proposal based on binned data.

Time Finally, Figure 3.6 shows time performances for the three strategies. Our CL-EM algorithm does not outperform subsampled EM in execution time, while it is faster than full data set EM. This result is coherent with our expectations. Indeed, even if both CL-EM and classic EM are linear with respect to input size (R and n respectively), the operations executed by CL-EM are more complex due to the presence of integrals (see Algorithm 1). Thus, if R and n are comparable (subsampling case), CL-EM is slower than classic EM, while it is faster if $R \ll n$ (full data set case). In analyzing Figure 3.6 we also have to point out that the `mclust` package is well-optimized. A possible way to speed up our code is the employment of Rcpp (Eddelbuettel and François, 2011), which enables integration between R and C++. According to Aruoba and Fernández-Villaverde (2015), Rcpp is faster than R about 100 times, so our time performances has to be scaled of at least a factor 100. The figure itself pictures our predicted performance after code optimization (blue boxplots), showing a remarkable improvement relatively to full data set analysis.



(a) HH (b) HM (c) HL

(d) MH (e) MM (f) ML

(g) LH (h) LM (i) LL

(j) VH (k) VM (l) VL

(m) 1HH (n) 1HM (o) 1HL

Figure 3.4: Clustering performances for subsampled EM (red boxplots), bin-marginal CL-EM (black boxplots) and full data EM (purple boxplots) expressed in terms of ARI in dependence on grid refinement/subsample size under condition of equal memory occupancy. Imbalance is decreasing from left to right and separation is decreasing from top to bottom.

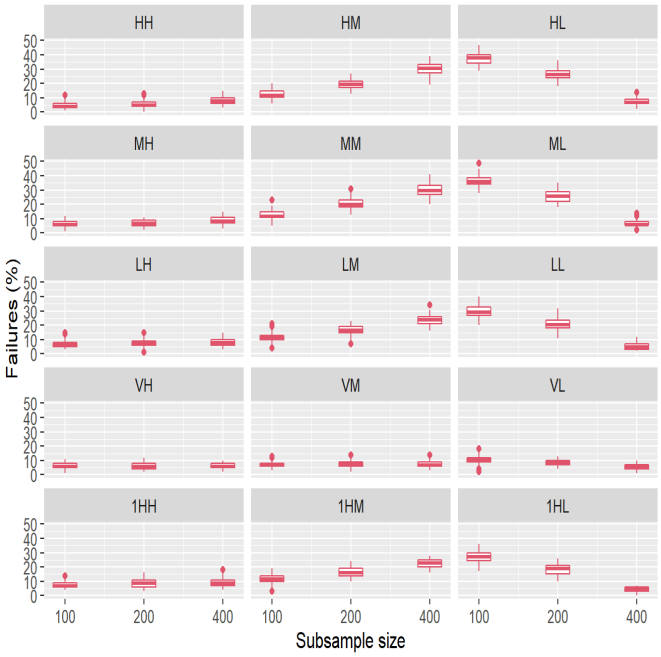


Figure 3.5: Percentage of subsampled EM failures.

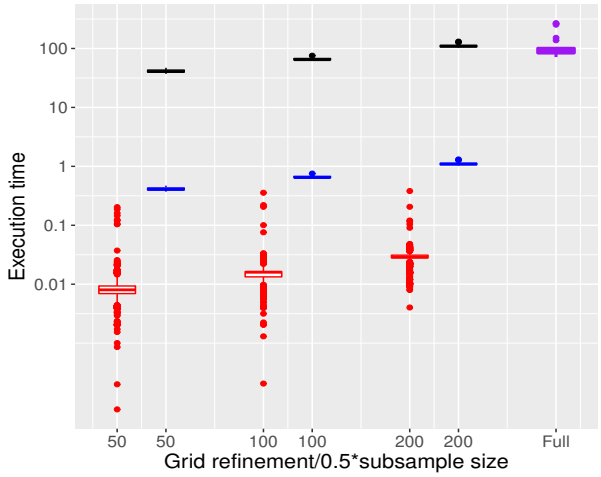
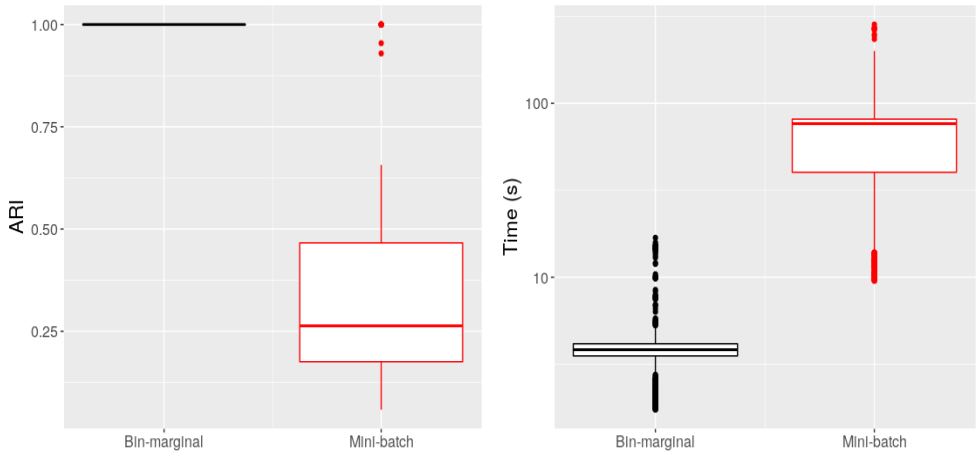


Figure 3.6: Scenario HH: execution time (in s) comparison between subsampled EM (red boxplots) and bin-marginal CL-EM (black boxplots) in dependence on grid refinement/subsample size in condition of equal memory occupancy. Blue boxplots show expected CL-EM time after optimization in language C++, while the purple boxplot represents time performance for the full data set analysis.



(a) (b)

Figure 3.7: Result comparison bin-marginal EM (black boxplots) and mini-batch EM (red boxplots).

Comparisons with multi-samples methods In these simulations we have seen that our proposal outperforms subsampling under the same memory restriction. Interesting further comparisons could be done with methods employing several subsamples in the same estimation process, such as the mini-batch EM described in Nguyen et al. (2020). Actually, our method remains competitive with them despite the increase of available information. Let consider a bivariate simulation involving data generating by a two-classes Gaussian mixtures with the following parameters:

$$\begin{aligned}\pi_1 &= 10^{-4}\\ \boldsymbol{\mu}_1 &= (-4,-4)\\ \boldsymbol{\mu}_2 &= (4,4)\\ \boldsymbol{\Sigma}_1 &= \boldsymbol{\Sigma}_2 = \boldsymbol{I}_2\end{aligned}$$

We simulated 50 different data sets of equal size (1 million) to have consistent results. For our method we used marginal grids with refinement 100, while the mini-batch EM uses the initial settings recommended in the cited reference. In each simulation, we initialized both algorithms from the same 100 starting points and we selected the result providing the best bin-marginal likelihood (for our method) and the full data raw likelihood (for the competitor, this information is also provided by StoEMMIX routines). For the two best results we calculated the inducted partitions and the corresponding ARI scores (Figure 3.7a). We also computed the time for a single execution of both algorithms (Figure 3.7a). From the analysis of these results, we can conclude that our method performs better than mini-batch EM. Moreover, mini-batch EM is much slower than bin-marginal EM, due to the drawing of several samples.

3.5 Real data sets

The presented methodology is now applied to several real imbalanced data sets. Here we show three applications from different fields of interest, which are image segmentation, fraud detection and recognition of potentially hazardous asteroids. In the last two cases, we have considered a subset of three variables for each data set. We have chosen those ones whose histograms visually resulted to be close to GMM hypotheses and with a low percentage of missing values (less than the 5% of the original data). A comprehensive view of the used data sets is given in Table 3.2.

Table 3.2: Real data sets description.

Data set	n	D	Small class proportion
Cell-1	101,430	3	unknown
Cell-2	65,536	3	unknown
Cell-3	685,020	3	unknown
Comet	1,083,681	3	unknown
Asteroids	932,341	3	0.002
Credit card	284,807	3	0.0014

3.5.1 Data sets and methods

Image segmentation Image segmentation (Pal and Pal, 1993) consists in partitioning an image into homogeneous parts and it is useful to detect and locate objects. Here we focus on those images where there are very tiny objects: for this purpose we segment three cell images available on Kaggle (To, 2021) and an image picturing a distant active comet observed by NASA's Hubble Space Telescope (NASA, 2017). After a brief pre-processing phase, these images result in 3-dimension data sets with a number of records ranging from 65,536 to 1,083,681. The lines of these data sets correspond to RGB pixels, that could be analyzed with our method.

Asteroids Asteroid data set is a collection of information about asteroids available on Kaggle (Hossain, 2020). It consists in 958,524 records of 45 variables. The purpose of the analysis is to detect potentially hazardous asteroids (PHAs), which are those asteroids approaching very close to the Earth. In particular, an asteroid with small magnitude (H) and Earth minimum orbit intersection distance (moid) is considered a PHA (Quarta and Mengali, 2010). We use only a subset of the features contained in this data set, using these two variables and adding information regarding orbit eccentricity in order to remain in a more interesting 3-dimensional problem where our method has already been tested in the simulation phase. Due to the presence of missing values, the analyzed data set contains now 932,341 records out of 958,524. The rest of the variables were discarded because they contain too many missing values (less than the 5% of the original data) and their histograms were judged not to be close to GMM hypothesis.

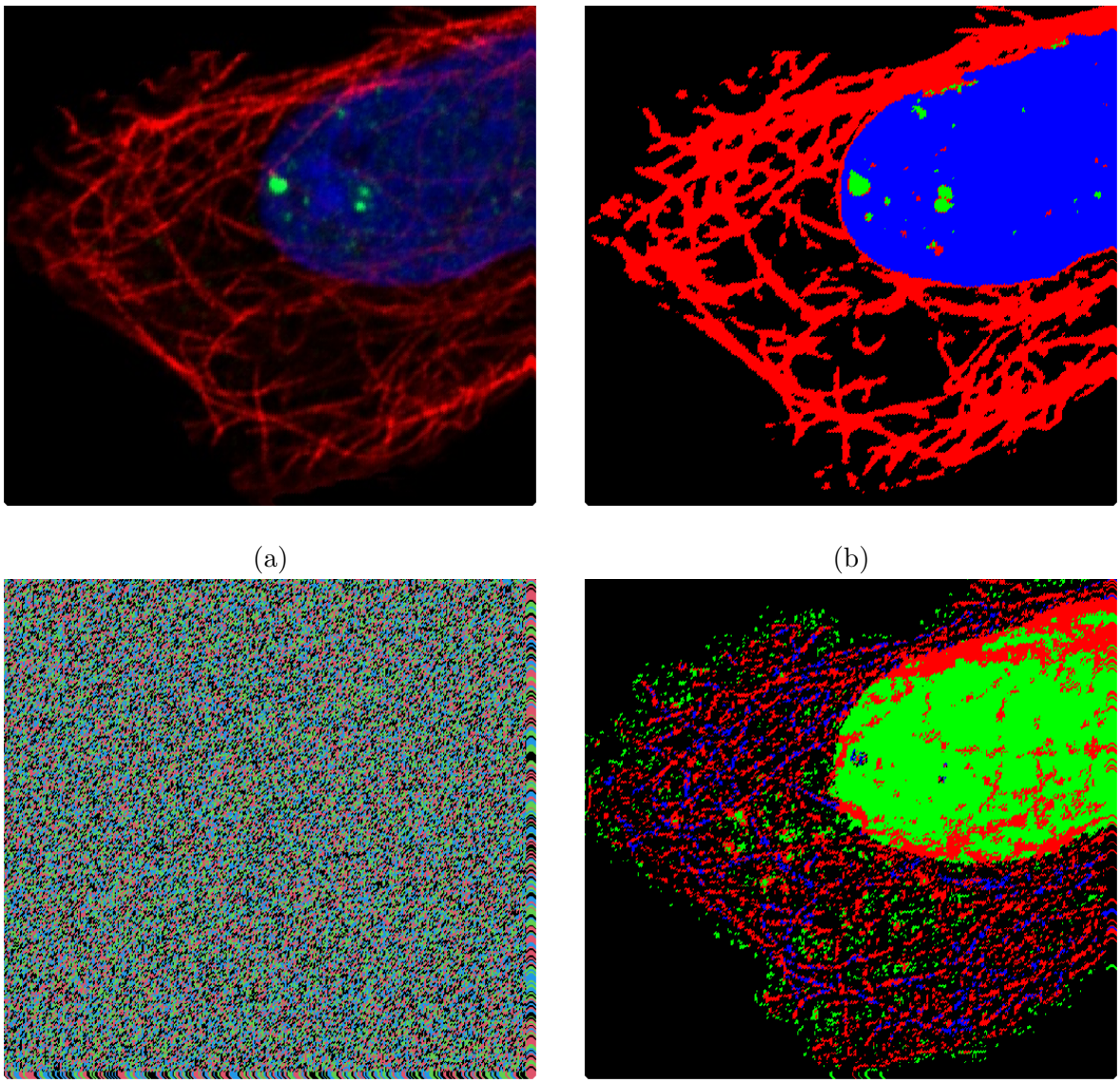
Credit card fraud detection Kaggle credit card data set (ULB, 2018) is a public repository which was massively analyzed in literature (Dal Pozzolo et al., 2017, 2014; Niu et al., 2019) to detect frauds. This data set contains 284,807 transactions, of which 492 are frauds, made by credit cards in September 2013 by European cardholders. All information given by 31 variables are anonymized and they are the result of a PCA transformation, so the original meaning of the variables is missed. Following the same ideas of the previous data set, we kept only three variables (V10-V14-V17), selecting those whose histograms seemed to be closer to Gaussian assumptions.

Methods For image segmentation, we will simply use the K-class partitions obtained with both our proposal and subsampling. For Asteroids and Credit Card data sets we perform a two-classes clustering comparing our method to both subsampled EM and full data set EM. Actually, true classification labels are provided by the original data sets but we use them only as a benchmark, as we want to follow a completely unsupervised approach. In particular we will employ them to rank results based on ARI score (Hubert and Arabie, 1985). Similarly to simulations, we used our self-written R code for bin-marginal CL-EM and `mclust` for all versions of classical EM .

3.5.2 Results and discussion

Image segmentation Figures 3.8-3.11 synthesize results obtained for the image segmentation of the four images. Figures marked with (a) represent the true images and those denoted with (b) the segmentation obtained with binned data. Finally, figures (c)-(d) are the best and worst (respectively associated to the full data set likelihood of the estimated parameter) segmentation obtained with classical subsampling in condition of equal memory occupancy. It can be seen that our method successfully detects the objects, while subsampling results in very noisy segmentations. Regarding the binning grid employed, we used marginal grids of refinement 20 for all Cell images and a finer ones with 400 intervals for Comet. In addition for Cell images we selected $K = 4$, where 4 colours are recognizable, and $K = 3$ for Comet, as in this image there is a consistent group of noise (represented in our segmentation by black points)

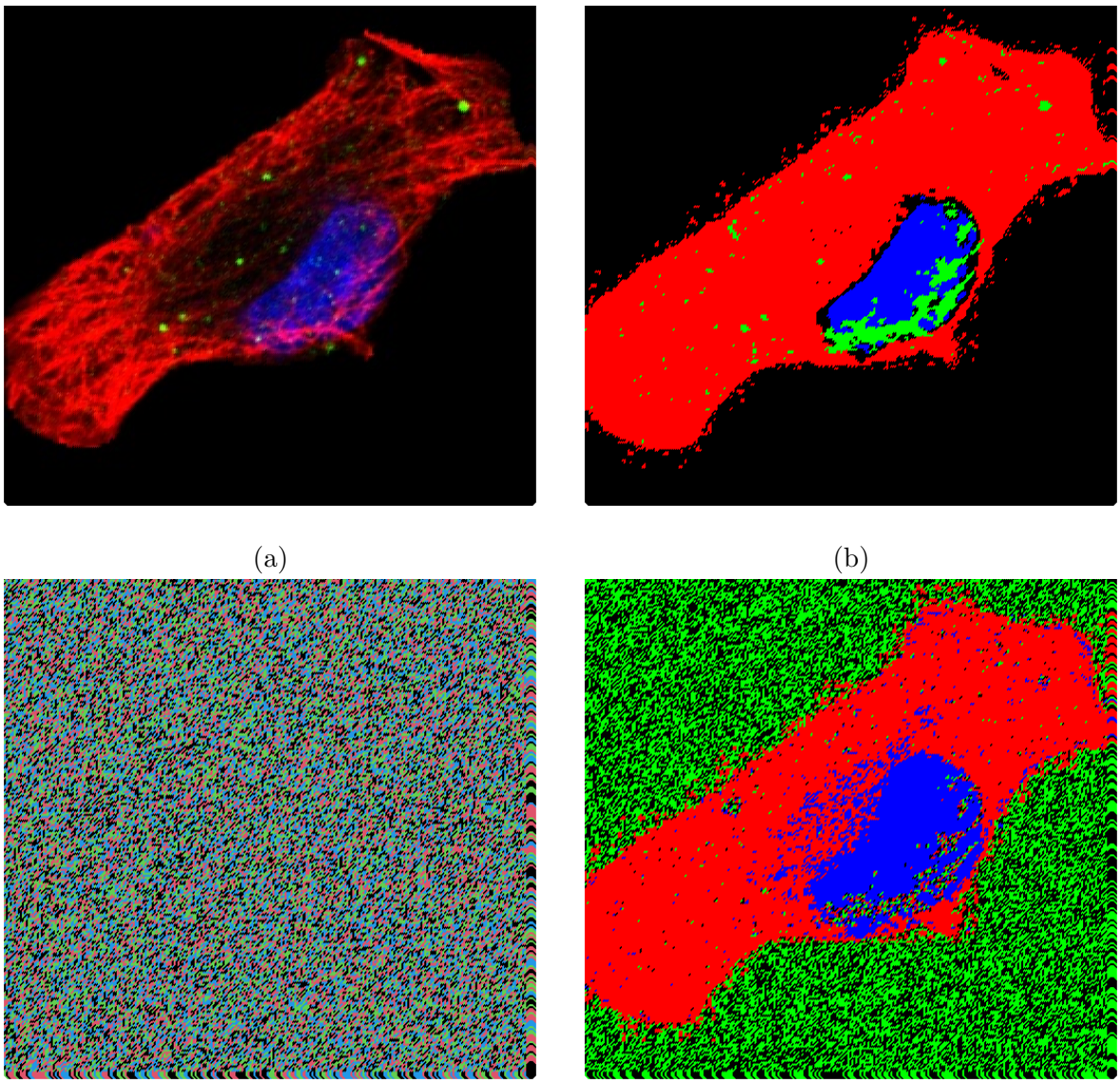
Asteroids Figure 3.12a reports the result of the comparison between our bin-marginal CL-EM and classical EM with both subsampling and full data set. In absolute terms, generically low ARI scores suggest that a total unsupervised approach could be very risky in this case. However, our objective is to analyze the results of our proposal relatively to our competitors. Concerning this, Figure 3.12a shows that, despite the loss of information, bin-marginal method (black circle) has globally better performances than both subsampling (red boxplot) and full data set EM (purple circle). Moreover, bin-marginal CL-EM is not prone to the variability of subsampling, whose result highy depends on subsample choice.



(a) (b)

(c) (d)

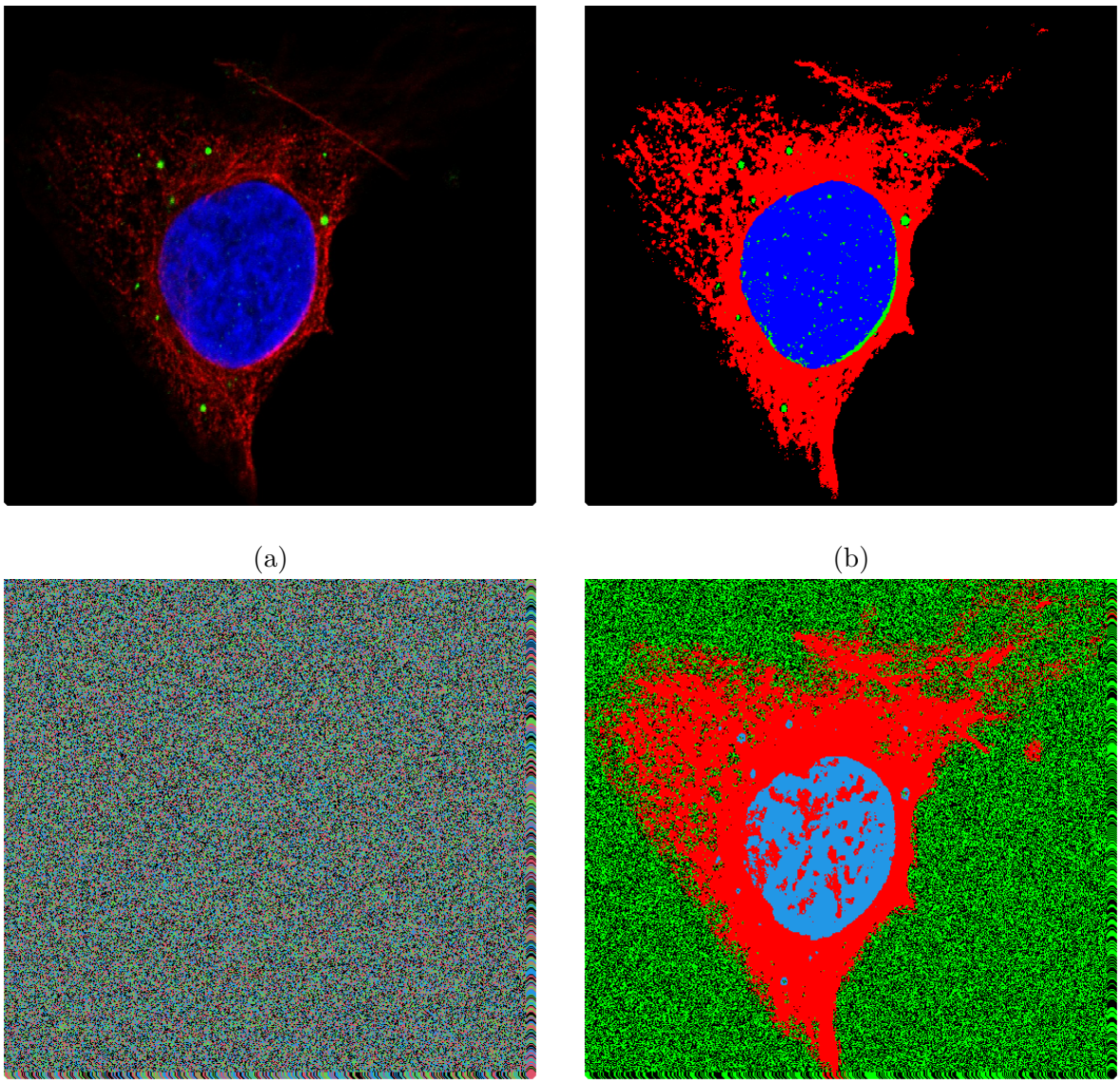
Figure 3.8: Cell-1 segmentation: (a) Original image; (b) Segmentation obtained with bin-marginal CL-EM; (c)-(d) Worst and best segmentation obtained with two subsampled EM.



(a) (b)

(c) (d)

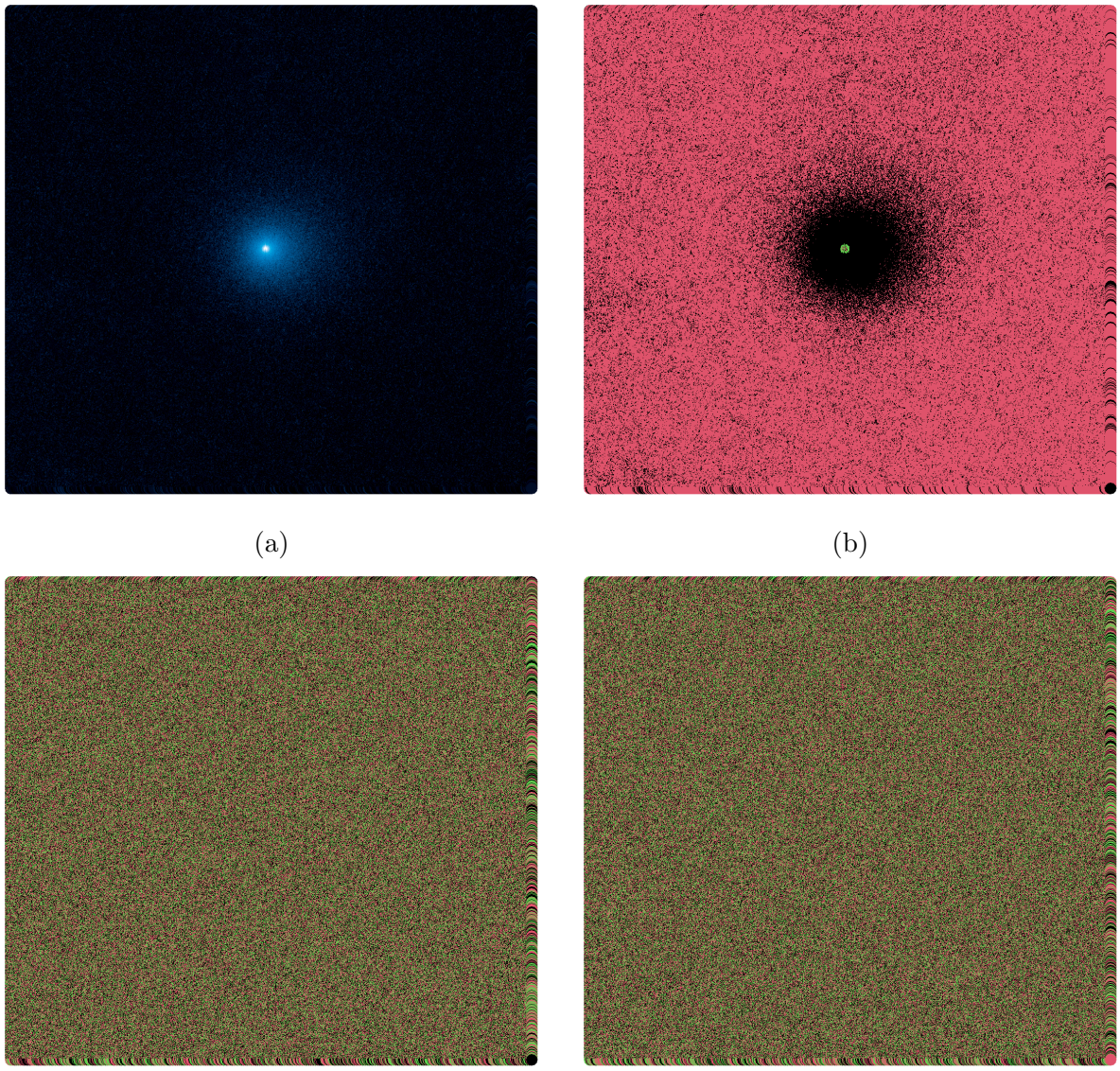
Figure 3.9: Cell-2 segmentation: (a) Original image; (b) Segmentation obtained with bin-marginal CL-EM; (c)-(d) Worst and best segmentation obtained with two subsampled EM.



(a) (b)

(c) (d)

Figure 3.10: Cell-3 segmentation: (a) Original image; (b) Segmentation obtained with bin-marginal CL-EM; (c)-(d) Best and worst segmentation obtained with two subsampled EM.

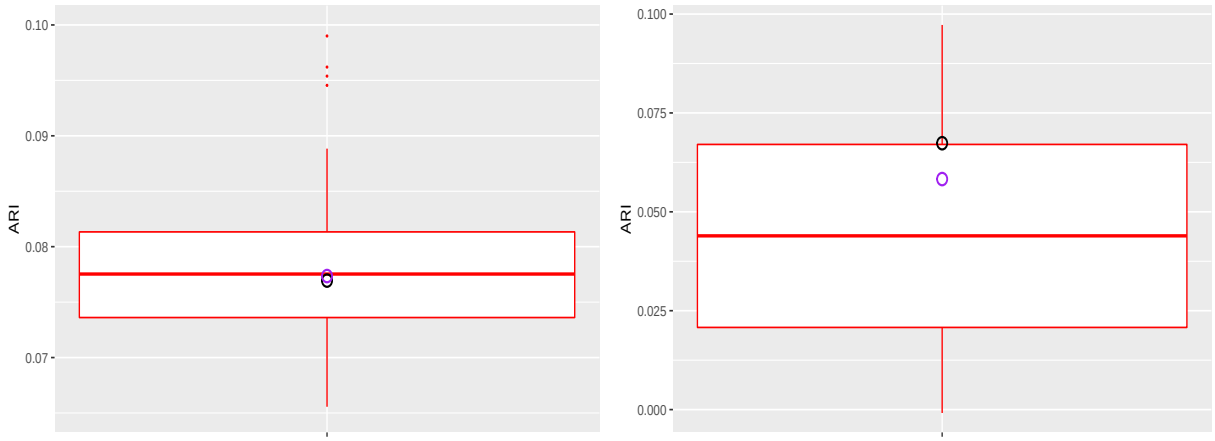


(a) (b)

(c) (d)

Figure 3.11: Comet image segmentation: (a) Original image; (b) Segmentation obtained with bin-marginal CL-EM; (c)-(d) Worst and best segmentation obtained with two subsampled EM.

Credit card fraud detection Following the same strategy used for Asteroids data set, we build a two-classes partition using our bin-marginal technique to detect frauds among the set of credit card transactions. Based on Figure 3.12b, similar comments could be made. Our method seems to be globally better with our direct competitors, avoiding the high variability of subsampling.



(a) (b)

Figure 3.12: Two-classes clustering performances in terms of ARI for subsampled EM (red boxplots), bin-marginal CL-EM (black circle) and full data set EM (purple circle). Data sets: (a) Asteroids; (b) Credit card fraud detection.

3.6 Conclusion

In this chapter we have defined a method based on Gaussian mixture models combining binned data with marginalization, which is able to detect, in an unsupervised way, imbalanced classes on large data sets under hard memory constraints. The theoretical results presented in this chapter have shown that the model and the proposed estimation procedure have good statistical properties, such as identifiability, despite the huge loss of statistical information caused by our heavy bin-marginal data compression. Both simulations and real applications have proved the competitiveness of our method with respect to the traditional subsampling method, in those cases where a full data set clustering is out of reach. In particular, it has revealed a great potential in the context of image segmentation when very tiny objects have to be detected.

These very encouraging results set the basis for a deeper research in topics commonly arising in Gaussian mixture modeling, such as the local maxima problem and the definition of a criterion to select the right number of components (thus, the number of classes). Proposing initialization strategies to avoid local maxima and model selection criteria is a fundamental step to further improve and automate our method. It is also important to quantify the impact of the binning

grid on clustering quality, in order to design highly efficient and frugal grids. Thus, we discuss in the next chapter these three topics from an experimental view, given first solutions and useful ideas for future research.

Chapter 4

Bin-marginal Gaussian mixtures: further experimental topics

The aim of this chapter is to study three topics regarding bin-marginal Gaussian mixtures: local maxima, definition of a model selection criteria and impact of the binning grid on clustering. These three problems are analyzed from an experimental point of view in imbalanced simulated scenarios, selected from the settings described in Table 3.1. The first topic regards a typical problem arising when a general mixture model is estimated through the maximization of its log-likelihood. This is because current estimation methods based on maximum likelihood principle, in particularly EM algorithm, can converge to points of local maximum instead of stabilizing around a point of absolute maximum. After a brief review about local maxima in raw and binned Gaussian mixture (Section 4.1), we experimentally explore this problem in bin-marginal Gaussian mixture models (Section 4.2). As regards the model choice, here we focus on the problem of choosing the right number of components for the underlying Gaussian mixture model. Typically, this issue is resolved through suitable choice criteria based on penalized form of model log-likelihood, as described in Section 4.3.1. As our estimation method is based on a marginal composite log-likelihood, we also review model choice criteria penalizing the marginal composite log-likelihood, which are more appropriate for our method. However, existing criteria (Section 4.3.2) turn out to be too difficult to compute (Section 4.3.3), so in Section 4.3.4 we define two new heuristics to select the right number of components in bin-marginal Gaussian mixtures. In Section 4.4, we try to understand how the refinement of the binning grid impacts the estimation process and, thus, the clustering. Moreover, as using a finer grid means more data to store, this has also an impact on the quantity of resources to employ. So, in this section we provide a first experimental guide to choose an optimal-refined grid which allows a good balance between clustering performances and employed resources.

4.1 Local maxima in raw and binned GMM

Mixture models, and Gaussian mixtures in particular, present two practical difficulties associated with their maximum likelihood estimation (Redner and Walker, 1984): firstly, mixture log-likelihood is not bounded above and it presents *spurious* local maxima with very high log-likelihood values (Kiefer and Wolfowitz, 1956); secondly, the log-likelihood function attains its largest local maximum value at different choices of $\boldsymbol{\psi}$ (*label switching*).

4.1.1 Spurious local maxima: definition and solutions

The log-likelihood of a Gaussian mixture may be unbounded or present local maxima (Day, 1969). Indeed, we may find high log-likelihood values corresponding to solutions where one of the estimated components has very small variance relative to the others (univariate case), or the determinant associated to the covariance matrix of a component is really small (multivariate case). In order to prevent the choice of these maxima, corresponding to *degenerate* solutions, several strategies have been proposed. Here, we revise techniques based on constrained estimation, strategies to find good initialization points for EM and stochastic variants of the EM algorithm itself.

Constrained estimation In Hathaway (1983) the idea of constraining the search of an estimate $\widehat{\boldsymbol{\psi}}$ into a specified subset of $\boldsymbol{\Psi}$ is widely debated, providing also guarantees about the consistency of the constrained solution. In the univariate setting, given a value $c \in (0, 1]$, this solution must lie in the subset given by:

$$\min_{j \neq k} \frac{\sigma_j^2}{\sigma_k^2} \geq c \quad j, k \in \{1, \ldots, G\}.$$

Thus, from a practical point of view, at the end of the EM algorithm it is possible to add a control phase regarding the relative magnitude of each variance. The same author (Hathaway, 1985) has proposed a similar rule in the multivariate case, where the MLE search is limited to the solutions respecting

$$\min_{j \neq k} \lambda(\boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_k^{-1}) \geq c \quad j, k \in \{1, \ldots, G\},$$

where $\lambda(\boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_k^{-1})$ denotes the collection of eigenvalues of the matrix $\boldsymbol{\Sigma}_j \boldsymbol{\Sigma}_k^{-1}$. Actually, this heuristic rule is hard to apply practically. In order to circumvent this problem, Ingrassia (2004) has proposed to search the MLE in the subset $\{\boldsymbol{\psi} \in \boldsymbol{\Psi} : a \leq \lambda_d(\boldsymbol{\Sigma}_k) \leq b, \;\; k = 1, \ldots, K, d = 1, \ldots, D\}$, where $a/b \geq c$, for $c \in (0, 1]$ and $\lambda_d(\boldsymbol{\Sigma}_k)$ is the $d-$th eigenvalues of Σ_k.

Initialization strategies Even if it is possible to constrain the optimization research in special subsets to avoid degenerate solutions, EM algorithm does not assure convergence towards the absolute maximum. This highly depends on how EM is initialized (Baudry and Celeux, 2015). Several authors propose different strategies to find good initialization points or variants of EM to be less dependent from the point of departure. In order to detect local maxima and reach the absolute maximum, it is suggested running EM several times starting from different initialization points and retaining that one with the highest likelihood (Biernacki et al., 2003). It is possible to choose the starting point randomly or in a deterministic way, through other clustering methods. A first random method consists in randomly assigning a subsample of data to the K components and, then, performing a first iteration of the M step on it (Coleman and Woodruff, 2000). Alternatively, one can choose equal proportions $\pi_k^{(0)} = 1/K$ for all k, variances equal to the covariance of the whole data set and means given by K subsampled data points (McLachlan and Peel, 2004). In addition, it is possible to evaluate the likelihood of several

initial random points and starting the EM routine from the best one (Maitra, 2009). Among the deterministic methods, we mention the use of K-means and hierarchical clustering (Scrucca and Raftery, 2015).

Stochastic variants of EM algorithm Another current option is to use variants of the EM algorithm where randomness is introduced to avoid local maxima (Celeux et al., 1995). The Stochastic EM (SEM) (Celeux and Diebolt, 1985) adds an intermediate stochastic step (S step) between E-step and M-step. At each iteration $j \geq 0$, group memberships for every observation $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iK})$ are drawn from a multinomial distribution with parameter given by $\boldsymbol{\tau}_i^{(j)} = (\tau_{i1}^{(j)}, \ldots, \tau_{iK}^{(j)})$. Then, M-step is performed on the complete log-likelihood as (estimated) group labels are known. The same authors proposed also a simulated annealing modification of SEM, the SAEM (Celeux and Diebolt, 1992). In this algorithm, traditional EM and SEM are both performed at the same iteration and their estimates are then combined with weights given by a sequence of positive decreasing real numbers (similarly to the Simulated Annealing algorithm (Van Laarhoven and Aarts, 1987)). The result is finally retained as the new estimate of the SAEM.

4.1.2 Label switching: definition and solutions

The problem of label switching is related to the particular mathematical form of the mixture density. Let consider a general permutation $\nu(\cdot)$ for the set of indexes $\{1, \ldots, K\}$. It is possible to define a corresponding permutation for the parameter $\boldsymbol{\psi}$:

$$\omega_\nu(\boldsymbol{\psi}) = (\pi_{\nu(1)}, \ldots, \pi_{\nu(K)}, \boldsymbol{\mu}_{\nu(1)}, \ldots, \boldsymbol{\mu}_{\nu(K)}, \boldsymbol{\Sigma}_{\nu(1)}, \ldots, \boldsymbol{\Sigma}_{\nu(K)}).$$

Then, it turns out that the likelihood $L(\boldsymbol{\psi})$ is the same for all permutations $\omega_\nu(\boldsymbol{\psi})$ (Stephens, 2000). This problem affects in particular methods based on a Bayesian approach and it is not a problem in the iterative computation of the MLE via the EM algorithm (our case) (McLachlan and Peel, 2004). A first solution to the problem consists in imposing identifiability constraints on the parameter space, such as $\pi_1 < \ldots < \pi_K$ (Stephens, 2000). As this solution is not effective for any given data set, researchers developed other approaches based on relabelling (Richardson and Green, 1997; Celeux, 1998; West, 1997).

4.1.3 Effect of binning on local maxima: the case of univariate binned GMM

The problem of local (degenerate) maxima for binned Gaussian mixtures has been deeply studied in Biernacki (2007) (in a univariate setting), presenting some particularities and previously unexpected behaviours. A first difference with its raw counterpart is that the binned likelihood of a Gaussian mixture is bounded above, as it is always lower than 1. However, this important characteristic does not prevent binned GMM from degeneracies. The author also notes a strong connection between the problem of local maxima for binned GMM and the choice of the binning grid, which is debated in Section 4.4. Indeed, the binned EM seems to converge more towards a degeneracy if the binning grid is rare. However, when the grid is not too fine, degeneracies can

be avoided as they tend not to be global maximizers. In addition, binned EM algorithm seems to move very slowly near a degeneracy which can act as both attractive and repulsive point for the algorithm itself. It means that, under certain conditions depending on data and model, an EM routine can "escape" from a degeneracy solution.

4.2 Local maxima in bin-marginal GMM

In this section, we investigate the local maxima problem for the bin-marginal likelihood with a practical point of view. Theoretically, we can say that the bin-marginal likelihood is bounded from above, as it is the sum of several binned univariate mixture likelihood (Section 4.1.3). The rest of the section is focused on practical experiments which show how many local maxima we can encounter, making comparisons with the traditional raw EM and the full binned EM (Algorithm 2 in Section 1.6.1). Furthermore, possible initialization rules are provided to find better starting points for our Bin-CL-EM algorithm.

4.2.1 Numerical experiments

Let consider again the scenarios HH and HL presented in Section 3.4 and described in Table 3.1. We consider the three algorithms (raw EM, full binned EM and Bin-CL-EM) with these initial settings: tolerance equal to 1e-8, maximum number of iterations fixed to 500, three marginal 100-bins grids for our bin-marginal method, a full $20 \times 20 \times 20$ three-variate grid for the full bin EM algorithm. Then, we run each algorithm from the same initial point. We consider a total of 200 different initializations. To simplify the visualization of results, we apply principal components to the data set containing all the 200 final estimates and we consider the first two principal components with maximum explained variance. Figures 4.1a-4.1c and 4.2a-4.2c show the results obtained. Bin-marginal algorithm exhibits more local maxima than both raw and full binned ones. In scenario HL, where imbalance is lower, the number of local maxima declines. In order to reduce local maxima and optimize our method, we propose in the next section two possible strategies of initialization for the Bin-CL-EM algorithm.

4.2.2 Possible initialization strategies for Bin-CL-EM algorithm

In order to cope with the presence of various local maxima, we formulate new paradigms of initialization. Here, we propose two possible schema: initialization with big class parameters fixed, where initial parameters for the big class are not chosen at random and estimated from data, and marginal initialization.

Initialization with big class parameters fixed As we analyze very imbalanced data sets, we can fix the starting parameter for big class mean and variance (which can be well approximated by sample mean and variance, for example) and maintain a random initialization for the proportions and small class parameters. In Figures 4.1d-4.2d results obtained with this kind of initialization are shown for both scenario, without highlighting a significant improvement.

Marginal initialization In this initialization paradigm, we propose to run several separate univariate (small) EM on each dimension. Then, as possible rule to gather together all marginal parameters, we suggest to match parameters according to the order given by proportions. The resulting multidimensional parameter is used to initialize a (long) iteration of our Bin-CL-EM algorithm. Figures 4.1e-4.2e show results obtained with marginal initialization. These good results enhanced significantly performances of bin-marginal EM, which converges mostly towards the right maximum.

4.3 Model selection criteria

In defining mixture models we have always supposed that the number of components K was known. When no prior information about K is available, it is of primary interest to assess it. One of the main approach consists in the formulation of model selection criteria based on penalized form of model likelihood (McLachlan and Peel, 2004). The involved penalizations try to quantify the complexity of the model, typically employing functions of the number of its parameters and, thus, of the number of components in mixture models. In this section, we firstly review common criteria to assess the number of components of Gaussian mixture models with full likelihood. Then, as our estimation method is based on composite likelihood, we illustrate model selection criteria based on it. These criteria turn out to be impossible to calculate with the only knowledge of bin-marginal data, so we propose two ready-to-use heuristics to select the number of components for a bin-marginal Gaussian mixture.

4.3.1 Full likelihood model selection criteria

Usually, a general full likelihood-based choice criterion $C(K)$ to select the number of components K has this form:

$$C(K) = -2\ell(\boldsymbol{\psi}_K; \mathbf{x}) + \mathrm{PEN}(K, n), \tag{4.1}$$

where $\mathrm{PEN}(K, n)$ is a *penalization term* depending on K and n and $\boldsymbol{\psi}_K$ is the MLE for the model with K components (McLachlan and Peel, 2004). Typically, the model minimizing the criterion 4.1 is the chosen one.

AIC criterion The *Akaike Information Criterion (AIC)* was proposed in Akaike (1973). The theoretical justification of this criterion relies on the minimization of the Kullback-Leibler divergence (Kullback and Leibler, 1951) between the true distribution and the fitted model. If the fitted density is denoted by $f(\boldsymbol{x}; \boldsymbol{\psi}_K)$ and the true model density is $q(\boldsymbol{x})$, their Kullback-Leibler divergence is given by:

$$KL(q, f) = \mathbb{E}_q[\log q(\boldsymbol{x})] - \mathbb{E}_q[\log f(\boldsymbol{x}; \boldsymbol{\psi}_K)]. \tag{4.2}$$

As the first term does not depend on $f(\boldsymbol{x}; \boldsymbol{\psi}_K)$, the minimization of $KL(q, f)$ implies maximizing $\mathbb{E}_q[\log f(\boldsymbol{x}; \boldsymbol{\psi}_K)]$. AIC criterion is derived as a bias-corrected estimate of $\mathbb{E}_q[\log f(\boldsymbol{x}; \boldsymbol{\psi}_K)]$ and it is defined as:

$$\mathrm{AIC} = -2\ell(\boldsymbol{\psi}_K; \mathbf{x}) + 2I_K,$$

where I_K is the total number of parameters of the model.

The derivation of AIC relies on some regularity conditions (Cramer, 1946) that do not hold when it is needed to select the right number of components of a mixture (McLachlan and Peel, 2004). However this, AIC is still used to choose K. Furthermore, AIC tends to overestimate the correct number of components (Celeux and Soromenho, 1996).

BIC criterion The *Bayesian Information Criterion (BIC)* has been derived using a Bayesian point of view in model selection (Schwarz, 1978). Let consider a set of models $\{\mathcal{M}_K, K = K_{\min}, \ldots, K_{\max}\}$ with prior probabilities $p(\mathcal{M}_K), K = K_{\min}, \ldots, K_{\max}$. From a Bayesian perspective, the model to retain is the model maximizing the posterior probability

$$p(\mathcal{M}_K|\mathbf{x}) \propto p(\mathbf{x}|\mathcal{M}_K)p(\mathcal{M}_K),$$

where $p(\mathbf{x}|\mathcal{M}_K)$ is called *integrated likelihood* and it is equal to

$$p(\mathbf{x}|\mathcal{M}_K) = \int p(\mathbf{x}|\boldsymbol{\psi}_K, \mathcal{M}_K)p(\boldsymbol{\psi}_K|\mathcal{M}_K)d\boldsymbol{\psi}_K.$$

If the prior probabilities are the same, then the chosen model it that one with the maximum integrated likelihood. As the evaluation of integrals in integrated likelihood is difficult (Fraley and Raftery, 2002), for regular models the following approximation for the logarithm of the integrated likelihood is used:

$$\log p(\mathbf{x}|\mathcal{M}_K) \approx 2\ell(\boldsymbol{\psi}_K; \mathbf{x}) - I_K \log(n), \tag{4.3}$$

where I_K is the total number of parameters of the model. Approximation (4.3) completely defines the BIC criterion, which is

$$\text{BIC} = -2\ell(\boldsymbol{\psi}_K; \mathbf{x}) + I_K \log(n).$$

Under proper regularity conditions, BIC criterion is *consistent*, i.e. it selects the right model for n large enough. As already seen for the AIC, these conditions are not satisfied in general for mixture models (Celeux et al., 2018). However, there are some results in favour of the use of BIC. Leroux (1992) proves that BIC does not asymptotically underestimate the true number of components. Roeder and Wasserman (1997) proves that BIC is consistent if Gaussian mixtures are used to estimate a univariate density in a non-parametric way. Keribin (2000) generalizes these results by proving that, under some conditions and for an appropriate penalty term, BIC does not either asymptotically overestimate the number of components. Finally, BIC shows encouraging good results in several practical applications (Dasgupta and Raftery, 1998; Stanford and Raftery, 2000; Campbell et al., 1999).

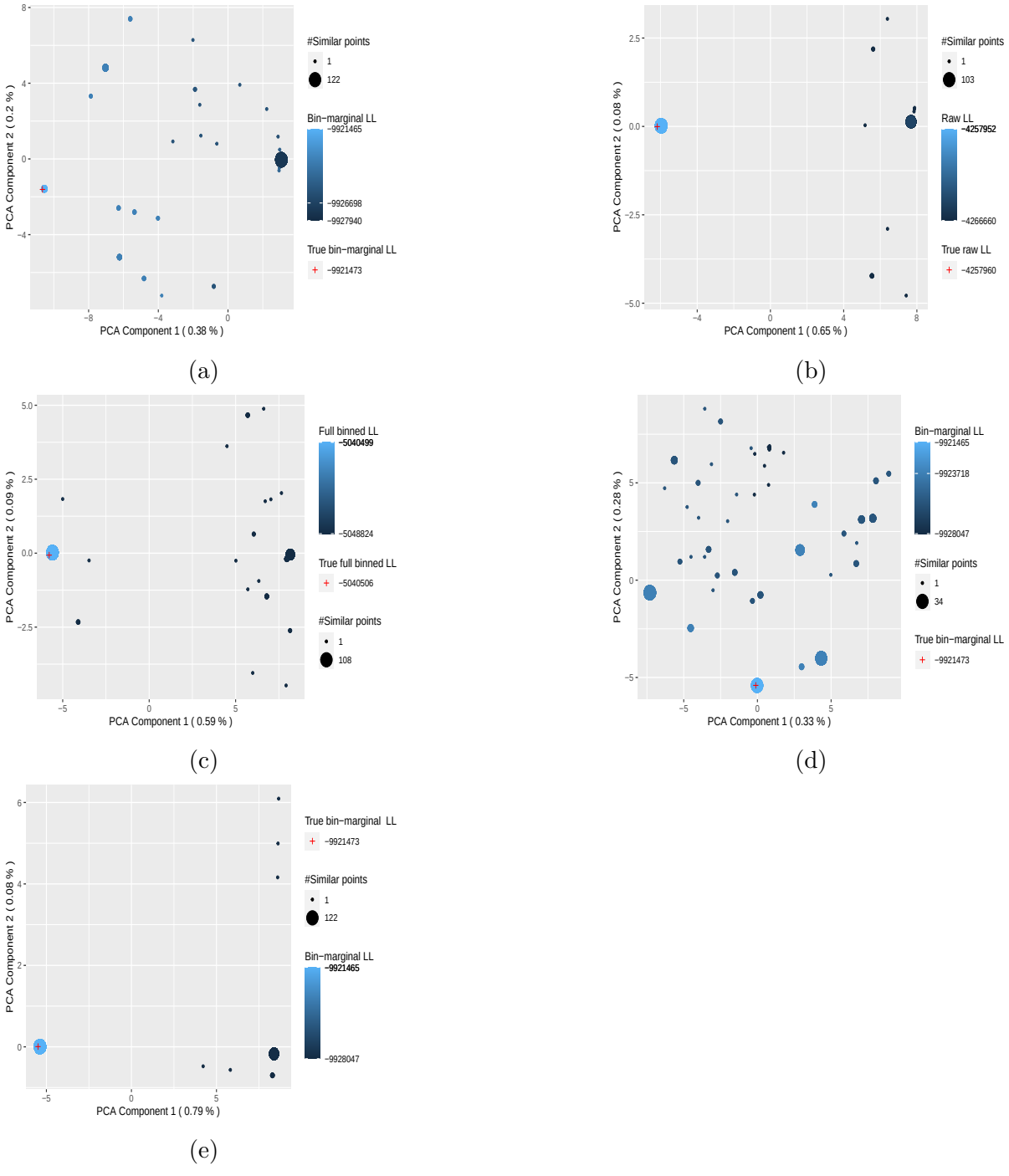


Figure 4.1: Scenario HH: PCA representation of local maxima found by: (a) Bin-marginal EM; (b) Raw EM; (c) Full Binned EM; (d) Bin-marginal EM initialized with big class given; (e) Bin-marginal EM with marginal initialization. Number between parentheses inform about the percentage of variance explained by each PCA component.

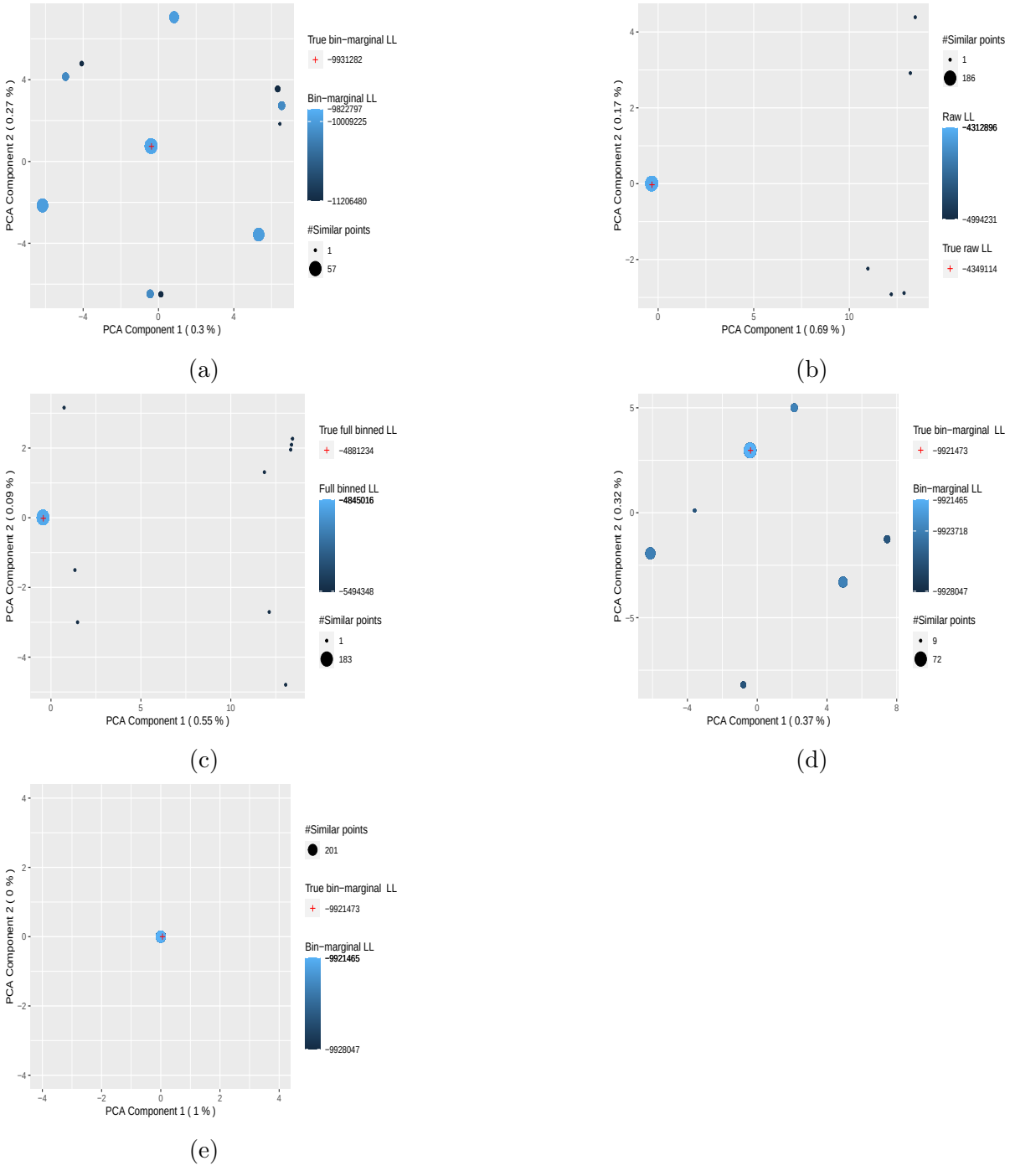


Figure 4.2: Scenario HL: PCA representation of local maxima found by: (a) Bin-marginal EM; (b) Raw EM; (c) Full Binned EM; (d) Bin-marginal EM initialized with big class given; (e) Bin-marginal EM with marginal initialization. Number between parentheses inform about the percentage of variance explained by each PCA component.

4.3.2 Composite likelihood model selection criteria

In the composite likelihood framework, authors have modified the common choice criteria replacing log-likelihood with the composite log-likelihood and redefining the penalization term. In particular, authors have replaced the number of parameters I_K with the *effective number of degrees of freedom* (Pan, 2001) $\tilde{I}_K = tr(\tilde{\boldsymbol{H}}^{-1}\tilde{\boldsymbol{J}})$, where, $\tilde{\boldsymbol{H}}$ and $\tilde{\boldsymbol{J}}$ are consistent estimates of $\mathbb{E}[\nabla^2\tilde{\ell}(\boldsymbol{\psi};\mathbf{X})]$ and $\mathbb{E}[(\nabla\tilde{\ell}(\boldsymbol{\psi};\mathbf{X}))(\nabla\tilde{\ell}(\boldsymbol{\psi};\mathbf{X}))^t]$. Thus, they are equal to:

$$\tilde{\boldsymbol{H}} = \frac{1}{n}\sum_{i=1}^{n}\nabla^2\tilde{\ell}(\boldsymbol{\psi};\boldsymbol{x}_i)$$
$$\tilde{\boldsymbol{J}} = \frac{1}{n}\sum_{i=1}^{n}(\nabla\tilde{\ell}(\boldsymbol{\psi};\boldsymbol{x}_i))(\nabla\tilde{\ell}(\boldsymbol{\psi};\boldsymbol{x}_i))^t$$

Applying these two slight modifications to the classical choice criteria and denoting with $\tilde{\boldsymbol{\psi}}_K$ the maximum composite likelihood estimate for the Gaussian mixture model with K components, Varin and Vidoni (2005) have provided the C-AIC (*Composite AIC*) criterion:

$$\text{C-AIC} = -2\tilde{\ell}(\tilde{\boldsymbol{\psi}}_K;\mathbf{x}) + 2\tilde{I}_K,$$

while Gao and Song (2010) have introduced the C-BIC (*Composite BIC*) criterion:

$$\text{C-BIC} = -2\tilde{\ell}(\tilde{\boldsymbol{\psi}}_K;\mathbf{x}) + \tilde{I}_K\log(n).$$

Both criteria have been used in Ranalli and Rocci (2016a) in order to estimate the number of components of a Gaussian mixture using a pairwise composite likelihood, showing similar good behaviours. However, the two criteria require the computation $\tilde{I}_K$ which can be difficult as pointed out in Ranalli and Rocci (2016c, 2017a).

4.3.3 Model selection criteria for bin-marginal Gaussian mixtures

These two criteria can be employed in our bin-marginal method simply using the bin-marginal composite log-likelihood $\tilde{\ell}_m(\boldsymbol{\psi};\boldsymbol{m})$ and finding the right effective number of degrees of freedom for our bin-marginal model, denoted with $\tilde{I}_{m,K}$. Similarly to $\tilde{I}_K$, we can pose $\tilde{I}_{m,K} = tr(\tilde{\boldsymbol{H}}_m^{-1}\tilde{\boldsymbol{J}}_m)$, where $\tilde{H}_m$ and $\tilde{J}_m$ are defined as:

$$\tilde{\boldsymbol{H}}_m = \frac{1}{n}\sum_{i=1}^{n}\nabla^2\tilde{\ell}_m(\boldsymbol{\psi};\mathbf{m}_i)$$
$$\tilde{\boldsymbol{J}}_m = \frac{1}{n}\sum_{i=1}^{n}(\nabla\tilde{\ell}_m(\boldsymbol{\psi};\mathbf{m}_i))(\nabla\tilde{\ell}_m(\boldsymbol{\psi};\mathbf{m}_i))^t.$$

In these definitions, each $\mathbf{m}_i$ is a realization of a $\sum_{d=1}^{D}B_d$-dimensional variable such that the generic element m_{idb_d} is equal to 1 if $a_{d(b_d-1)} \geq x_{di} < a_{db_d}$ for $d = 1,\ldots,D$, $b_d = 1,\ldots,B_d$ and

0, otherwise. We note that to calculate $\tilde{\boldsymbol{H}}_m$ and $\tilde{\boldsymbol{J}}_m$ we need to know at least the full binned observations, which is impossible if we decide to work only with marginal counts. Thus, new heuristics to choose the bin-marginal model are needed.

4.3.4 Two heuristics for the bin-marginal model

In the last section we note that present composite likelihood-based criteria are impossible to use properly with the only knowledge of bin-marginal data. To cope with this difficulty, we propose two heuristics to choose K, that we name C-BIC1 and C-BM-BIC1.

C-BIC1 Firstly, we can employ this BIC-like criterion using the same penalization term of BIC :

$$\text{C-BIC1} = -2\tilde{\ell}_m(\tilde{\boldsymbol{\psi}}_K; \boldsymbol{m}) + I_K \log(n), \tag{4.4}$$

This criterion is not new. Indeed, it was employed in Ranalli and Rocci (2016c), Ranalli and Rocci (2017a) to circumvent computational complexity of C-BIC criterion.

C-BM-BIC1 We can define another heuristic, which is specific for our bin-marginal model. It is derived as an approximation of a "true" BIC criterion based on the log-likelihood of the bin-marginal model. If we could calculate the MLE of the bin-marginal model, then we could easily define this proper BIC criterion, here denoted with BM-BIC:

$$\text{BM-BIC} = -2\ell_m(\boldsymbol{\psi}_K; \boldsymbol{m}) + I_K \log(n). \tag{4.5}$$

In the following we derive our heuristic approximating the BM-BIC criterion. Let introduce the following notations for $d = 1, \dots, D$:

$$\begin{aligned}\mathcal{F}_{\boldsymbol{m}_d} &= \{\boldsymbol{n}' : \boldsymbol{m}'_d = \boldsymbol{m}_d\},\\ \mathcal{F}_{-\boldsymbol{m}_d} &= \{\boldsymbol{n}' : \boldsymbol{m}'_d = \boldsymbol{m}_d, \boldsymbol{m}'_l \neq \boldsymbol{m}_l \ \ \forall l \neq d\}.\end{aligned}$$

It turns out that for each $d = 1, \dots, D$ the bin-marginal likelihood $L_m(\boldsymbol{\psi}_K; \boldsymbol{m})$ is equal to

$$\begin{aligned}L_m(\boldsymbol{\psi}_K; \boldsymbol{m}) &= \sum_{\boldsymbol{n}' \in \mathcal{F}_m} L(\boldsymbol{\psi}_K; \boldsymbol{n}') = \sum_{\boldsymbol{n}' \in \mathcal{F}_{\boldsymbol{m}_d}} L(\boldsymbol{\psi}_K; \boldsymbol{n}') - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\boldsymbol{\psi}_K; \boldsymbol{n}')\\ &= L(\boldsymbol{\psi}_K; \boldsymbol{m}_d) - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\boldsymbol{\psi}_K; \boldsymbol{n}'),\end{aligned}$$

where $L(\boldsymbol{\psi}_K, \boldsymbol{n}')$ is the binned likelihood for binned data $\boldsymbol{n}'$. Denoting with $\ell_m(\boldsymbol{\psi}_K; \boldsymbol{m})$ the logarithm of $L_m(\boldsymbol{\psi}_K; \boldsymbol{m})$, the previous equality can be rewritten as:

$$\ell_m(\boldsymbol{\psi}_K; \boldsymbol{m}) = \log[L(\boldsymbol{\psi}_K; \boldsymbol{m}_d) - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\boldsymbol{\psi}_K; \boldsymbol{n}')] = \ell(\boldsymbol{\psi}_k; \boldsymbol{m}_d) + \log[1 - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\boldsymbol{\psi}_k; \boldsymbol{n}')].$$

Thus, summing all D relations we obtain

$$\ell_m(\boldsymbol{\psi}_K; \boldsymbol{m}) = \frac{1}{D}\tilde{\ell}_m(\boldsymbol{\psi}_K; \boldsymbol{m}) + \frac{1}{D}\sum_{d=1}^{D} \log\Big[1 - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\boldsymbol{\psi}_K; \boldsymbol{n}')\Big].$$

Then, plugging this relation in BM-BIC and substituting $\boldsymbol{\psi}_K$ with the composite estimate $\tilde{\boldsymbol{\psi}}_K$, we obtain this new criterion:

$$\text{C-BM-BIC} = -\frac{2}{D}\tilde{\ell}_m(\tilde{\boldsymbol{\psi}}_K; \boldsymbol{m}) - \frac{2}{D}\sum_{d=1}^{D} \log\Big[1 - \sum_{\boldsymbol{n}' \in \mathcal{F}_{-\boldsymbol{m}_d}} L(\tilde{\boldsymbol{\psi}}_K; \boldsymbol{n}')\Big] + I_K \log(n).$$

Now we complete our heuristic ignoring the second term, which is really hard to compute due to the presence of several tables to calculate. This define a new possible choice criterion:

$$\text{C-BM-BIC1} = -\frac{2}{D}\tilde{\ell}_m(\tilde{\boldsymbol{\psi}}_K; \boldsymbol{m}) + I_K \log(n).$$

4.3.5 Practical experiences

In this section, we employ the two heuristics C-BIC1 and C-BM-BIC1 to select the right model in imbalanced scenarios, selected from Table 3.1. In particular, we select settings with medium (M) and low imbalance (L) (the situation with high (H) imbalance is excluded as we risk not to generate the small class with $n = 10^4$, as reported in the next paragraph), while the degree of separation can vary between very low (VL) and high (H).

Experience description For each scenario we generate 100 different data sets with n data. In order to quantify practically the consistency of the two chosen heuristics, n vary between $10^4, 10^5, 10^6$. Then, the right number of components is chosen among the set $K = \{1, 2, 3, 4\}$, according C-BIC1 and C-BM-BIC1.

Results Table 4.1 reports results for each imbalanced scenario. Despite of being two heuristics, both C-BIC1 and C-BM-BIC1 show good results, predicting in the majority of cases the right number of classes. However, in some settings with a very low degree of separation (V, VM, VL), performances do not improve with n increasing.

4.4 Impact of the binning grid

In Chapter 3, we described our bin-marginal method in order to frugally perform Gaussian model-based clustering. In our procedure, a key-role is played by the binning grids which performed a heavy data size reduction. For this reason, the refinement parameter R contribute to determine both the clustering quality and the quantity of time and memory used by the process. Thus, it is natural to quantify the impact of grid refinement on our bin-marginal approach. In this section, we show some practical examples where grids with different refinements are in action, revealing how this sort of hyper-parameter impacts clustering performance.

Table 4.1: Results obtained in imbalanced scenarios.

		C-BIC1				BM-BIC-1			
Scenario	Size	$K=1$	$K=2$	$K=3$	$K=4$	$K=1$	$K=2$	$K=3$	$K=4$
HM	10^4	-	100	-	-	-	100	-	-
	10^5	-	100	-	-	-	100	-	-
	10^6	-	100	-	-	-	100	-	-
HL	10^4	-	100	-	-	-	100	-	-
	10^5	-	100	-	-	-	100	-	-
	10^6	-	100	-	-	-	100	-	-
MM	10^4	-	100	-	-	1	99	-	-
	10^5	-	100	-	-	-	100	-	-
	10^6	-	100	-	-	-	100	-	-
ML	10^4	-	100	-	-	-	100	-	-
	10^5	-	100	-	-	-	100	-	-
	10^6	-	100	-	-	-	100	-	-
LM	10^4	22	78	-	-	90	10	-	-
	10^5		82	13	3	-	85	15	-
	10^6	-	92	8	-	-	92	8	-
LL	10^4	-	100	-	-	-	100	-	-
	10^5	-	100	-	-	-	100	-	-
	10^6	-	100	-	-	-	100	-	-
VM	10^4	100	-	-	-	100	-	-	-
	10^5	100	-	-	-	100	-	-	-
	10^6	84	16	-	-	100	-	-	-
VL	10^4	78	22	-	-	-	100	-	-
	10^5	-	82	18	-	-	100	-	-
	10^6		81	19	-	-	81	19	-

4.4.1 Description of scenarios

In this section we consider three imbalanced scenarios depicted in Table 4.2. For each of them, $n = 10^6$ data are generated by a three bivariate two-class mixtures where small class proportion π_1 varies from 10^{-2} to 10^{-4}. In all the three settings, ordered by imbalance degree, we set a high separation between the two classes in the first axis, while, in the second dimension, classes are closer. The separation by clusters is given by means (respectively, $\boldsymbol{\mu}_1 = (-4, -2)$ and $\boldsymbol{\mu}_2 = (4, 2)$), while variances are fixed to the identity matrix $\boldsymbol{I}_2$.

Table 4.2: Description of the scenarios to evaluate grid impact. Covariance matrices $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$ are equal to the identity matrix $\boldsymbol{I}_2$, $\pi_2 = 1 - \pi_1$ and $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1$.

Scenario	Small class proportion (π_1)	Small class means ($\boldsymbol{\mu}_1$)
L	10^{-2}	$(-4, -2)$
M	10^{-3}	$(-4, -2)$
H	10^{-4}	$(-4, -2)$

4.4.2 Results

Results obtained for the three scenarios are represented in Figure 4.3. We can see in all three figures that performance consistently degrades if $R_1 = 5$, a value which is very close to the limit of our condition of identifiability (Section 3.2.2). As the first marginal grid is related to the well-separated axis, we can infer that grid refinement has an impact especially on those axes where clusters are well-separated. This impact increases as the imbalance increases, while in general ARI decreases, as expected. Finally, we can note that, except for very coarse grids, there is no difference between sufficient dense grids.

4.5 Conclusion

In this chapter we have dealt with three complementary topics from an experimental point view. The conducted analyses regarded the problem of local maxima occurring in bin-marginal Gaussian mixture estimation, the choice of the right number of components in the same models and the impact of the binning grid on our method.

Concerning local maxima, we have noticed that our data-reduction technique multiplies the number of local maxima with respect to the corresponding situations when raw or full binned data are used. However, it is possible to develop promising initialization strategies providing very good results, especially with the marginal initialization. Bin-marginalization also complicates the use of information criteria to choice the number of classes K. This is because full likelihood-based criteria are not available (we use marginal composite likelihood) and composite likelihood-based criteria encounter computational difficulties if we do not know at least full binned data. So, we have employed two heuristics to have a first guide in model selection, obtaining promising

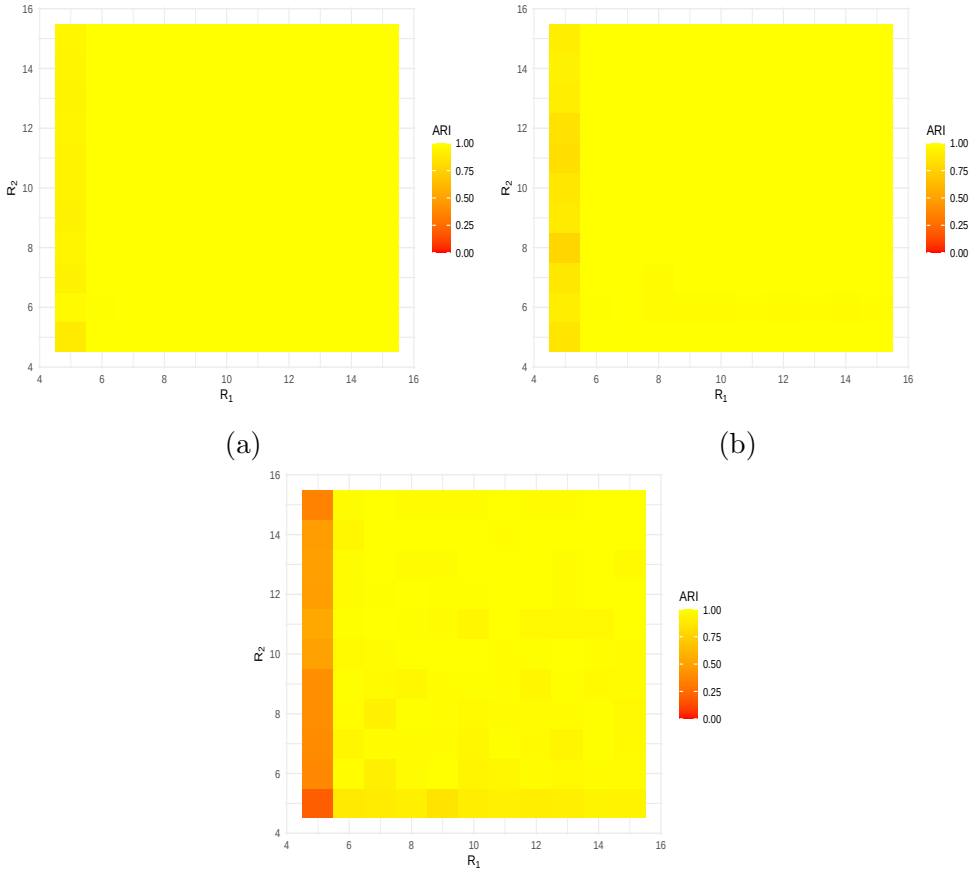


Figure 4.3: Impact of the grid refinement on clustering three simulated bivariate two-class mixtures. (a) Scenario L; (b) Scenario M; (c) Scenario H.

results. Finally, we have noticed that it is important to use finer grids in those dimensions that are more separated, while grids on badly separated axes have not a huge impact.

All the three analyses have been conducted from a fully experimental point of view in order to show a first exploration of the three related problems. We have noticed that even with bin-marginal data it is possible to deal with current issues in mixture models, such as local maxima and model choice, despite further complications caused by the extreme compression of statistical information. These first experimental results are encouraging for future research on the topic, which could be also conducted with the help of theoretical tools that have not been adopted in this work. The same tools could be useful, finally, to definitively assess mathematical rules in order to find an optimal grid, allowing a good balance between performances and computational savings.

Chapter 5

Application: anomaly detection in time series

In the previous chapters of this work we have presented a frugal method to perform Gaussian model-based clustering on huge data sets, mostly imbalanced. Up to now, we have applied our technique to continuous real data instances without any spatio-temporal relation between them. In this chapter, we aim to develop more deeply the potential of the presented work, by applying it to detect, in an unsupervised way, anomalies in time-series. This new problem is highly connected to our initial goal (frugal Gaussian clustering for huge imbalanced data sets). Indeed, the aim is always to recognize a very small suspicious data class (or several ones)class among a huge amount of normal data with unsupervised tools. What it changes here is that data instances are time-dependent.

This chapter is structured as follows: in Section 5.1, we precisely define the framework of anomaly detection, illustrating current approaches and methods. In Section 5.2, we apply our method on time-dependent data provided by the start-up DiagRAMS Technologies of Lille (website: https://diagrams-technologies.com) in two possible scenarios. In the first one, we adopt an anomaly detection point of view; in the second, we try to recognize anomalies from a clustering perspective.

5.1 Anomaly detection: background

Anomaly detection refers to the problem of finding patterns, the *anomalies*, in data, that do not conform to expected behaviour (Chandola et al., 2009). Anomalies appear in a wide variety of applications such as fraud detection (Fawcett and Provost, 1997), abnormalities detection in medical samples (Campbell and Bennett, 2000), network intrusion detection (Yeung and Chow, 2002) and industrial damage detection (Guttormsson et al., 1999). It is possible to identify various approaches to anomaly detection depending not only on the nature of data instance (for example: numerical, categorical), but also on the type of anomaly to detect and on the availability of labels distinguishing between normal and anomalous data.

5.1.1 Types of anomalies

It is possible to encounter various types of anomalies. It is important to provide a classification of anomalies according to their nature, as it changes the kind of approach to detect them. Chandola et al. (2009) propose to group anomalies into these three categories:

1. **Point anomalies:** an individual data point is considered as anomalous with respect to the rest of data. An example is depicted in Figure 5.1a, where the two points A_1 and A_2 are anomalies.

2. **Contextual anomalies:** a data point is an anomaly in a specific *context*, but not otherwise. They are common in time-series (Salvador et al., 2004) or spatial data (Kou et al., 2006), where the context is given by specific time periods or spatial regions. In Figure 5.1b an example of contextual anomaly in a time-series is shown.

3. **Collective anomalies:** in this case, the anomaly is given by the presence of a collection of data points whose occurrence together is considered as anomalous. The points belonging to the red segment highlighted in Figure 5.1c have a value which could be considered as normal relatively to the whole time-series. But, if the whole sequence of red points is considered, they constitute a collective anomaly, because this same value is repeated for an abnormal long time. Collective anomalies are distinct features of data sets where instances are related, such as time series (Warrender et al., 1999) and spatial data (Shekhar et al., 2001).

5.1.2 Approaches to anomaly detection

In this section, we aim to review and classify the principal techniques developed in scientific literature to detect anomalies. As in other machine learning problems, a first classification distinguishes between supervised, semi-supervised and unsupervised approaches, depending on the availability of *labels* denoting if a data instance is anomalous or not. Here, we briefly resume the characteristics of these three settings.

- **Supervised anomaly detection** In this case, labels distinguishing normal data from anomalies are available (Chawla et al., 2004; Joshi et al., 2001). The goal is to build a predictive model on a *train set*, whose performance in recognising anomalies and normal data is evaluated on a new set of labeled data, the *test set*. This approach is affected by all the issues related to labeling, which may be difficult especially when sample size is large

- **Semi-supervised anomaly detection** Labels are required only for normal data in semi-supervised methods (Warrender et al., 1999; Dasgupta and Nino, 2000). These data are used in the training phase to build a model describing normal behaviours. Such a model is then employed to recognize (unlabeled) anomalies in the test set.

- **Unsupervised anomaly detection** These techniques (Goldstein and Uchida, 2016; Eskin et al., 2002) do not need labeled records. It is implicitly assumed that anomalies are

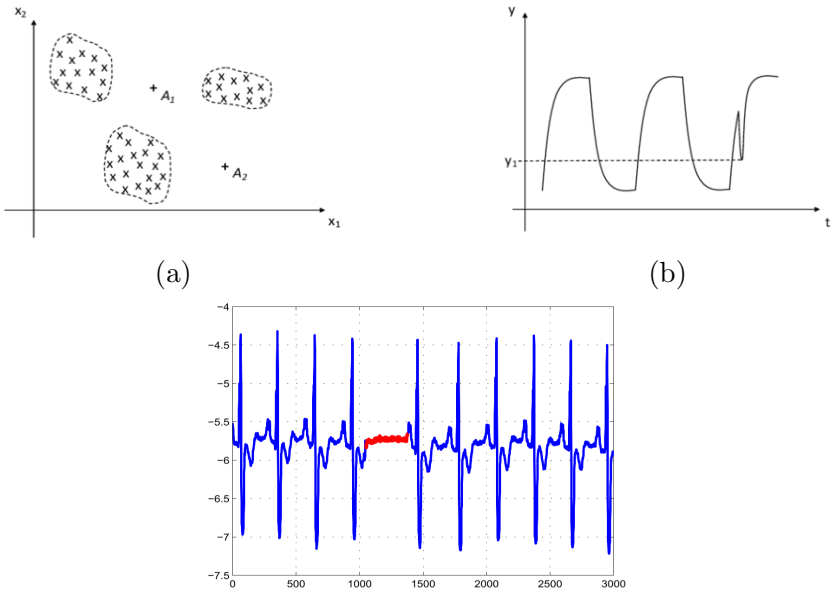


Figure 5.1: Three types of anomalies. (a) Point anomalies represented by points A_1 and A_2; (b) Contextual anomaly at level y_1; (b) Collective anomaly in a time series (in red) (Xu and Saleh, 2021; Chandola et al., 2009).

less frequent than normal instances. If this assumption is not true the methods of this class can suffer from high *false alarm rate*, i.e. they tend to detect as anomalous a normal behaviour.

It is also possible to classify the main approaches to anomaly detection according to the models and the machine learning tools involved in the analysis. Following these criteria, we can identify methods based on:

- Classification;
- Nearest Neighbors;
- Clustering;
- Statistical techniques;
- Information theoretic;
- Spectral approaches.

Before describing each of these techniques, we point out that it is possible to find in the same category both supervised and unsupervised methods.

Classification This category includes mostly supervised and semi-supervised techniques. The goal of these methods is to build a *classifier* on the train set able to recognize anomalies in the test set. In a semi-supervised setting, it is possible to distinguish *multi-class* and *one-class* classifications. In the *multi-class* case, it is supposed that the train test contains instances belonging to multiple normal classes. Then, a classifier able to distinguish between each class and the rest is built (De Stefano et al., 2000; Barbara et al., 2001). In the test phase, a point is considered as an anomaly if it is not classified as normal by any of the classifiers. In *one-class* classification, it is assumed that all training data points belong to the same class. The goal of these techniques is to learn a boundary around normal points able to discriminate whether a test instance is anomalous or not (Schölkopf et al., 2001; Roth, 2004). The classifiers can also be build with different machine learning tools, such as neural networks (Ghosh and Reilly, 1994; Augusteijn and Folkert, 2002) and support vector machines (King et al., 2002),(Davy and Godsill, 2002). Association rules have also been used to detect anomalies in an unsupervised way (Agrawal and Srikant, 1995).

Nearest Neighbors In a Nearest Neighbors-based technique a point is classified as anomalous if it is too far from its closest neighbors or if its neighborhood is sparse. Thus, it is possible to define two classes of methods to detect anomalies: *distance-based* and *density-based*. For the first class of techniques, the definition of a distance between data points is required. Then, the anomalies can correspond to, for example, those points which are far from the k-th neighbors (Byers and Raftery, 1998), or those data whose sum of distances from the first k neighbors is too high (Eskin et al., 2002). Several variants of distance-based method can be defined in order to handle various data types, such as categorical data (Wei et al., 2003) or spatial data (Kou et al., 2006). Density-based approaches requires the definition of a measure to quantify the density of a point neighborhood. These methods can be misleading if there are regions of data with different densities. For this reason, current density-based techniques employ *local* versions of data density (Breunig et al., 2000; Tang et al., 2002).

Clustering In this approach, clustering is used to detect anomalies principally in an unsupervised fashion according to three different criteria. In the first category of clustering-based methods, anomalies are points which are not assigned to any cluster. This requires clustering algorithms which do not force all the points to be part of a cluster, as DBSCAN (Ester et al., 1996) or ROCK (Guha et al., 2000). Another way consists in considering as anomalous those points which are too far from from the nearest cluster centroid (Smith et al., 2002; Labib and Vemuri, 2002), which is the mean of all points belonging to the same cluster. So, in this case, the analysis is conducted in two steps: in the first one, data are grouped into clusters and in the second phase, points are classified according their distance from the nearest centroid. Finally, it is also possible to detect anomalies assuming that they belong to very sparse or small clusters (Pires and Santos-Pereira, 2005; Otey et al., 2003).

Statistical techniques The methods belonging to this group of strategies share the idea of fitting a statistical model to normal data and, then, using a statistical test to determine if an unseen instance is anomalous or not. Both parametric and non-parametric techniques can be

applied to build such statistical model, depending on a-priori assumptions regarding its shape. Simple assumptions of Gaussianity for normal behaviour model can be used with the definition of specific tests (Barnett and Lewis, 1984; Grubbs, 1969). More complex models used in this context are regression models (especially for time-series (Abraham and Box, 1979; Abraham and Chuang, 1989)) and mixture models (Spence et al., 2001; Reddy et al., 2017). Regarding non-parametric techniques to fit the normal data model, popular choices are histograms (Eskin, 2000; Endler, 1998) and kernel functions (Yeung and Chow, 2002; Čampulová et al., 2018; Holešovskỳ et al., 2018).

Information theoretic The methods belonging to this category are based on measures quantifying the amount of *information content* of a data set, such as *Kolmogorov Complexity* (Li et al., 2008) and entropy (Shannon, 1948). They consist in finding a minimal subset of data instances such that the difference between the information of the whole data set and the information of the data set without those points is maximum. This minimal subset is considered as anomalous. Kolmogorov Complexity is used in several techniques as principal information measure (Keogh et al., 2004; Arning et al., 1996), while entropy is often used in categorical data applications (Lee and Xiang, 2000; Ando, 2007). Information theoretic has also been applied to detect anomalies in more complex data structure, such as time-series (Lin et al., 2005), spatial data (Lin and Brown, 2006) and graph (Noble and Cook, 2003).

Spectral approaches These techniques assume that it exists a subspace of the original sample space where anomalies can be easily identified. Typically, these interesting subspaces are found by using spectral methods which provide embeddings or projections of the initial data set. A common technique used in this context is *Principal Component Analysis (PCA)* (Parra et al., 1996; Dutta et al., 2007), where anomalies correspond to those points having large values on components with low explained variance. Further methods of data projection have been also employed to detect anomalies in graph time-series (Sun et al., 2004), intrusion domain (Shyu et al., 2003) and in space craft components (Fujimaki et al., 2005).

5.2 Detecting anomalies with bin-marginal Gaussian clustering

In this section, we employ our bin-marginal method to detect anomalies in time-series. Data are provided by DiagRAMS Technologies, a french startup working on the domain of anomaly detection in industrial processes. The analyses of this section give a first answer to the needs of DiagRAMS of developing a tool able to:

- Easily manage huge data sets. In their domain, it is possible to collect long time series (months) where data appears with a high frequency rate (1 data/ms). Thus, methods that can easily deal with millions of instances are required.

- Detect anomalies in an unsupervised way. Actually, labels for supervised detection are often not provided and manual labeling is a very difficult and long process. Therefore, an unsupervised method is needed.

Our method satisfies both requirements even if it has been built in a time-independent setting. Actually, we do not apply our technique to the raw time-series, but on additional "static" data sets containing useful synthetic information about original time-dependent data. Further details are given in Section 5.2.1. We point out that these synthetic data sets have the same characteristic of those described in previous chapters (huge imbalanced data sets), where our bin-marginal technique can be applied to perform a Gaussian clustering and compared with two competitors, subsampling and full data set analysis. The results of all analyses and comparisons are contained in Section 5.2.2 and Section 5.2.3.

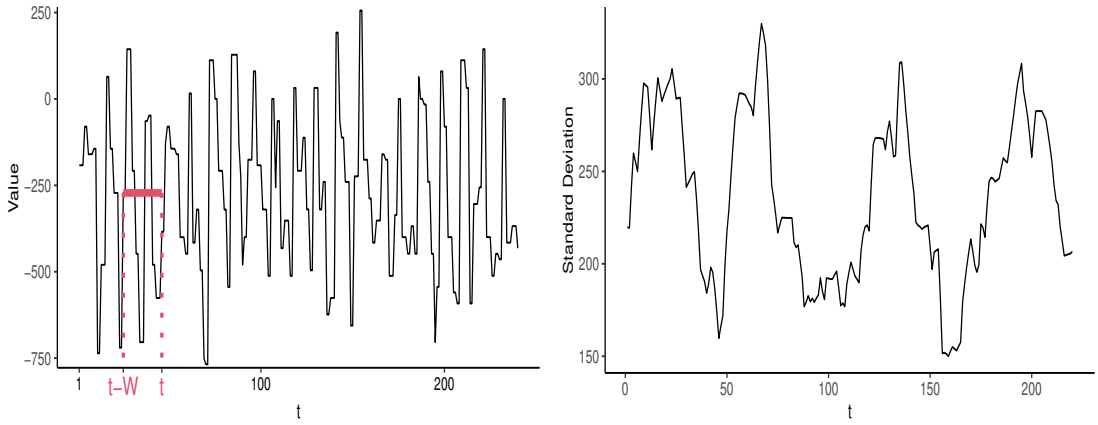
5.2.1 Context

All data analyzed in this chapter are generated by a test-bed machine owned by DiagRAMS. This machine allows to generate time series with reliable anomaly patterns that can be found in any industrial process. Indeed, the test-bed is employed by DiagRAMS to test new data science methods to detect anomalies and to show the performances of current techniques during expositions. This test-bed, built by ICAM (Institut Catholique d'Arts et Métiers de Lille), is composed by an engine and a brake, connected by a band. It is also equipped with an accelerometer and a thermometer, which register, respectively, the speed and the temperature of the system. There is also a module collecting data and extracting synthetic information about the current system of the machine. These statistics are the object of our analyses.

The test-bed provides three time-series representing the speeds of the system along the three-dimensional spatial axes. More important, it also provides additional data regarding the standard deviation of data contained inside *sliding windows* of length W. At each instant t, the standard deviation of data between the instant t and $t - W$ is calculated. In Figure 5.2 an example is depicted for a simple time-series with 240 instances (Figure 5.2a). In this figure, the window is highlighted in red and the resulting standard deviation time-series is represented in Figure 5.2b. In our analyses, we consider the data set containing the standard deviations of the three speeds. Thus, if the original time-series has n data, we analyze a three-dimensional data set with $n - W$ instances.

We conduct these two different analyses on two different time series data:

1. In the first scenario, we consider a *train set* where there are two classes of normal behaviour and anomalies are absent. One of the two classes is much smaller than the other. The goal of the experiment is to estimate a two-classes mixture on the train set, estimating, thus, a parametric density denoted by $f(\cdot, \hat{\boldsymbol{\psi}})$. Then, this model of normal behaviour is tested on a *test set*, where a third anomalous class in present. In the test phase, anomalies will be those points $\boldsymbol{x}$ in the test set whose estimated density with respect to the normal model, i.e., $f(\boldsymbol{x}, \hat{\boldsymbol{\psi}})$, is lower than a certain threshold. This is an authentic anomaly detection task.



(a) (b)

Figure 5.2: Illustration of *sliding windows* processing. (a) A window of size W (in red) is fixed for each instant t to compute the standard deviation of all points of the original time-series between $t - W$ and t. (b) The resulting standard deviation time-series.

2. In the second experiment we consider a single data set containing two normal classes and a third very rare anomalous class. The aim is to recover a third-class partition which distinguishes anomalies from normal instances. This is more a clustering-like task.

5.2.2 First scenario

Data In the training phase of the first scenario we consider a three-dimensional time-series with $n = 202,002$ instances and two classes of normal behaviour, where test-bed engine works at capacity 1,000 and 3,000 (a very rare, but normal, behaviour). This case corresponds to a practical situation where a machine normally works at a fixed capacity (here 1,000), but, in very few cases, it can work at a higher speed (here 3,000) due to intense tasks to accomplish. A possible anomaly could be, for example, working at an intermediate capacity due to an internal disequilibrium or damages. This is the case of our test time-series ($n = 23,002$) where, in addition to the two normal classes, there is an anomalous condition where machine works at capacity 2,000. Figures 5.3a-5.3c depict the three speeds of the system on axes X, Y, Z for the train time-series. The central peak represents the small normal class (capacity 3,000). In Figures 5.3d-5.3f instances of the small class (the central peak) are zoomed. The time-series used as test set has 3,000 instances having or one of the two normal behaviours (capacity 1,000 or 3,000) or an anomalous one (capacity 2,000). Here, anomalies are at the beginning of the time-series, as depicted in Figure 5.5.

As specified in Section 5.2.1, we use for our analyses a $(n - W) \times 3$ additional data set (both for train and test sets) containing variances inside each sliding window of length W. As we specified 15 possible values of W varying between 10 and 150, we analyze 15 different data sets. As example, in Figure 5.4 and in Figure 5.6 standard deviations obtained for $W = 20$ are

depicted for train and test set, respectively. To simplify the analyses, we train and test on data sets with the same value of W.

Methods For each data set corresponding to a fixed value of W, we fit a two-classes Gaussian mixture on the train data using our bin-marginal method, EM with subsampling and a full data EM. The bin-marginal technique uses a grid refinement $R = 100$ and, hence, subsampling is conducted with a subsample of size $2R$ in order to use the same memory space of our proposal. To evaluate its variability, subsampling performances are evaluated on 100 different subsamples. Once fitted the two-classes Gaussian mixture, we fix a threshold and we classify as anomalous all points $\boldsymbol{x}$ in the test set whose estimated density with respect to the train model $f(\boldsymbol{x}, \hat{\boldsymbol{\psi}})$ is below this threshold. When labels are available in the train phase (here we use labels only for final comparisons), it is possible to select a optimal threshold (acting like a hyperparameter) with the use of a third *validation set* (Hastie et al., 2009). Alternatively, one can a-priori fix the threshold to a certain quantile α of the distribution of test points estimated densities, making the assumption that in the test set there is a percentage α of anomalies. Here, we choose this last strategy and we use for α different values between 0.01 and 0.13. Once obtained such a classification, this is then compared with the true labels and its goodness is measured by the ARI score (Hubert and Arabie, 1985).

Results and discussion The results for the first scenario are depicted in Figure 5.7. We note that the best results are obtained when window size W is equal to 20 and threshold α is 0.09. For lower and higher values of the threshold, results deteriorate, as there is an high number of false anomalies and false normal points, respectively. The rate of false anomalies (false positive error) and the rate of false normal points (false negative error) for threshold 0.03, 0.07 and 0.13 are represented in Figures 5.8. This confirms that a low threshold increases the rate of false normal points, while a too high threshold has a small false negative error, but also a higher false positive rate. We note that false positive rate is high when the threshold is equal to 0.03. This is surprising as we would expect a very low value. This is because of *transitions*, i.e., the points where the system passes from two different speeds. It turns out that these points are detected as anomalies, while they are labeled as normal. Actually, there is a delay between the time at which test-bed reports a speed and the time at which the system actually works at that speed. This fact causes an initial wrong labeling which has an impact of our analyses and also explain why bin-marginal method outperforms full data set EM. While subsampling was expected to be worse than our proposal, the results reported by full EM are surprising. In Figure 5.9 we give an explanation to these phenomena. In this figure, we represent the anomalies (red points) and normal instances (black points) estimated by bin-marginal method and full EM for thresholds 0.03 (Figures 5.9a-5.9b), 0.07 (Figures 5.9c-5.9d) and 0.13 (Figures 5.9e-5.9f). From the analysis of these six results, we conclude that transitions are prone to be detected as anomalies enhancing classification errors. In addition, full EM is not able to find the small normal class being influenced by transitions and this explains why it performs worse than bin-marginal. The transition management can be an interesting starting point for future research in order to improve the proposed method.

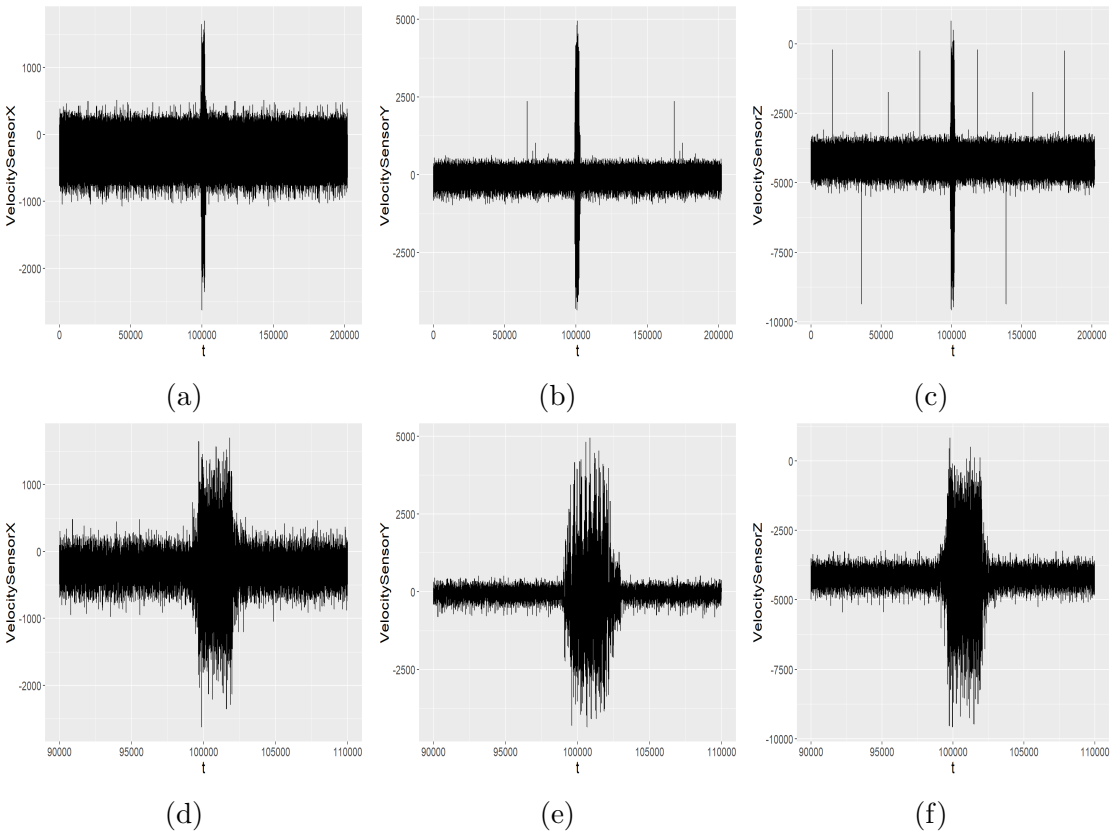
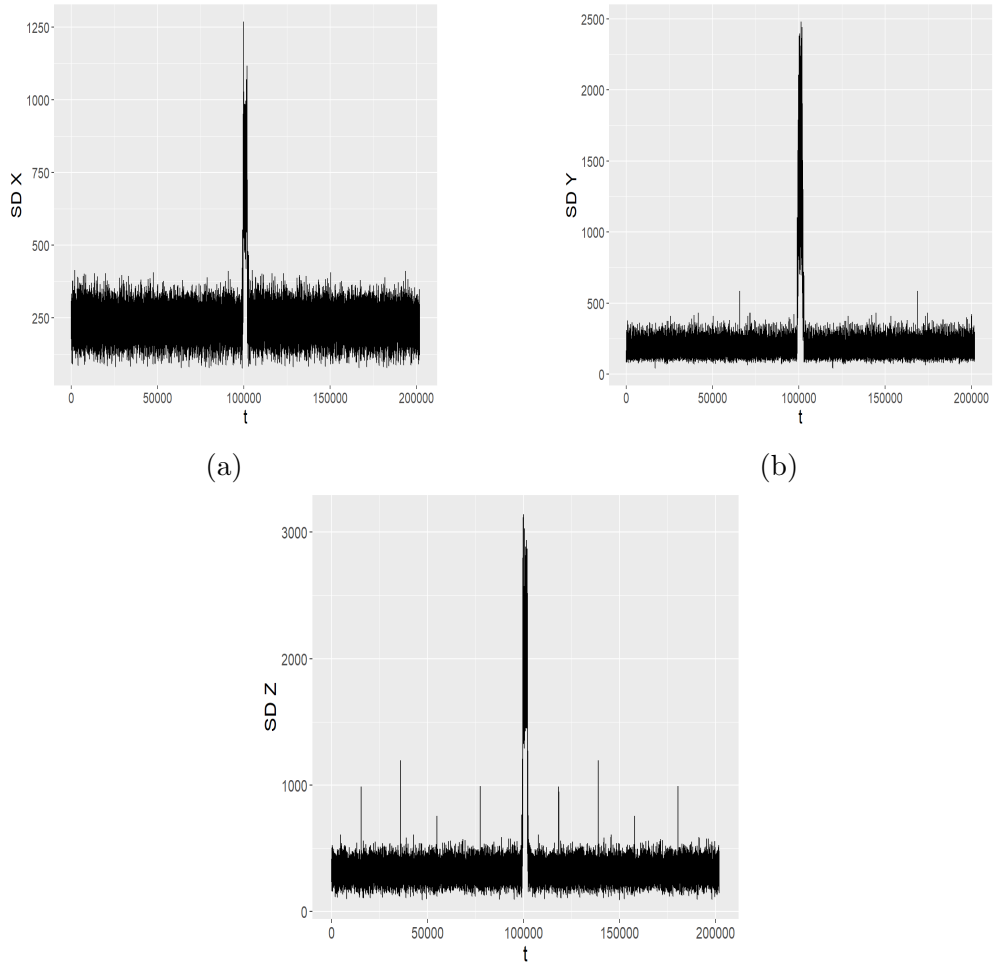


Figure 5.3: First scenario: train time-series. (a) Speed axis X (b) Speed axis Y (c) Speed axis Z. (d)-(f) Zoom around the anomalous class.



(a) (b)

(c)

Figure 5.4: First scenario: train data set containing standard deviations when $W = 20$. (a) axis X (b) axis Y (c) axis Z.

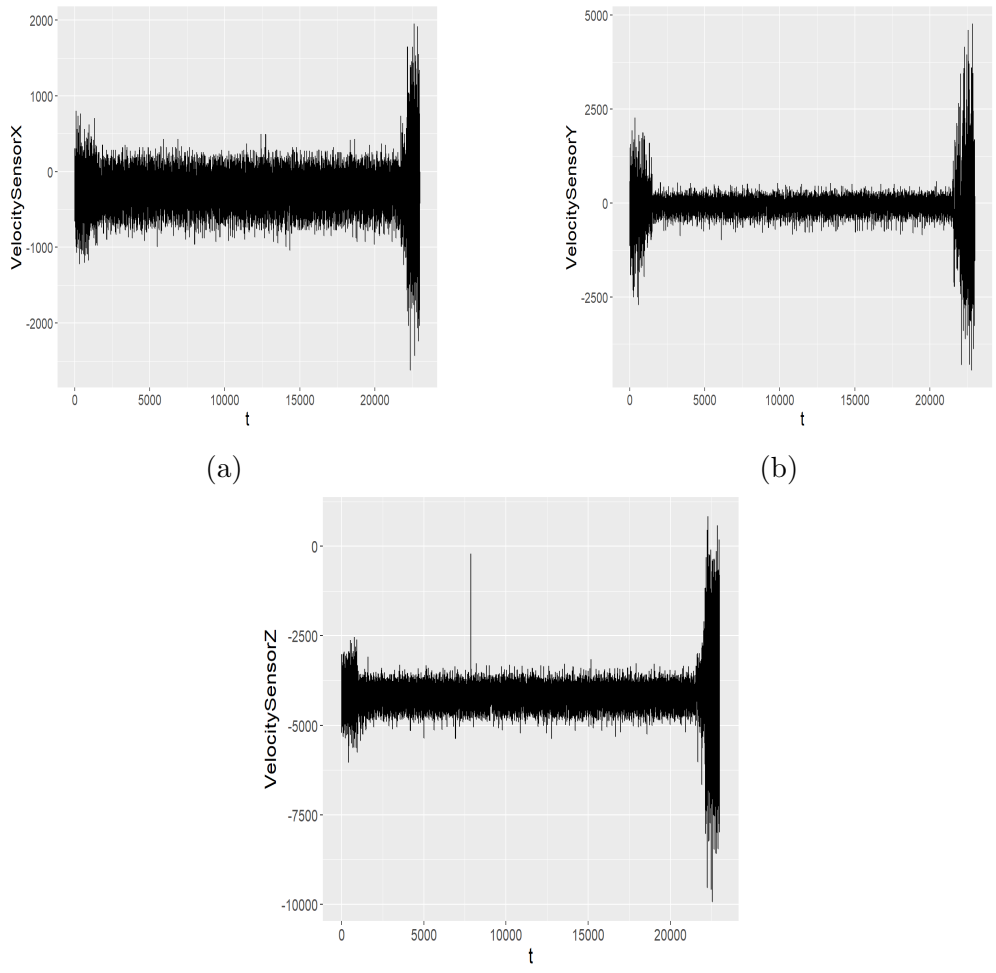
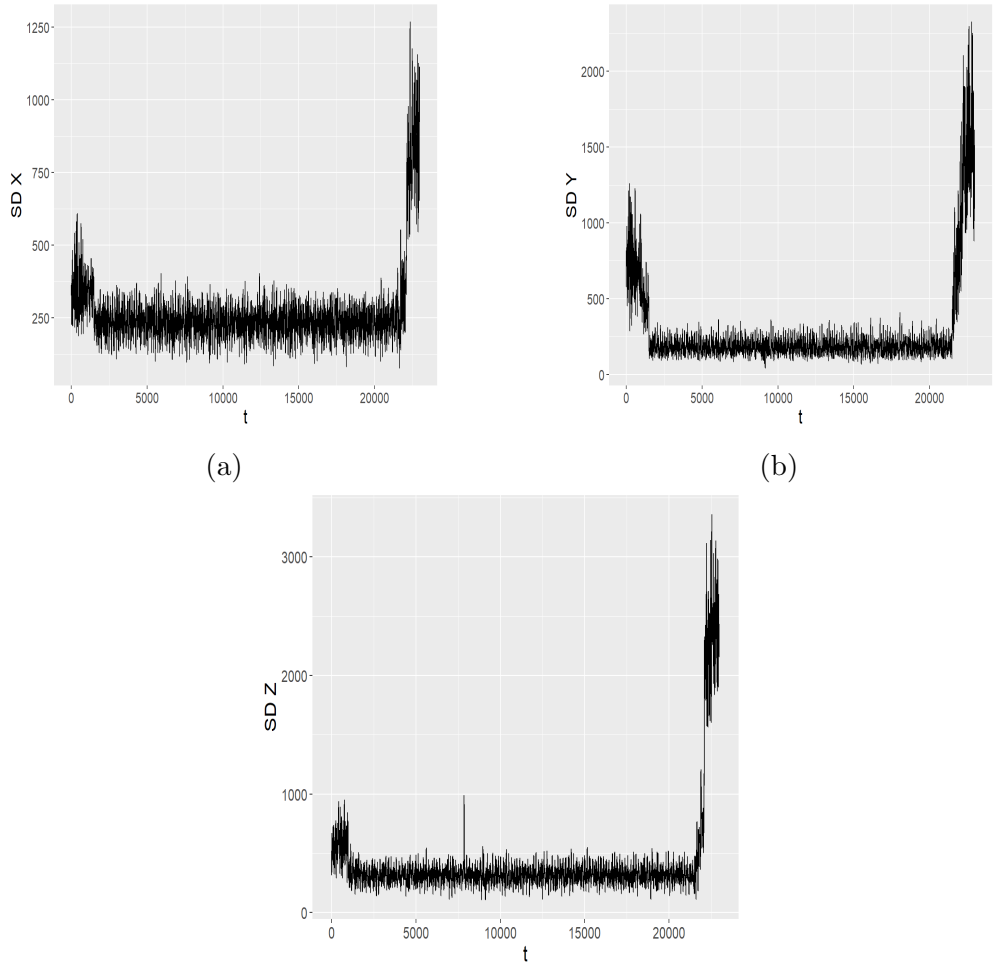


Figure 5.5: First scenario: test time-series. (a) Speed axis X (b) Speed axis Y (c) Speed axis Z.



(a) (b)

(c)

Figure 5.6: First scenario: test data set containing standard deviations when $W = 20$. (a) axis X (b) axis Y (c) axis Z.

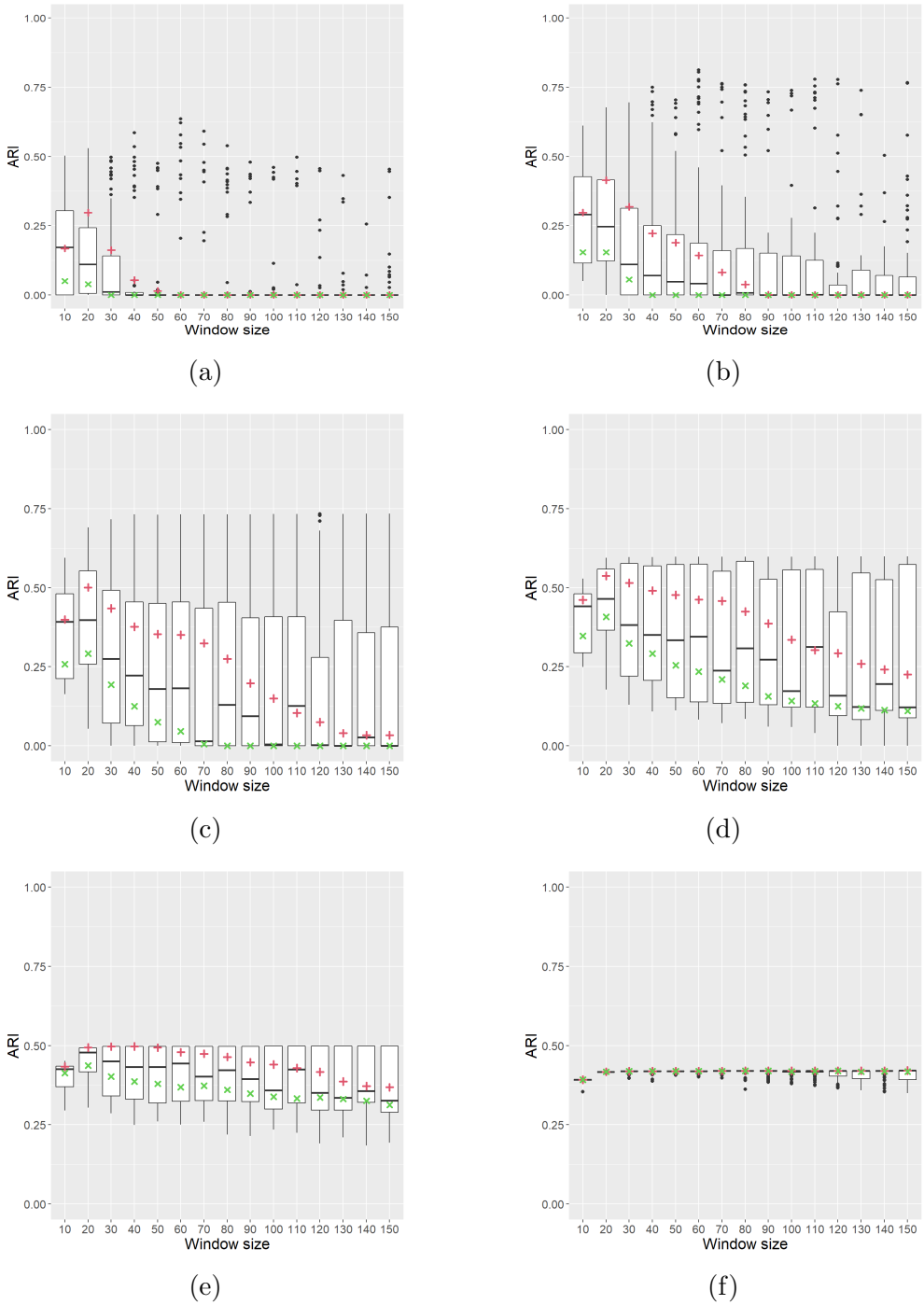


Figure 5.7: First scenario results: ARI in function of the window size for a fixed threshold for bin-marginal CL-EM (red crosses), subsampled EM (black boxplots) and full data EM (green crosses). (a) Threshold 0.03. (b) Threshold 0.05. (c) Threshold 0.07. (d) Threshold 0.09. (e) Threshold 0.11. (f) Threshold 0.13.

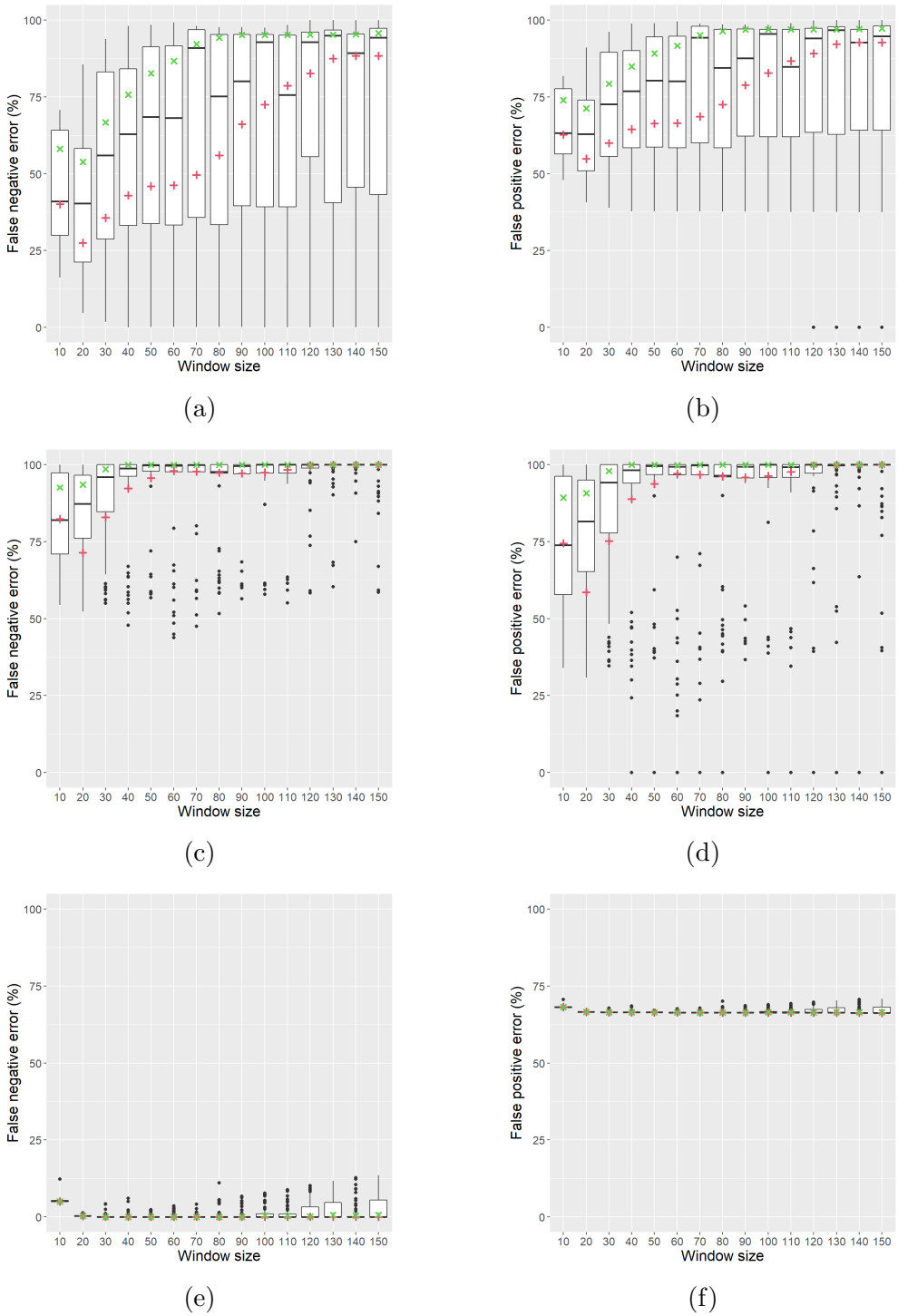
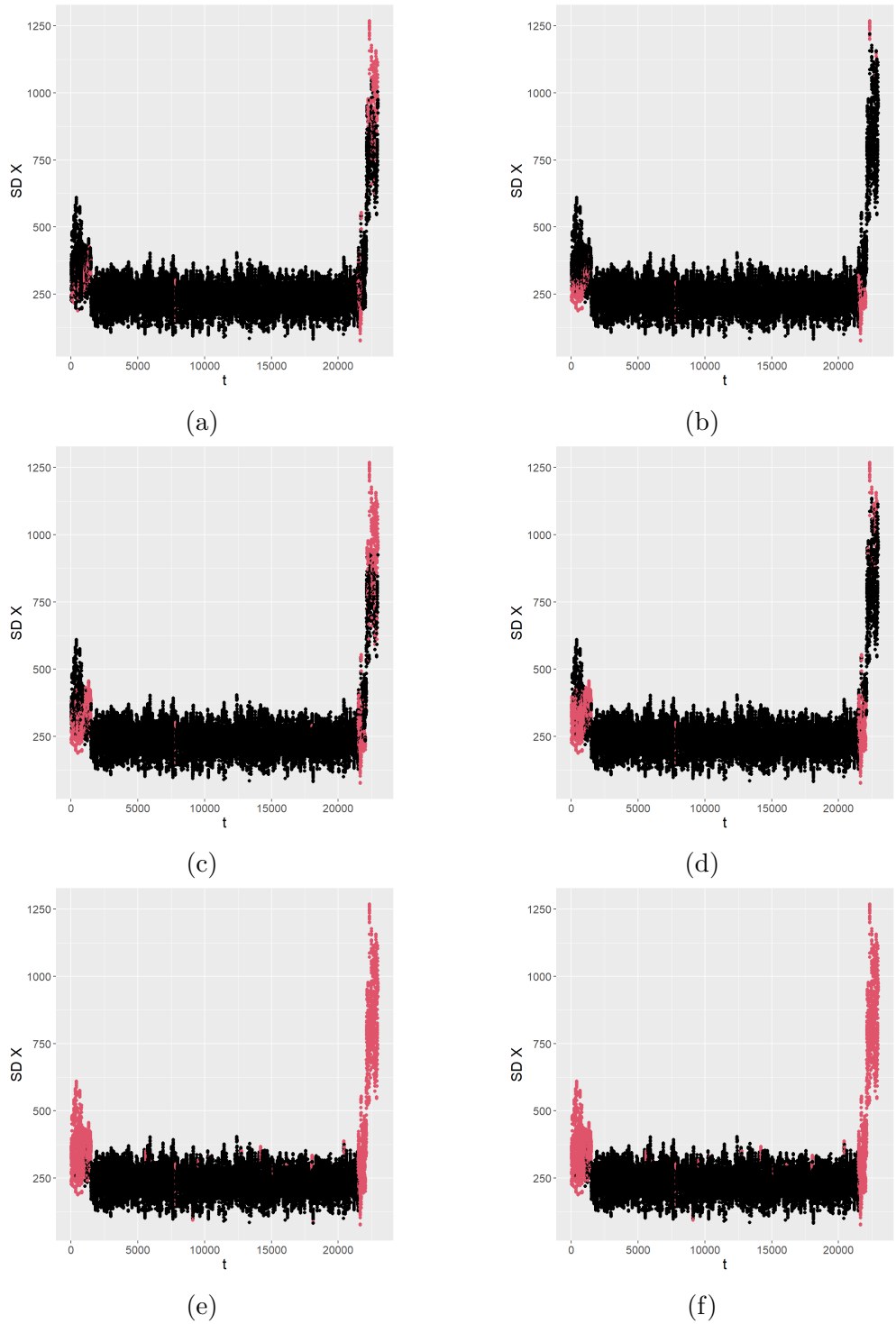


Figure 5.8: First scenario: false negative and false positive error for different thresholds and window sizes for bin-marginal CL-EM (red crosses), subsampled EM (black boxplots) and full data EM (green crosses). (a)-(b) Threshold 0.03. (c)-(d) Threshold 0.07. (e)-(f) Threshold 0.13.



(a) (b) (c) (d) (e) (f)

Figure 5.9: First scenario: transitions study on standard deviation data set when $W = 20$. (a) Full EM classification at threshold 0.03. (b) Bin-marginal classification at threshold 0.03. (c) Full EM classification at threshold 0.07. (d) Bin-marginal classification at threshold 0.07. (e) Full EM classification at threshold 0.13. (f) Bin-marginal classification at threshold 0.13.

5.2.3 Second scenario

Data In the second scenario, we consider four time-series with 315,502 instances and three classes: two normal ones (capacity 1,000 and 2,000, respectively) and an anomalous one (capacity 3,000). The difference between the four time-series is given by the proportion of the small anomalous class which can be equal to $0.0005, 0.001, 0, 005, 0.01$. As example, the three speeds at dimension X, Y, Z for the forth time series (small class proportion equal to 0.01) are represented in Figures 5.10a-5.10c. In Figures 5.10d-5.10f we zoom observations corresponding to capacity 2,000 and 3,000. This setting reproduces a practical circumstance where a system can normally work at two different capacities (low and intense states, here 1,000 and 2,000), and an example of anomaly is working at an unexpected and too high capacity (here 3,000). As for the first scenario, we consider the additional data sets containing the three variances inside sliding windows of length W and we consider the same values of W used in Section 5.2.2. In Figure 5.11 the standard deviations obtained for $W = 20$ are presented.

Methods For each data set corresponding to a fixed values of W, we fit a three-classes Gaussian mixture on the given data using our proposed bin-marginal method, EM with subsampling and full data EM. We use the same settings of Section 5.2.2 for the grid refinement ($R = 100$), the subsample size (equal to $2R = 200$) and the number of different subsamples (100). In this scenario, for each method and setting given by the window length W, we use the fitted model to obtain a partition through the MAP rule. This partition is then compared to the true one and its quality is measured by the ARI score (Hubert and Arabie, 1985).

Results Figure 5.12 reports the results obtained in this second scenario. We note that bin-marginal method (red crosses) is competitive with full data EM (green crosses) and subsampling (black boxplots). Performances depend on the window size W and we can obtain very bad results if W is too low, especially with full EM. This is because small windows can not separate well standard deviations of the three classes. If W is too high, results degrade, as the variances captured by a too large window may be equal to the variance of the whole sample. In general, there is an improvement in performance when the small class proportion increases. In addition, we note that box-plots corresponding to subsampling are very tight or they do not exist. In fact, subsampling produces an high percentage of failures, as represented in Figure 5.13. If the small class proportion increases, this percentage decreases, confirming results of Chapter 3.

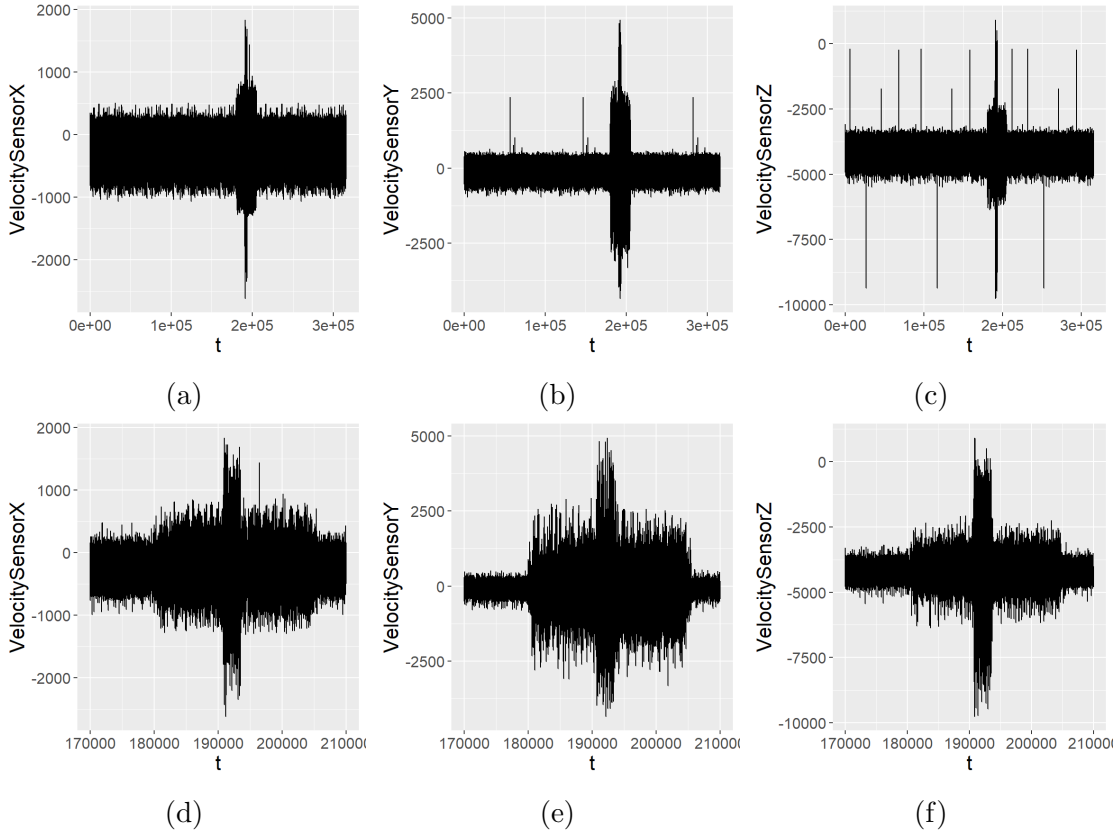
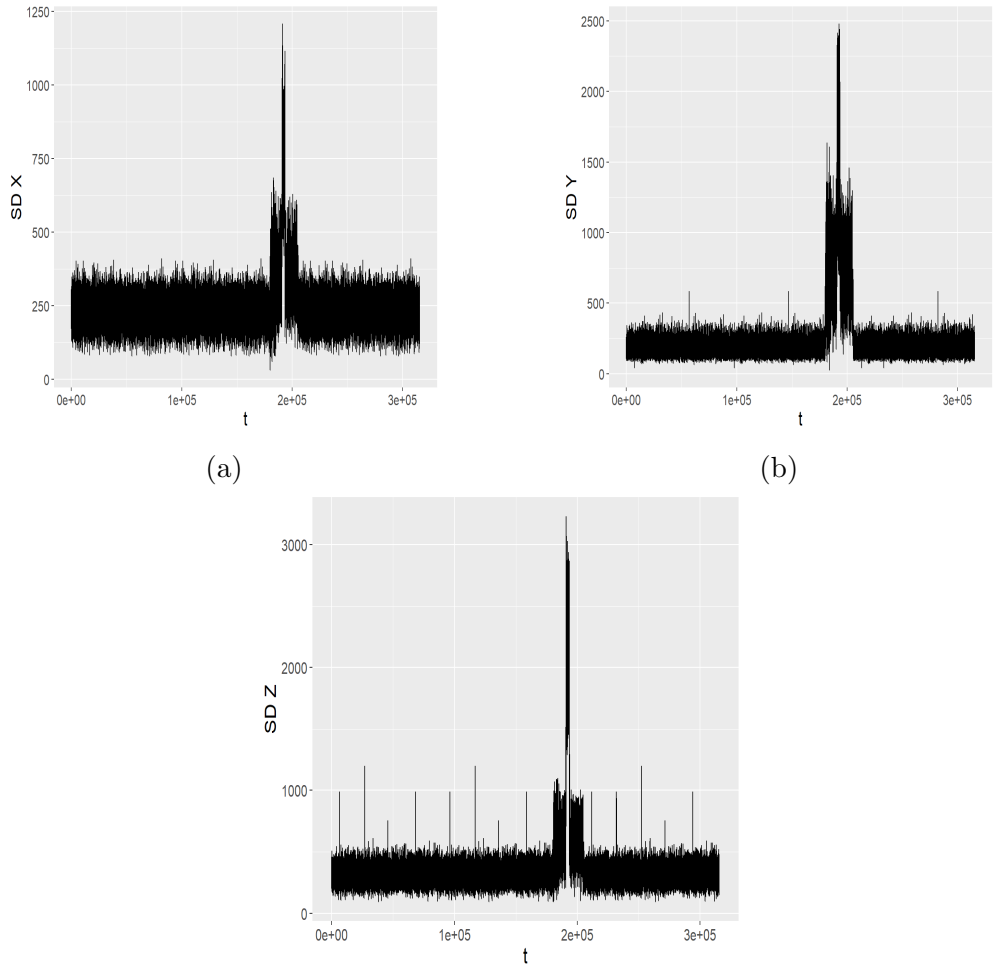


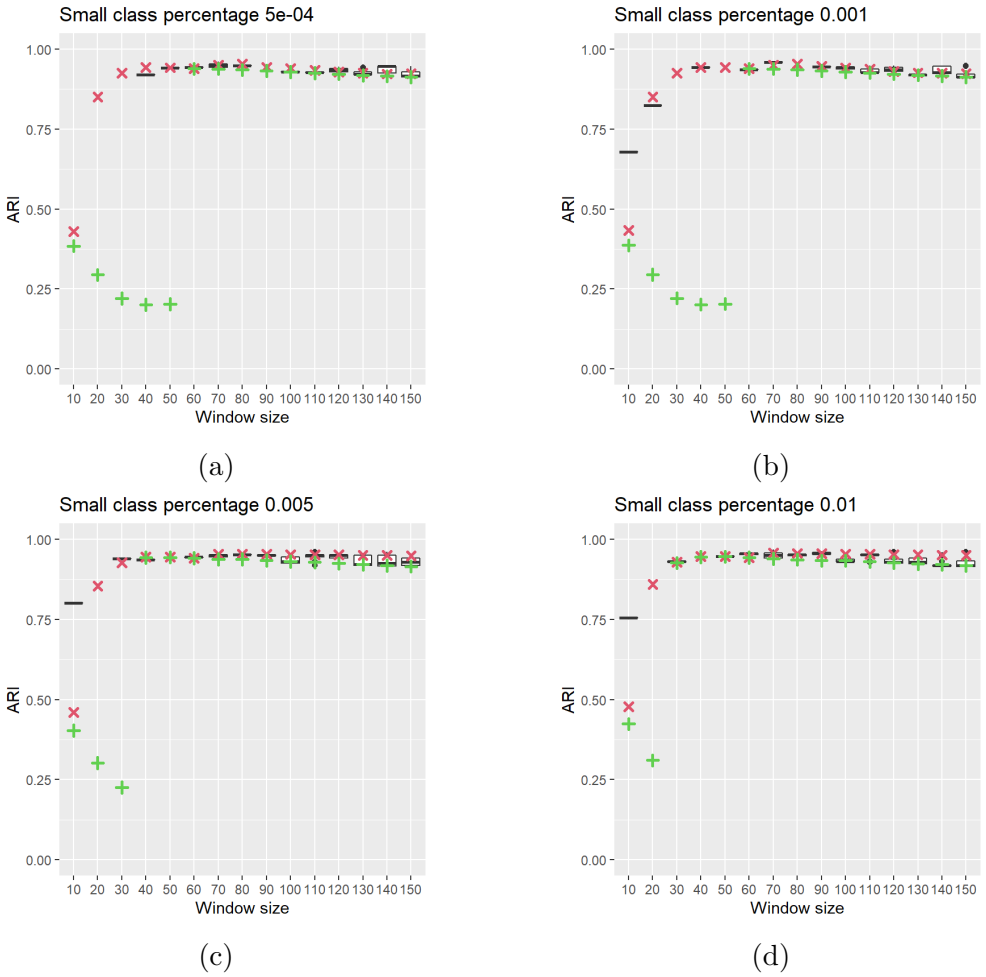
Figure 5.10: Second scenario: time-series for small class proportion equal to 0.0005. (a) Speed axis X. (b) Speed axis Y. (c) Speed axis Z. (d)-(f) Zoom around points with capacity 2,000 and 3,000.



(a) (b)

(c)

Figure 5.11: Second scenario: data set containing standard deviations when $W = 20$. (a) axis X (b) axis Y (c) axis Z.



(a) (b) (c) (d)

Figure 5.12: Second scenario: ARI in function of the window size for time-series for bin-marginal CL-EM (red crosses), subsampled EM (black boxplots) and full data EM (green crosses). Small class proportion equal to (a) 0.0005. (b) 0.001. (c) 0.005. (d) 0.01.

Figure 5.13: Second scenario: percentage of subsampled EM failures in function of small class proportion and window size.

5.3 Conclusion

In this chapter we have tested the bin-marginal method in a real problem with time-series, extending the range of potential applications of the technique to time-dependent data. In particular, bin-marginal Gaussian clustering has been employed to detect anomalies in time-series provided by a test-bed machine own by the start-up DiagRAMS Technologies of Lille. Anomaly detection turned out to be an application field that fits well with the main aims of our methods, as it is requested to recognize a very tiny, but anomalous, class between a huge amount of normal instances. Moreover, our method has encountered industrial interest, as it allows a frugal and unsupervised approach to this problem, avoiding issues related to labeling and data size.

In this application, our proposal has been applied on additional data sets providing information about data variance inside fixed intervals of time. In this way, we could use the method on data similar to those of previous chapters, comparing it to subsampling and full data EM and obtaining good results. In our perspective, we aim to apply our method directly on time-series data, avoiding the additional data sets. We have also noticed that our technique can be influenced by transitions. A further axis of research could concern possible methods to recognize transitions and improve the analysis. Actually, transitions detection could be more powerful as, in our case, they correspond to the temporal instants before anomalies. Thus, they can be identified as possible symptoms of incoming anomalous behaviours, providing important information for timely interventions on the machine.

Chapter 6

Conclusions and perspectives

6.1 Summary of the thesis

In this thesis we developed a frugal method to perform Gaussian model-based clustering on huge imbalanced data sets under conditions of limited computational resources (storage and time). This technique is principally based on a bin-marginal data reduction that allows sensible storage and time savings, which are also made possible by a model estimation based on composite likelihood theory.

In our first contribution (Chapter 2), we introduced the idea of binning univariate data through a binning grid to reduce storage and time consumption. In a very simple setting, where data were generated by a single Gaussian with mean unknown, it was possible to show some theoretical properties of the binned maximum likelihood estimator (regarding bias and variance) and we defined a criterion to choice an optimal grid reducing the variance of the corresponding estimator. We proved identifiability for binned univariate Gaussian mixture with K classes as the main theoretical result of this contribution. Then, a binned EM algorithm was applied to estimate univariate Gaussian mixture models on simulated data, comparing it to subsampling, which, in our initial review, had turned out to be a popular method to make current clustering methods frugal. We noted that our proposal allowed to well estimate the underlying mixture, while subsampling needed large samples to obtain similar results. Moreover, this practical experience confirmed remarkable savings in terms of time and memory.

In Chapter 3, we presented our main contribution, consisting in an extension of the previous univariate method to a multivariate Gaussian setting. We noticed that a trivial use of multivariate binned data were not feasible, as D-dimensional binning grids could produce an amount of binned data difficult to store and analyze, even if D is moderate ($D > 2$). As a first solution to this issue, we proposed to employ only the vector of marginal counts, managing in obtaining a more manageable data collection. The use of marginal counts involved the definition of new model to estimate, that we have called *bin-marginal* mixture model. Then, we defined an EM algorithm maximizing its likelihood to estimate it. Actually, this full-likelihood EM algorithm revealed to be numerically intractable and, thus, we decided to estimate bin-marginal Gaussian mixtures with a composite likelihood approach. Using both marginal counts and composite likelihood, we defined a bin-marginal composite likelihood EM algorithm (bin-CL-EM). The proposed algorithm was applied on both simulate and real data, in comparison with an EM

with subsampling and a full data EM. Results confirmed that our proposal outperformed both competitors. In particular, under the same memory constraints, bin-marginal technique outperforms subsampling in clustering quality (measured by ARI score), being also faster and more frugal than full EM. In real data analyses, we employed our contribution in a huge variety of applications, including image segmentation, fraud detection and recognition of potential hazardous asteroids. The chapter also contains remarkable theoretical results regarding the identifiability of full binned diagonal multivariate Gaussian mixture and bin-marginal Gaussian mixtures.

In Chapter 4, we dealt with further topics regarding multivariate bin-marginal Gaussian mixtures from an experimental point of view. In particular, we debated the problem of local maxima in bin-marginal composite likelihood, we furnished heuristics to choose the number K of components and we provided a first experimental insight regarding the influence of the binning grid on clustering quality.

Chapter 5 presented an application of our proposal to time-series data provided by the start-up DiagRAMS Technologies of Lille, with the aim of detecting anomalies in industrial processes. This real data experience showed how flexible our technique is, as it exhibited good performances in time-dependent data, even if it had been firstly developed to analyze cross-sectional data sets. Moreover, our method revealed to be attractive for industries, as it has very appealing features, such as frugality and the fact of being an unsupervised method.

6.2 Perspectives

There are several possible directions for future research on this topic. In this section, we describe some future perspectives of the present work.

- In this thesis we have considered data sets with a moderate number of classes. It could be very timely to extend the presented method to those challenging situations where several small classes might appear. This will allow to recognize more types of interesting and hidden patterns inside very huge data collections. Probably, in these more complex tasks, our data-reduction is really extreme and, maybe, a softer compression is needed to save enough multivariate statistical information. Thus, possible extension could be either an intelligent and frugal usage of bivariate grids or hybrid methods involving both bin-marginal and raw data. Another option could be the use of several univariate binned data corresponding to projections of the original data on randomly selected axes. In all of these cases, it is demanded to remain inside the computational constraints of the context of reference.
- In Chapter 4, we provided two heuristics to choose the number of classes for Gaussian mixtures when using our bin-marginal method. The two heuristics, in particular the second one (BM-BIC-1), provides possible lines of future research that could be followed to provide a reliable model choice criterion. This is not only a pure theoretical issue, as it has important practical implications. Indeed, the formulation of a model choice criterion would finally allow the complete automation of the technique and a more precise clustering.

- The influence of the grid on our method performances has only been investigated experimentally in this work, showing that there is no difference between sufficiently dense grids. It would be interesting to know what the minimal acceptable refinement degree is, in order to maximize storage savings (more bins implies more data to store). We can also imagine that binning grid also could help in selecting variables. Indeed, we can reasonably suppose that the selection of a certain grid refinement degree for a variable is correlated to the degree of importance of the same variable for the clustering. According to this heuristic, we could infer that uninformative variables are associated to very coarse grids. Indeed, at the limit case when a marginal grid is restricted to a single bin, we can recognize an outcome equivalent to variable selection. Thus, such an approach could provide a very appealing half-way strategy instead of "hard" classical variable selection.

- Bin-marginal CL-EM initialization also requires special care in order to avoid local maxima. For this reason, it is necessary to investigate the theoretical properties of bin-marginal composite log-likelihood, studying in particular its local maxima and the rate of convergence of the related estimator towards the true parameter. In addition to the definition of smart initialization paradigms, it could be profitable to design stochastic versions of the bin-CL-EM algorithm, using as reference the works of Celeux and Diebolt (1985) (Stochastic EM algorithm) and Celeux and Diebolt (1992) (Simulated Annealing EM algorithm).

- As we have seen in Chapter 5, our method can be applied to time-series data. In the relative analyses we have noticed that substantial improvements could be obtained if we successfully managed system transitions. It is in this direction that we recommend further research, suggesting developing solutions to detect them. It could be also of interest to develop a frugal pre-processing technique in order to apply our proposal directly on time-series without using the statistical information provided by the additional data sets described in Chapter 5. We also aim to include in the analysis other statistical quantities, such as means and Fourier coefficient, in order to improve performances and increase the possibility of detecting other anomalous patterns.

Bibliography

Abraham, B. and Box, G. E. (1979). Bayesian analysis of some outlier problems in time series. *Biometrika*, 66(2):229–236.

Abraham, B. and Chuang, A. (1989). Outlier detection and time series modeling. *Technometrics*, 31(2):241–248.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105.

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the eleventh international conference on data engineering*, pages 3–14. IEEE.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, pages 267–281. Akadémiai Kiadó Location Budapest, Hungary.

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132.

Ando, S. (2007). Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 13–22. IEEE.

Arning, A., Agrawal, R., and Raghavan, P. (1996). A linear method for deviation detection in large databases. In *KDD*, volume 1141, pages 972–981.

Aruoba, S. B. and Fernández-Villaverde, J. (2015). A comparison of programming languages in macroeconomics. *Journal of Economic Dynamics and Control*, 58:265–273.

Assent, I., Krieger, R., Müller, E., and Seidl, T. (2007). DUSC: Dimensionality unbiased subspace clustering. In *seventh IEEE international conference on data mining (ICDM 2007)*, pages 409–414. IEEE.

Augusteijn, M. and Folkert, B. (2002). Neural network classification and novelty detection. *International Journal of Remote Sensing*, 23(14):2891–2902.

Banfield, J. D. and Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821.

Barbara, D., Wu, N., and Jajodia, S. (2001). Detecting novel network intrusions using bayes estimators. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17. SIAM.

Barnett, V. and Lewis, T. (1984). Outliers in statistical data. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*.

Baudry, J.-P. and Celeux, G. (2015). EM for mixtures. *Statistics and computing*, 25(4):713–726.

Bellman, R. E. (1961). *Adaptive control processes*. Princeton university press.

Biernacki, C. (2007). Degeneracy in the maximum likelihood estimation of univariate gaussian mixtures for grouped data and behaviour of the EM algorithm. *Scandinavian journal of statistics*, 34(3):569–586.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4):561–575.

Boehmke, B. and Greenwell, B. (2019). *Hands-on machine learning with R*. Chapman and Hall/CRC.

Bradley, P. S., Fayyad, U., and Reina, C. (1998a). Scaling clustering algorithms to large databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, page 9–15. AAAI Press.

Bradley, P. S., Fayyad, U., Reina, C., et al. (1998b). Scaling EM (expectation-maximization) clustering to large databases. *Microsoft Research*, pages 0–25.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Byers, S. and Raftery, A. E. (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584.

Cadez, I. V., Smyth, P., McLachlan, G. J., and McLaren, C. E. (2002). Maximum likelihood estimation of mixture densities for binned and truncated multivariate data. *Machine Learning*, 47(1):7–34.

Campbell, C. and Bennett, K. (2000). A linear programming approach to novelty detection. *Advances in neural information processing systems*, 13.

Campbell, J. G., Fraley, C., Stanford, D., Murtagh, F., and Raftery, A. E. (1999). Model-based methods for textile fault detection. *International Journal of Imaging Systems and Technology*, 10(4):339–346.

Čampulová, M., Michálek, J., Mikuška, P., and Bokal, D. (2018). Nonparametric algorithm for identification of outliers in environmental data. *Journal of Chemometrics*, 32(5):e2997.

Celeux, G. (1998). Bayesian inference for mixture: The label switching problem. In *Compstat*, pages 227–232. Springer.

Celeux, G., Chauveau, D., and Diebolt, J. (1995). *On stochastic versions of the EM algorithm*. PhD thesis, INRIA.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

Celeux, G. and Diebolt, J. (1992). A stochastic approximation type EM algorithm for the mixture problem. *Stochastics: An International Journal of Probability and Stochastic Processes*, 41(1-2):119–134.

Celeux, G., Frühwirth-Schnatter, S., and Robert, C. P. (2018). Model selection for mixture models—perspectives and strategies. *Handbook of mixture analysis*, pages 121–160.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793.

Celeux, G. and Soromenho, G. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, 13(2):195–212.

Chan, P. and Stolfo, S. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

Chen, Y.-Z. and Lai, Y.-C. (2016). Universal structural estimator and dynamics approximator for complex networks.

Cheng, C.-H., Fu, A. W., and Zhang, Y. (1999). Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93.

Choi, J. and Kwon, H.-J. (2015). The information filtering of gene network for chronic diseases: Social network perspective. *International Journal of Distributed Sensor Networks*, 2015:1–6.

Coleman, D. A. and Woodruff, D. L. (2000). Cluster analysis for large datasets: An effective algorithm for maximizing the mixture likelihood. *Journal of Computational and Graphical Statistics*, 9(4):672–688.

Cramer, H. (1946). *Mathematical Methods of Statistic*. Princeton university press.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., and Bontempi, G. (2017). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–14.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., and Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, 41(10):4915–4928.

Dasgupta, A. and Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American statistical Association*, 93(441):294–302.

Dasgupta, D. and Nino, F. (2000). A comparison of negative and positive selection algorithms in novel pattern detection. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, volume 1, pages 125–130. IEEE.

Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–1313. IEEE.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474.

De Stefano, C., Sansone, C., and Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1):84–94.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C., and Pregibon, D. (1999). Squashing flat files flatter. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 6–15.

Dutta, H., Giannella, C., Borne, K., and Kargupta, H. (2007). Distributed top-K outlier detection from astronomy catalogs using the demac system. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 473–478. SIAM.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18.

Endler, D. (1998). Intrusion detection. applying machine learning to solaris audit data. In *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217)*, pages 268–279. IEEE.

Entezami, A., Sarmadi, H., and Razavi, B. (2020). An innovative hybrid strategy for structural health monitoring by modal flexibility and clustering methods. *Journal of Civil Structural Health Monitoring*, 10.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316.

Fernández, A., del Río, S., Chawla, N. V., and Herrera, F. (2017). An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems*, 3(2):105–120.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631.

Fraley, C., Raftery, A. E., Murphy, T. B., and Scrucca, L. (2012). mclust version 4 for R: normal mixture modeling for model-based clustering, classification, and density estimation. Technical report, Technical report.

Fujimaki, R., Yairi, T., and Machida, K. (2005). An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 401–410.

Gao, X. and Song, P. X.-K. (2010). Composite likelihood bayesian information criteria for model selection in high-dimensional data. *Journal of the American Statistical Association*, 105(492):1531–1540.

Gao, X. and Song, P. X.-K. (2011). Composite likelihood EM algorithm with applications to multivariate hidden Markov model. *Statistica Sinica*, pages 165–185.

Ghosh, S. and Reilly, D. L. (1994). Credit card fraud detection with a neural-network. In *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, volume 3, pages 621–630. IEEE.

Goil, S., Nagesh, H., and Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets. Citeseer.

Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

Guha, S., Rastogi, R., and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *ACM Sigmod record*, 27(2):73–84.

Guha, S., Rastogi, R., and Shim, K. (2000). ROCK: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366.

Guttormsson, S. E., Marks, R., El-Sharkawi, M., and Kerszenbaum, I. (1999). Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion*, 14(1):16–22.

Hassan, N., Yau, K.-L. A., and Wu, C. (2019). Edge computing in 5G: A review. *IEEE Access*, 7:127276–127289.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hathaway, R. J. (1983). *Constrained maximum-likelihood estimation for a mixture of m univariate normal distributions*. PhD thesis, Rice University.

Hathaway, R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13(2):795–800.

Holešovskỳ, J., Čampulová, M., and Michálek, J. (2018). Semiparametric outlier detection in nonstationary times series: Case study for atmospheric pollution in brno, czech republic. *Atmospheric Pollution Research*, 9(1):27–36.

Hossain, M. S. (2020). Asteroid dataset. `https://www.kaggle.com/sakhawat18/asteroid-dataset`.

Huang, D., Wang, C.-D., Wu, J.-S., Lai, J.-H., and Kwoh, C.-K. (2019). Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Ingrassia, S. (2004). A likelihood-based constrained algorithm for multivariate normal mixture models. *Statistical Methods and Applications*, 13(2):151–166.

Jin, H., Wong, M.-L., and Leung, K.-S. (2005). Scalable model-based clustering for large databases based on data summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1710–1719.

Joshi, M. V., Agarwal, R. C., and Kumar, V. (2001). Mining needle in a haystack: classifying rare classes via two-phase rule induction. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 91–102.

Kailing, K., Kriegel, H.-P., and Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 246–256. SIAM.

Kaufman, L. and Rousseeuw, P. (1990). Finding groups in data; an introduction to cluster analysis. Technical report, J. Wiley.

Keogh, E., Lonardi, S., and Ratanamahatana, C. A. (2004). Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215.

Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 49–66.

Kiefer, J. and Wolfowitz, J. (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, pages 887–906.

King, S., King, D., Astley, K., Tarassenko, L., Hayton, P., and Utete, S. (2002). The use of novelty detection techniques for monitoring high-integrity plant. In *Proceedings of the International Conference on Control Applications*, volume 1, pages 221–226. IEEE.

Kou, Y., Lu, C.-T., and Chen, D. (2006). Spatial weighted outlier detection. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 614–618. SIAM.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Labib, K. and Vemuri, R. (2002). NSOM: A real-time network-based intrusion detection system using self-organizing maps. *Networks and Security*, 21(1).

Lee, W. and Xiang, D. (2000). Information-theoretic measures for anomaly detection. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 130–143. IEEE.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30.

Leroux, B. G. (1992). Consistent estimation of a mixing distribution. *The Annals of Statistics*, pages 1350–1360.

Li, M., Vitányi, P., et al. (2008). *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer.

Lin, J., Keogh, E., Fu, A., and Van Herle, H. (2005). Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. IEEE.

Lin, S. and Brown, D. E. (2006). An outlier-based data association method for linking criminal incidents. *Decision Support Systems*, 41(3):604–615.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary mathematics*, 80(1):221–239.

Liu, G., Li, J., Sim, K., and Wong, L. (2007). Distance based subspace clustering with flexible dimension partitioning. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 1250–1254. IEEE.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Maitra, R. (2009). Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):144–157.

McLachlan, G. and Jones, P. (1988). Fitting mixture models to grouped and truncated data via the EM algorithm. *Biometrics*, pages 571–578.

McLachlan, G. J. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

McNicholas, P. D. and Murphy, T. B. (2008). Parsimonious gaussian mixture models. *Statistics and Computing*, 18(3):285–296.

Molenberghs, G. and Verbeke, G. (2005). Models for discrete longitudinal data.

Moore, A. (1998). Very fast EM-based mixture model clustering using multiresolution kd-trees. *Advances in Neural information processing systems*, 11.

Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

NASA (2017). Nasa's hubble observes the farthest active inbound comet yet seen. `https://hubblesite.org/contents/news-releases/2017/news-2017-40.html`. Accessed: 2021-08-10.

Ng, R. T. and Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016.

Nguyen, H. D., Forbes, F., and McLachlan, G. J. (2020). Mini-batch learning of exponential family finite mixture models. *Statistics and Computing*, 30(4):731–748.

Niu, X., Wang, L., and Yang, X. (2019). A comparison study of credit card fraud detection: Supervised versus unsupervised. *arXiv preprint arXiv:1904.10604*.

Noble, C. C. and Cook, D. J. (2003). Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636.

Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., and Panda, D. (2003). Towards nic-based intrusion detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–728.

Pal, N. R. and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294.

Pan, W. (2001). Akaike's information criterion in generalized estimating equations. *Biometrics*, 57(1):120–125.

Pandove, D., Goel, S., and Rani, R. (2018). Systematic review of clustering high-dimensional and large datasets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):1–68.

Parra, L., Deco, G., and Miesbach, S. (1996). Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269.

Pires, A. and Santos-Pereira, C. (2005). Using clustering and robust estimators to detect outliers in multivariate data.

Quarta, A. A. and Mengali, G. (2010). Electric sail missions to potentially hazardous asteroids. *Acta Astronautica*, 66(9-10):1506–1519.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rajaraman, V. (2016). Big data analytics. *Resonance*, 21(8):695–716.

Ranalli, M. and Rocci, R. (2016a). Mixture models for ordinal data: a pairwise likelihood approach. *Statistics and Computing*, 26(1-2):529–547.

Ranalli, M. and Rocci, R. (2016b). Standard and novel model selection criteria in the pairwise likelihood estimation of a mixture model for ordinal data. In *Analysis of large and complex data*, pages 53–68. Springer.

Ranalli, M. and Rocci, R. (2016c). Standard and novel model selection criteria in the pairwise likelihood estimation of a mixture model for ordinal data. In *Analysis of large and complex data*, pages 53–68. Springer.

Ranalli, M. and Rocci, R. (2017a). Mixture models for mixed-type data through a composite likelihood approach. *Computational Statistics & Data Analysis*, 110:87–102.

Ranalli, M. and Rocci, R. (2017b). A model-based approach to simultaneous clustering and dimensional reduction of ordinal data. *psychometrika*, 82(4):1007–1034.

Rao, C. R. (1957). Maximum likelihood estimation for the multinomial distribution. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 18(1/2):139–148.

Reddy, A., Ordway-West, M., Lee, M., Dugan, M., Whitney, J., Kahana, R., Ford, B., Muedsam, J., Henslee, A., and Rao, M. (2017). Using Gaussian mixture models to detect outliers in seasonal univariate network traffic. In *2017 IEEE Security and Privacy Workshops (SPW)*, pages 229–234. IEEE.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM review*, 26(2):195–239.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792.

Roeder, K. and Wasserman, L. (1997). Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, 92(439):894–902.

Roth, V. (2004). Outlier detection with one-class kernel fisher discriminants. *Advances in Neural Information Processing Systems*, 17.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE.

Salvador, S., Chan, P., and Brodie, J. (2004). Learning states and rules for time series anomaly detection. In *FLAIRS conference*, pages 306–311.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, pages 461–464.

Scrucca, L. and Raftery, A. E. (2015). Improved initialisation of model-based clustering using gaussian hierarchical partitions. *Advances in data analysis and classification*, 9(4):447–460.

Sembiring, R. W., Mohamad Zain, J., and Abdullah, E. (2011). A comparative agglomerative hierarchical clustering method to cluster implemented course. *Journal of Computing*, 2.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB*, volume 98, pages 428–439.

Shekhar, S., Lu, C.-T., and Zhang, P. (2001). Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 371–376.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. Technical report, Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering.

Smith, R., Bivens, A., Embrechts, M., Palagiri, C., and Szymanski, B. (2002). Clustering approaches for anomaly based intrusion detection. *Proceedings of intelligent engineering systems through artificial neural networks*, 9.

Spence, C., Parra, L., and Sajda, P. (2001). Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *Proceedings IEEE workshop on mathematical methods in biomedical image analysis (MMBIA 2001)*, pages 3–10. IEEE.

Stanford, D. C. and Raftery, A. E. (2000). Finding curvilinear features in spatial point patterns: principal curve clustering with noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):601–609.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.

Sun, H., Bao, Y., Zhao, F., Yu, G., and Wang, D. (2004). Cd-trees: An efficient index structure for outlier detection. In *International Conference on Web-Age Information Management*, pages 600–609. Springer.

Tahir, M. A., Kittler, J., Mikolajczyk, K., and Yan, F. (2009). A multiple expert approach to the class imbalance problem using inverse random under sampling. In *International workshop on multiple classifier systems*, pages 82–91. Springer.

Tang, J., Chen, Z., Fu, A. W.-C., and Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548. Springer.

Tian, Z., Ramakrishnan, R., and Birch, L. M. (1996). An efficient data clustering method for very large databases. In *Proc of the ACM SIGMOD International Conference on Management of Data. Montre—al, Canada*, pages 103–114.

To, H. Q. (2021). Single cell images fold 0 [hpa]. `https://www.kaggle.com/quochungto/cells-fold0`. Accessed: 2021-08-10.

Tomar, D. and Agarwal, S. (2013). A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology*, 5(5):241–266.

Tsapanos, N., Tefas, A., Nikolaidis, N., and Pitas, I. (2016). Efficient mapreduce kernel K-means for big data clustering. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, pages 1–5.

ULB, M. L. G. (2018). Credit card fraud detection. `https://www.kaggle.com/mlg-ulb/creditcardfraud`. Accessed: 2021-08-10.

Valiant, G. J. (2012). *Algorithmic approaches to statistical questions*. PhD thesis, UC Berkeley.

Van Laarhoven, P. J. and Aarts, E. H. (1987). Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer.

Varin, C., Reid, N., and Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, pages 5–42.

Varin, C. and Vidoni, P. (2005). A note on composite likelihood inference and model selection. *Biometrika*, 92(3):519–528.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Wald, A. (1949). Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, 20(4):595–601.

Ward, J. H. J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Warrender, C., Forrest, S., and Pearlmutter, B. (1999). Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE symposium on security and privacy (Cat. No. 99CB36344)*, pages 133–145. IEEE.

Wei, L., Qian, W., Zhou, A., Jin, W., and Yu, J. X. (2003). Hot: Hypergraph-based outlier test for categorical data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 399–410. Springer.

West, M. (1997). Hierarchical mixture models in neurological transmission analysis. *Journal of the american statistical association*, 92(438):587–606.

Whitaker, T., Beranger, B., and Sisson, S. A. (2020). Composite likelihood methods for histogram-valued random variables. *Statistics and Computing*, pages 1–19.

Wolfe, J., Haghighi, A., and Klein, D. (2008). Fully distributed EM for very large datasets. In *Proceedings of the 25th international conference on Machine learning*, pages 1184–1191.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

Xu, Z. and Saleh, J. H. (2021). Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliability Engineering & System Safety*, 211:107530.

Yakowitz, S. J. and Spragins, J. D. (1968). On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, pages 209–214.

Yeung, D.-Y. and Chow, C. (2002). Parzen-window network intrusion detectors. In *Object recognition supported by user interaction for service robots*, volume 4, pages 385–388. IEEE.

Yu, H., Ni, J., Dan, Y., and Xu, S. (2012). Mining and integrating reliable decision rules for imbalanced cancer gene expression data sets. *Tsinghua Science and Technology*, 17(6):666–673.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., Franklin, M. J., Shenker, S., and Stoica, I. (2012). Resilient distributed datasets: A Fault-Tolerant abstraction for In-Memory cluster computing. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 15–28.

Zayani, A., N'Cir, C.-E. B., and Essoussi, N. (2016). Parallel clustering method for non-disjoint partitioning of large-scale data based on spark framework. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1064–1069. IEEE.

Zhang, H., Zhang, Z., Zhang, L., Yang, Y., Kang, Q., and Sun, D. (2019). Object tracking for a smart city using IoT and edge computing. *Sensors*, 19(9):1987.

Zhao, W., Ma, H., and He, Q. (2009). Parallel K-means clustering based on mapreduce. In *IEEE international conference on cloud computing*, pages 674–679. Springer.