On unsupervised learning in high dimension

Mehdi Sebbar

THÈSE DE DOCTORAT

de

L'Université Paris-Saclay

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École nationale de la statistique et de l'administration économique

Établissement d'accueil : *Laboratoire d'accueil :* ENSAE-X Centre d'économie, statistique et sociologie, UMR 9194 CNRS

Spécialité de doctorat : Mathématiques appliquées

Mehdi SEBBAR

On unsupervised learning in high dimension

Date de soutenance : 12 décembre 2017

Après avis des rapporteurs : Clément MARTEAU (Université Lyon I)
Vincent RIVOIRARD (Université Paris Dauphine)

Jury de soutenance :

Clément MARTEAU	(Université Lyon I) Rapporteur
Vincent RIVOIRARD	(Université Paris Dauphine) Rapporteur
Arnak DALALYAN	(ENSAE & CREST) Directeur de thèse
Alexandre TSYBAKOV	(ENSAE & CREST) Président
Katia MEZIANI	(Université Paris Dauphine) Examinateur
Philippe ROLLET	(Artefact) Examinateur

NNT : 2017SACLG003

Remerciements

Je tiens tout d'abord à remercier infiniment mon directeur de thèse, Arnak Dalalyan, d'avoir accepté de me prendre en tant qu'étudiant doctorant. Merci pour son engagement, sa disponibilité et sa patience. Ces trois années sous sa direction ont été d'une richesse immense.

Je tiens à exprimer ma plus grande considération aux rapporteurs, Clément Marteau et Vincent Rivoirard, et j'adresse également mes remerciements aux membres du jury de thèse, Katia Meziani, Alexander Tsybakov, et Philippe Rolet qui m'a permis de réaliser cette thèse au sein d'Artefact.

Je remercie les doctorants du CREST, Alexander, Vincent, Edwin, James, Lena, Lionel, The Tien, Mohamed, Adil, Gauthier, Geoffrey et Alexis avec qui j'ai passé de très bons moments. Je remercie aussi les chercheurs du CREST, Pierre Alquier, Nicolas Chopin, Guillaume Lecué, Cristina Butucea et Marco Cuturi pour tout ce que j'ai appris avec eux. Je remercie aussi la salle café du CREST pour son ambiance et les débats riches qui ont eu lieu. J'en garderai un souvenir ému.

Je tiens aussi à exprimer mes remerciements à Artefact, en particulier Philippe et Guillaume pour m'avoir permis de faire cette thèse, j'en garderai un excellent souvenir. Une pensée particulière à mes collègues de l'équipe R&D: Xavier, Ulysse, Thibaut, William, Baptor, Guillaume, Hanan, Leopold, Dayvid, Joachim, François, Wouter, Maxime, Charles, Félicien, Chaimaa, Gauthier, Sarah, Damien, Stéphane, Vanessa, Samuel, Matthieu, Diane, Aurélien et Evgeniy. J'ai passé d'excellents moments avec vous !

Je ne pourrai remercier assez mes parents et mes sœurs, Sarah et Jehane, leur soutien infaillible pendant ces trois années a été très important pour moi et je leur en suis infiniment reconnaissant. Je remercie tous mes amis, en particulier Flavie, Julien et Nirmala pour leur bonne humeur et tous les moments passés ensemble. Enfin, merci Maud pour ton immense patience et tes encouragements qui n'ont jamais fait défaut.

Contents

Résumé substantiel

Dans ce mémoire de thèse[1], nous abordons deux thèmes, le clustering en haute dimension d'une part et l'estimation de densités de mélange d'autre part. Le premier chapitre est une introduction au clustering. Nous y présentons différentes méthodes répandues et nous nous concentrons sur un des principaux modèles de notre travail qui est le mélange de Gaussiennes. Nous abordons aussi les problèmes inhérents à l'estimation en haute dimension (Section 1.3) et la difficulté d'estimer le nombre de clusters (Section 1.1.4). Nous exposons brièvement ici les notions abordées dans ce manuscrit.

Considérons une loi mélange de K Gaussiennes dans $\mathbb{R}^p$ et notons f sa densité par rapport à la mesure de Lebesgue. Notons $\boldsymbol{\mu}_k$ et $\boldsymbol{\Sigma}_k$ respectivement la moyenne et la variance de la k-ème composante Gaussienne. Alors, la densité f peut s'écrire pour $x \in \mathbb{R}^p$: $f(x) = \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(x)$, où $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ est le vecteur de poids du mélange dans $[0,1]^K$ qui vérifie $\sum_{k=1}^{K} \pi_k = 1$, et $\varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}$ est la densité de la k-ième Gaussienne. Soient $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$, n variables aléatoires i.i.d. dans $\mathbb{R}^p$. Une des approches courantes pour estimer les paramètres du mélange est d'utiliser l'estimateur du maximum de vraisemblance: $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathrm{P}_{\boldsymbol{\theta}}(\boldsymbol{X}_1, \dots, \boldsymbol{X}_n)$. Dans la dernière formule, la notation θ désigne l'ensemble des paramètres décrivant une loi de mélange (moyennes, matrices de covariances et poids des composantes). Ce problème n'étant pas convexe, on ne peut garantir la convergence des méthodes classiques comme la descente de gradient ou l'algorithme de Newton. Cependant, en exploitant la biconvexité de la log-vraisemblance négative, on peut utiliser la procédure itérative "Expectation-Maximization" (EM) décrite dans la Section 1.2.1. Malheureusement, cette méthode n'est pas bien adaptée pour relever les défis posés par la grande dimension. Par ailleurs, il est nécessaire de connaître le nombre de clusters afin de l'utiliser.

Le Chapitre 2 présente trois méthodes que nous avons développées pour tenter de résoudre les problèmes décrits précédemment. Les travaux qui y sont exposés n'ont pas fait l'objet de recherches approfondies pour diverses raisons. La première méthode que l'on

[1] Thèse effectué dans le cadre d'une convention CIFRE avec l'entreprise ARTEFACT.

pourrait appeler "lasso graphique sur des mélanges de Gaussiennes" consiste à estimer les matrices inverses des matrices de covariance $\mathbf{\Sigma}$ (Section 2.1) dans l'hypothèse où celles-ci sont parcimonieuses. Nous adaptons la méthode du lasso graphique de [Friedman et al., 2007] sur une composante dans le cas d'un mélange et nous évaluons expérimentalement cette méthode. Les deux autres méthodes abordent le problème d'estimation du nombre de clusters dans le mélange. La première est une estimation pénalisée de la matrice des probabilités postérieures $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n \times K}$ dont la composante (i, j) est la probabilité que la i-ème observation soit dans le j-ème cluster. Malheureusement, cette méthode s'est avérée trop coûteuse en complexité (Section 2.2.1). Enfin, la deuxième méthode considérée consiste à pénaliser le vecteur de poids $\boldsymbol{\pi}$ afin de le rendre parcimonieux. Cette méthode montre des résultats prometteurs (Section 2.2.2).

Dans le Chapitre 3, nous étudions l'estimateur du maximum de vraisemblance d'une densité de n observations i.i.d. sous l'hypothèse qu'elle est bien approximée par un mélange de plusieurs densités données. Nous nous intéressons aux performances de l'estimateur par rapport à la perte de Kullback-Leibler. Nous établissons des bornes de risque sous la forme d'inégalités d'oracle exactes, que ce soit en probabilité ou en espérance. Nous démontrons à travers ces bornes que, dans le cas du problème d'agrégation convexe, l'estimateur du maximum de vraisemblance atteint la vitesse $((\log K)/n)^{1/2}$, qui est optimale à un terme logarithmique près, lorsque le nombre de composant est plus grand que $n^{1/2}$. Plus important, sous l'hypothèse supplémentaire que la matrice de Gram des composantes du dictionnaire satisfait la condition de compatibilité, les inégalités d'oracles obtenues donnent la vitesse optimale dans le scénario parcimonieux. En d'autres termes, si le vecteur de poids est (presque) D-parcimonieux, nous obtenons une vitesse $(D \log K)/n$. En complément de ces inégalités d'oracle, nous introduisons la notion d'agrégation (presque)-D-parcimonieuse et établissons pour ce type d'agrégation les bornes inférieures correspondantes.

Enfin, dans le Chapitre 4, nous proposons un algorithme qui réalise l'agrégation en Kullback-Leibler de composantes d'un dictionnaire telle qu'étudiée dans le Chapitre 3. Nous comparons sa performance avec différentes méthodes: l'estimateur de densité à noyaux, l'estimateur "Adaptive Dantzig", l'estimateur SPADES et EM avec le critère BIC. Nous proposons ensuite une méthode pour construire le dictionnaire de densités et l'étudions de manière numérique.

Chapter 1

Introduction

Contents

In this thesis, we focus on the unsupervised learning problem through the study of clustering in high dimensional (Gaussian) mixtures and density estimation. In this chapter, we introduce the clustering problem in the first section and the Gaussian mixtures framework in the second. In the third section, we highlight the complexities inherent to the high dimension. Then we will discuss some of the work carried out during this thesis but has been left unfinished for various reasons.

1.1 Clustering and density estimation problem

The goal of cluster analysis is to find groups in data so that each element within a group is more similar to other elements of the same group rather than to those outside of the group. The literature is rich on this topic, with different approaches coming from statistics and computer science. A clustering problem has several dimensions: the input data can be a distance or similarity matrix, the similarity can be for instance a kernel chosen by expert knowledge. The input can also be a raw data matrix with N rows, the observations and p columns, the features. Another dimension for clustering methods is 'hard versus soft' assignment of points to clusters. In hard assignment, a point is assigned to a unique cluster. In soft assignment, for each point, the probabilities of belonging to each clusters are furnished. This particularity is specific to probabilistic methods. A third dimension is

flat versus hierarchical clustering. In flat clustering, the output is a partition of the dataset or the state-space into disjoint clusters whereas in hierarchical clustering the output is a tree of nested clusters. The latter is a finite sequence of nested partitions. We will give a glimpse on 4 well-known clustering techniques, K-means, Hierarchical clustering, Spectral clustering and the Gaussian mixtures model with the Expectation-Maximization algorithm (EM) which will be our topic of main interest. The reader can refer to [Hennig et al., 2015] for an extensive review of cluster analysis.

1.1.1 Centroid-Based Clustering: K-means

K-means is a popular method of clustering which aims to partition the data into K clusters such that the within-cluster sum of squares of distances to the cluster center is minimal. It has been introduced in signal theory for vector quantization by [MacQueen, 1967]. Given N points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ in $\mathbb{R}^p$, the goal of K-means is to find a set of centers $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ that minimizes the following objective function:

$$\mathcal{L}_{k\text{-means}}(\mathcal{C}) = \sum_{i=1}^{N} \min_{\boldsymbol{c} \in \mathcal{C}} \|\boldsymbol{x}_i - \boldsymbol{c}\|^2. \tag{1.1}$$

Clearly this objective function is not convex and finding an exact solution of this problem is known to be NP-hard, even for 2-means [Dasgupta, 2008, Aloise et al., 2009]. As a matter of fact, for K and p fixed, the problem can be solved exactly in $O(n^{Kp})$ iterations [Inaba et al., 1994]. A simple and yet widely used approximation method to resolve the K-means minimization problem is Lloyd's algorithm [Lloyd, 1982]. Today, because of its popularity, Lloyd's method is assimilated with the minimization problem of K-means (eq. (1.1)). A key element of this method is the Voronoi partitioning:

Definition 1. *(Voronoi Partition) Given n points in $\mathbb{R}^p$, K points $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K \in \mathbb{R}^p$ and a distance d, a Voronoi partition of $\mathbb{R}^p$ consists on K disjoint clusters such that for $i \in [K]$, cluster i is the set of points satisfying $d(\boldsymbol{x}, \boldsymbol{c}_i) \leq d(\boldsymbol{x}, \boldsymbol{c}_j)$ for all $j \neq i$.*

Lloyd's procedure, depicted in Figure 1.1 and described in Figure 1.2, consists in building a Voronoi partition of the data from K initial randomly chosen centers and iterate partitioning with the cell-means of the previous partition. The following lemma will help us to understand the convergence of the algorithm:

Lemma 1.1.1. *Consider a finite set $\mathcal{X} \subset \mathbb{R}^p$ and denote by $\boldsymbol{\mu}$ its mean. Let d be a metric. For any $\mathbf{y} \in \mathbb{R}^p$, we have that*

$$\sum_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, \mathbf{y})^2 = \sum_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, \boldsymbol{\mu})^2 + |\mathcal{X}| d(\boldsymbol{\mu}, \mathbf{y})^2. \tag{1.2}$$

The reader can refer to Fact 5.1 of Hennig et al. [2015] for a simple proof. This lemma claims that, after a Voronoi partitioning, replacing a center by the mean of the cell can not increase the K-means cost. Hence this ensures the convergence of the algorithm. Unfortunately, Lloyd's algorithm tends to reach local optima of the K-means objective. Therefore, several runs of the algorithm are necessary to ensure an acceptable clustering.

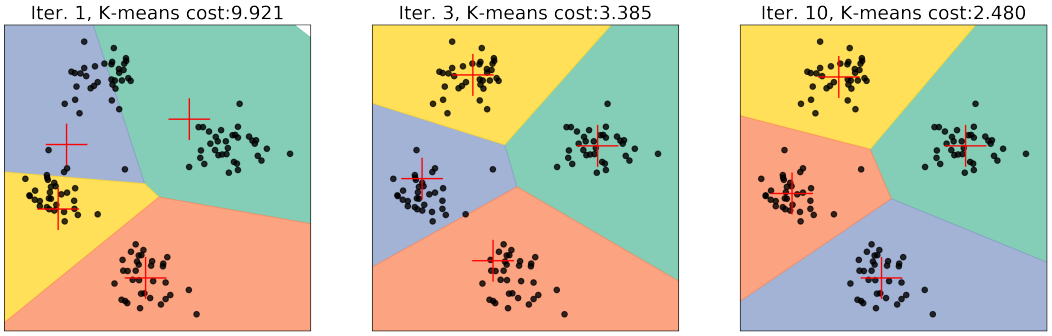

Figure 1.1: Llyod's algorithm with randomly initialized centers and final Voronoi partitions at different steps: with 1 iteration (left), 3 (middle) and 10 (right) iterations (the algorithm converged). K-means costs are given on top.

Input: N points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K.
Output: Cluster centers $\widehat{\boldsymbol{c}}_1, \ldots, \widehat{\boldsymbol{c}}_K$ and clusters assignments.
Init: Set $\mathcal{L}_{\text{old}} = \infty$. and chose K seed points $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$. Compute the K-means cost $\mathcal{L}_{\text{curr}}$ given in eq. (1.1) with these points as centers.
while $\mathcal{L}_{\text{curr}} < \mathcal{L}_{\text{old}}$ **do**
1: Compute the Voronoi partitioning of the data with $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$ as centers. Get K clusters, $C_1, \ldots, C_K$.
2: For each cluster, compute the sample means $\widehat{\boldsymbol{c}}_1, \ldots, \widehat{\boldsymbol{c}}_K$:

$$\widehat{\boldsymbol{c}}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \tag{1.3}$$

3: Set $\mathcal{L}_{\text{old}} = \mathcal{L}_{\text{curr}}$ and compute the new K-means cost $\mathcal{L}_{\text{curr}}$ with $\widehat{\boldsymbol{c}}_1, \ldots, \widehat{\boldsymbol{c}}_K$ as centers.
end while

Figure 1.2: K-means Lloyd's algorithm

Note that Lloyd's algorithm has several drawbacks:

1. It is a hard-assignment method since it assigns points to clusters and does not reflect a level of uncertainty on the assignments such as a probability of belonging to a cluster.

2. The number of clusters has to be given, we will see some techniques to select the number of clusters in Section 2.2.

3. The worst-case time complexity $T(n)$ is superpolynomial, $T(n) = 2^{\Omega(\sqrt{n})}$ iterations [Arthur and Vassilvitskii, 2006] (not bounded above by any polynomial). Fortunately, in practice it is observed that Lloyd's algorithm converges quickly to a local minimum.

4. If the initial centers are chosen randomly, the resulting K-means cost can be made arbitrarily bad compared to the optimal clustering (see section 5.2 of [Hennig et al., 2015]). K-means++ [Arthur and Vassilvitskii, 2007] addresses this problem by choosing carefully the initial centers in Lloyd's algorithm, see Figure 1.3 for the procedure. Furthermore, K-means++ is a $\log K$ approximation algorithm for the K-means objective in the following sense.

   Theorem 1.1.1. *[Arthur and Vassilvitskii, 2007] Let $\mathcal{S}$ be the set of centers output by the algorithm K-means++ and $\mathcal{L}(S)$ be the K-means cost of the clustering obtained using $\mathcal{S}$ as the centers. Then $\mathbb{E}[\mathcal{L}(\mathcal{S})] \leq O(\log(K))\mathcal{L}^*$, where $\mathcal{L}^*$ is the cost of the optimal K-means solution.*

5. K-means can not distinguish noise or select relevant features. This last point is particularly important in the case of high dimensional data, since it is generally accepted that the most relevant clusters lies in subspaces of much smaller dimension, we will discuss this phenomenon in section 1.3. An idea would be to adapt Lloyd's method to the weighted q^{th}-root of the Minkowski metric

$$d_{\boldsymbol{w}}(\boldsymbol{x}, \mathbf{y}) = \sum_{l=1}^{p} w_l |\boldsymbol{x}_l - \mathbf{y}_l|^q, \tag{1.4}$$

   with $\boldsymbol{w}$ a weight vector updated at each iterations. A first method of weighted K-means has been introduced in [Makarenkov and Legendre, 2001] and further developed in [Zhexue Huang et al., 2007] (WK-Means) for the Euclidean norm. An extension to the Minkowski metric (MWK-Means) is proposed in [Cordeiro de Amorim and Mirkin, 2012] that outperforms K-means and WK-Means. Note that the use of a different metric has a profound impact on the implementation and running time since the computation of Minkowski centers is not straightforward.

Input: N points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K.
Init: Choose one center $\boldsymbol{c}_1$ uniformly at random among the data points and add it to the set $\mathcal{S}$.
for $j = 2$ to K **do**
1: Choose a point $\boldsymbol{x}$ from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with probability proportional to $\min_{\boldsymbol{c} \in \mathcal{S}} d(\boldsymbol{x}, \boldsymbol{c})^2$ and add it to $\mathcal{S}$.
end for
2: Proceed with K-means algorithm and the set $\mathcal{S}$ as initialized centers.

Figure 1.3: K-means++ algorithm

The research on K-means is dense and several variants of this method has been developed. For instance K-medoids [Kaufman and Rousseeuw, 1990] uses points of the data as centers, Mini-batch K-means [Sculley, 2010] takes mini-batches of data to reduce significantly computational cost without penalizing too much the K-means cost, or clustering algorithms that enjoy strong theoretical guarantees on non-worst case scenarios using the notion of stability [Ostrovsky et al., 2006]. The reader can refer to [Hennig et al., 2015] for further details on this topic.

1.1.2 Agglomerative Hierarchical Methods

In this section, we will present the Agglomerative Hierarchical clustering, a very popular method due to its simplicity and the nested structure of clusters that it produces. The idea of Hierarchical clustering is to form a hierarchy of clusters (*i.e.* nested partitions) according to a merging rule which helps us to see how clusters are related to each other (a structure unavailable with the other methods). There exist two types of hierarchical clustering: agglomerative and divisive. The first type consists in starting from N clusters, each containing one element of the dataset and in merging clusters iteratively into larger groups according to an agglomeration rule and a similarity. This process builds a hierarchy, until finding only one cluster that contains the whole dataset. The similarity can be a Minkowski distance, the cosine similarity or other distance such as Hamming, Hellinger or Mahalanobis. The divisive procedure is the opposite of the agglomerative: starting from the whole dataset and splitting iteratively until obtaining N clusters. Divisive methods are generally very expensive, with a complexity of $O(2^n)$ [Guénoche et al., 1991], and are therefore not used in practice. Let us consider the agglomerative procedure and a metric d, a simple implementation is to build the dissimilarity matrix of the N original clusters

$\{\boldsymbol{x}_1\}, \ldots, \{\boldsymbol{x}_N\}$ noted $S = (d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j\in[N]^2}$ (which is symmetric) and consider the couple (i, j) such that d_{ij} is the smallest dissimilarity in S. We create a new cluster $i \cup j$, add it to the matrix S with the rule $d_{i\cup j,k} = \min\{d_{ik}, d_{jk}\}$ and remove the rows and columns of sets i and j from S. The iteration of this procedure leads to one final cluster containing all points in the dataset. This method is called 'Single-linkage' clustering [Graham and Hell, 1985]; a naive implementation of it with a complexity of $O(n^3)$ is given in Figure 1.5. Note that it can be optimized to $O(n^2)$[Murtagh and Contreras, 2012]. The hierarchy can be visualized via a binary tree called "dendrogram", see an illustration in Figure 1.4. This method has a severe drawback called 'chaining phenomenon' referring to the fact that clusters can be merged due to the presence of close points even if they contain other points that are very distant. An alternative method called 'Complete-linkage' clustering solves this problem by taking the maximum instead of the minimum in step 1 of the Single-linkage algorithm in Figure 1.5. Similarly to Single-linkage method, the complexity of the naive implementation is $O(n^3)$ but can be optimized to $O(n^2)$. Another popular method worth mentioning for its use of cluster centers is Ward's method[Jr., 1963] also called Ward's minimum variance method which consists in optimizing an objective function, generally the sum of squared Euclidean distances between points. Let us consider the merging cost of combining clusters A and B. If $A \cap B = \emptyset$, then

$$\begin{aligned}\Delta(A, B) &= \sum_{i\in A\cup B} \|\boldsymbol{x}_i - \boldsymbol{c}_{A\cup B}\|^2 - \sum_{i\in A} \|\boldsymbol{x}_i - \boldsymbol{c}_A\|^2 - \sum_{i\in B} \|\boldsymbol{x}_i - \boldsymbol{c}_B\|^2 \\ &= \frac{n_A n_B}{n_A + n_B} \|\boldsymbol{c}_A - \boldsymbol{c}_B\|^2,\end{aligned}$$

where $\boldsymbol{c}_I$ and n_I are the center of cluster I and its size respectively. This quantity is positive, hence the within-group variance increases when merging two clusters. Ward's method seek to minimize this growth. Alternatively, this amounts to looking for the maximum between-cluster variance.

We can notice that agglomerative methods might differ on the computation of dissimilarities following the agglomeration process (step 2 in Figure 1.5). Lance and Williams developed an updating formula [Lance and Williams, 1967] for these dissimilarities that generalizes several agglomerative methods. The dissimilarity between a new merged cluster $i \cup j$ and cluster k is

$$d(i \cup j, k) = \alpha_i d(i, k) + \alpha_j d(j, k) + \beta d(i, j) + \gamma |d(i, k) - d(j, k)|, \tag{1.5}$$

where the parameters $\alpha_i, \alpha_j, \beta, \gamma$ depend on the clustering criterion. For instance, the single-linkage method is recovered by setting $\alpha_i = \alpha_j = 1/2$, $\beta = 0$ and $\gamma = -1/2$, the

complete-linkage method with $\alpha_i = \alpha_j = 1/2$, $\beta = 0$ and $\gamma = 1/2$ and Ward's method can be expressed in this framework [Batagelj, 1988, Murtagh, 1985, Jambu, 1989] with $\alpha_i = (n_i + n_k)/(n_i + n_j + n_k)$, $\alpha_j = (n_j + n_k)/(n_i + n_j + n_k)$, $\beta = -n_k/(n_i + n_j + n_k)$ and $\gamma = 0$. The reader can find parameters for other methods in Table 6.1 of [Hennig et al., 2015]. More efficient algorithms rely on 'Nearest Neighbor Chains'. We invite the

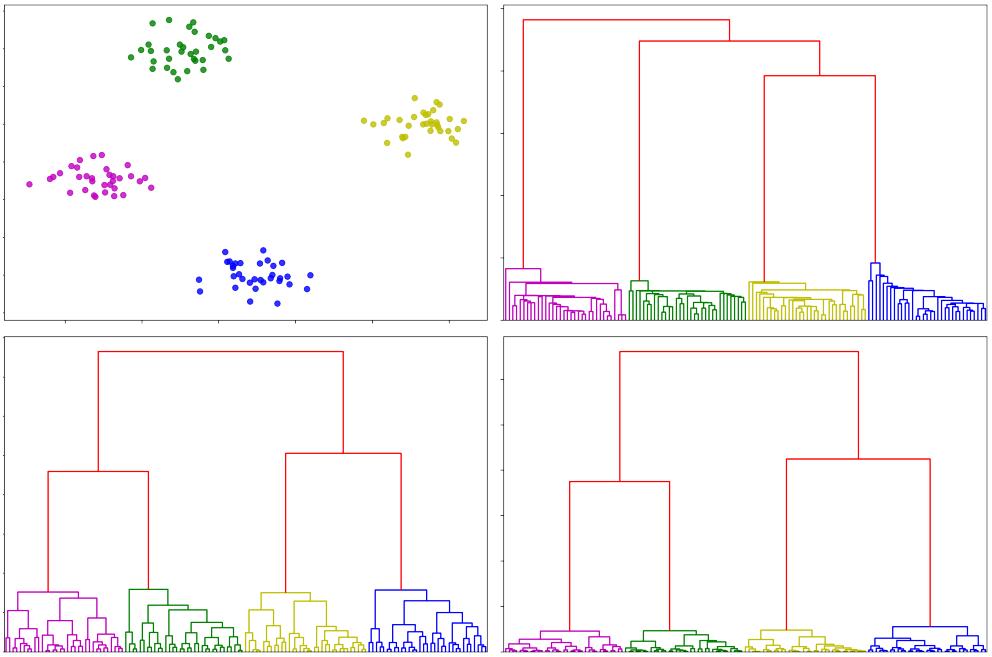

Figure 1.4: A dataset with 4 clusters (top-left) used with the Agglomerative hierarchical clustering and the corresponding dendrogram, single-link (top-right), complete-link (bottom-left) and Ward's method (bottom-right). A simple way for finding clusters would be to cut the dendrogram with a horizontal line from bottom to top until finding the number of clusters desired. Note the difficulty to recover the 4 original clusters.

reader to refer to [Murtagh and Contreras, 2012] for a detailed review on Agglomerative hierarchical methods.

1.1.3 Spectral clustering

Recently, spectral clustering has become widely used thanks to its performance compared to traditional clustering techniques and its computational attractiveness. One interesting feature of spectral clustering is that it does not make any assumption on the form of the clusters contrary to K-means. This method of clustering relies deeply on the graph theory [Donath and Hoffman, 1973, Fiedler, 1973]. The reader can refer to [von Luxburg,

Input: A similarity matrix S.
while at least 2 objects remain in S **do**
1: Determine the smallest dissimilarity d_{ij} in S.
2: Let m be the size of S, compute the dissimilarities for the new cluster $i \cup j$:

$$d_{i\cup j,k} = \min\{d_{ik}, d_{jk}\}, \quad k \in [m], k \neq i, j. \tag{1.6}$$

3: Add the dissimilarities of $i \cup j$ in S and remove those of clusters i and j.
end while

Figure 1.5: Simple single-linkage hierarchical clustering

2007] and [Spielman and Teng, 2007] for a survey of the literature on this topic. Although several methods exist which all are referred to as "Spectral clustering" we will describe the simplest formulation of this method. Let us consider $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, N points in $\mathbb{R}^p$ and a similarity measure $s_{ij} \geq 0$ between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We can construct the similarity matrix $\boldsymbol{S} = (s_{ij})_{i,j\in[N^2]}$ which can be represented by a similarity graph $G = (V, E)$ where the vertices $v_1, \ldots, v_N$ correspond to the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and the edge between v_i and v_j exists if $s_{ij} \neq 0$ and thus has weight s_{ij}. Note that G is an undirected graph, *i.e.* $s_{ij} = s_{ji}$. The main idea of spectral clustering is to find a partition of G with minimal cuts, that is to find a partition such that the cumulative weight of the edges between different groups is low and those within a group are high. This can be done by analyzing the spectrum of the Laplacian matrix $\boldsymbol{L}$ of $\boldsymbol{S}$ and a clustering such as K-means in a low-dimensional subspace spanned by eigenvectors of $\boldsymbol{L}$ corresponding to its largest eigenvalues. It is clear that a sparse graph G is interesting for such a cutting problem. There exist several methods to sparsify $\boldsymbol{S}$:

K-nearest neighbor graphs: Modify the similarity matrix $\boldsymbol{S}$ by keeping for each nodes the k-nearest neighbors and set $s_{ij} = 0$ for the other vertices. We can make this graph undirected in different ways, see section 2.2 of [von Luxburg, 2007].

ε-neighborhood graph: We connect nodes v_i and v_j if $s_{i,j} \geq \varepsilon$, this graph is usually unweighted.

We will note $\boldsymbol{W}$ the resulting weighted adjacency matrix. The reader can find more details on the behavior of these different graphs in section 8 of [von Luxburg, 2007]. The simplest approach for spectral clustering is to consider the 'unnormalized graph Laplacian', $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix called the 'degree matrix' and the element i of its diagonal is the degree of the vertex v_i, $d_i = \sum_{j=1}^{N} s_{ij}$. An important property of this

matrix is that its smallest eigenvalue is 0 and the corresponding eigenvector is the constant vector (see Proposition 1 of [von Luxburg, 2007]). In the sequel, we say that $A \subset G$ is connected if any two vertices in A can be joined with a path such that all the intermediate vertices lie in A. The subgraph A is a connected component if it is connected and there are no connections between A and its complement $\bar{A}$. An important result for spectral clustering is the following proposition:

Proposition 1 (Number of connected components, proposition 2 in [von Luxburg, 2007])**.** *The multiplicity k of the eigenvalue 0 of $\boldsymbol{L}$ is the number of connected components $A_1, \dots, A_k$ in the graph G. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $1_{A_1}, \dots, 1_{A_k}$ of these components.*

The simplest implementation of the spectral clustering is given in Figure 1.6. Two other

Input: A similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ and the number of clusters to find k.
Output: Clusters $A_1, \dots, A_k$.
1: Construct the weighted adjacency matrix $\boldsymbol{W}$.
2: Compute the unnormalized Laplacian $\boldsymbol{L}$.
3: Compute the k eigenvectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$ of $\boldsymbol{L}$ corresponding to the k smallest eigenvalues of $\boldsymbol{L}$.
4: Let $\boldsymbol{V} \in \mathbb{R}^{N \times k}$ be the matrix containing the vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$ as columns.
5: For $i \in [N]$, let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the i^{th} row of $\boldsymbol{V}$.
6: Cluster the points $(\mathbf{y}_i)_{i \in [N]} \in \mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.
7: Construct clusters $A_1, \dots, A_k$ with $A_i = \{i, \mathbf{y}_i \in C_i\}$

Figure 1.6: Unnormalized spectral clustering according to [von Luxburg, 2007]

types of Laplacian matrices are used in the literature called "normalized graph Laplacians" and offer theoretical advantages compared to the unnormalized Laplacian (see section 8.4 of [von Luxburg, 2007]). They are defined as follows:

$$\boldsymbol{L}_{\text{sym}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2} \boldsymbol{W} \boldsymbol{D}^{1/2} \quad \text{and} \quad \boldsymbol{L}_{\text{rw}} = \boldsymbol{I} - \boldsymbol{D}^{-1} \boldsymbol{W}. \tag{1.7}$$

We will refrain from addressing these two matrices, we shall content ourselves with saying that there exist more efficient spectral clustering algorithms called "Normalized spectral clustering" that are of the same spirit as Figure 1.6, the reader can refer to [Shi and Malik, 2000, Ng et al., 2001, von Luxburg, 2007] for a deeper analysis of the use of these Laplacians. We will simply give an insight on the mechanics behind the spectral clustering

algorithm and we shall highlight the problem from a graph point of view. The spectral algorithm is an approximation to the problem of partitioning the graph G. For A and B, two disjoint subsets of the vertex set V of G we define the cut of A and B as:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij}. \tag{1.8}$$

Two common objective functions to minimize for such partitioning are RatioCut [Hagen and Kahng, 1992] and Ncut [Shi and Malik, 2000] defined as

$$\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|},$$

$$\text{Ncut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{\text{vol}(A_i)},$$

where $|A|$ is the number of vertices in A and $\text{vol}(A) = \sum_{i \in A} d_i$. Note that these two objective functions try to achieve a "balanced" partitioning, a small component leads to a high value of these objective functions. Unfortunately, solving such a partitioning problem is NP-hard [Wagner and Wagner, 1993, von Luxburg, 2007]. Fortunately, a relaxation of this problem with RatioCut is:

$$\min_{\boldsymbol{H} \in \mathbb{R}^{N \times K}} Tr(\boldsymbol{H}^T \boldsymbol{L} \boldsymbol{H}) \quad \text{subject to} \quad \boldsymbol{H}^T \boldsymbol{H} = \boldsymbol{I}, \tag{1.9}$$

and can be solved explicitly. It turns out that choosing $\boldsymbol{H}$ as the matrix with the first k eigenvectors of $\boldsymbol{L}$ as columns is a solution of Equation (1.9) (see 5.2 of [von Luxburg, 2007]) which is exactly step 4 in Figure 1.6. Similar relaxation can be done for the Ncut objective function

$$\min_{\boldsymbol{U} \in \mathbb{R}^{N \times k}} Tr(\boldsymbol{U}^T \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} \boldsymbol{U}) \quad \text{subject to} \quad \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}. \tag{1.10}$$

These relaxations do not give guarantees on the quality of the solutions and the resulting partition can be significantly worse than the optimal one in regards to RatioCut and Ncut [Guattery and Miller, 1998, Nadler and Galun, 2007]. In particular, spectral clustering methods are global methods and fail to identify clusters at different scales [Nadler and Galun, 2007]. But these approximations are computationally attractive and very simple to solve, especially with a sparse weighted adjacency matrix.

1.1.4 Finding the number of clusters

The determination of the number of clusters in a dataset is fundamental and still unsolved problem. Numerous approaches to this problem has been developed over the years, see

[Hardy, 1996, Milligan and Cooper, 1985] and Chapter 26 of [Hennig et al., 2015]. Several popular heuristics rely on a graphical interpretation of the quality of clustering. The most popular is the Elbow criterion which consists in performing clusterings with different number of clusters K and computing the ratio of the between group variance and the total variance (the F-test statistic) for each K. The detection of an 'elbow' indicates the appropriate number of clusters; *i.e.* clusterings with larger K do not really improve the explained proportion of the variance. Another technique that relies on a graphical interpretation is the Silhouette method [Rousseeuw, 1987] which assesses how well a point is assigned to its cluster compared to nearest neighbor cluster. A more formal approach is the 'Gap Statistic' developed in [Tibshirani et al., 2001], an efficient statistical procedure that compares the change in within-cluster dispersion with that expected under a reference null distribution. In the case of the spectral clustering, one can use the 'eigengap' heuristic on the eigenvalues $\lambda_1, \ldots, \lambda_N$ of the Laplacian matrix by choosing K such that $\lambda_1, \ldots, \lambda_K$ are small and λ_{K+1} is relatively large (see section 8.3 of [von Luxburg, 2007]). Finally, a model selection criterion widely used in probabilistic models and important in our work is the Bayesian Information Criterion (BIC). This method has been developed in [Schwarz, 1978] following the work of Akaike on the AIC [Akaike, 1973]. The idea of this method, under the assumption that the observations $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ are drawn from an exponential family, is to derive from the approximation of the asymptotic expansion of the Bayes estimator the following quantity

$$\mathrm{BIC} = \widehat{\ell}_j(\widehat{\theta}_j, \boldsymbol{x}) - \frac{1}{2} k_j \log(N), \tag{1.11}$$

where $\widehat{\ell}_j(\widehat{\theta}, \boldsymbol{x})$ is the maximized log-likelihood of model j, $\widehat{\theta}_j$ is the MLE and k_j is the number of free parameters of model j. Therefore, the model selection rule is to choose the model for which the BIC is the largest. BIC has several nice properties; in particular, it penalizes the complexity of the model which is interesting since choosing the model only on the criterion of the likelihood in the case of Gaussian mixtures leads to select as many components as there are points.

One can remark that all procedures mentioned previously require to perform a large number of clusterings and select the best model according to a given criterion. Such an approach is computationally expensive. In Section 2.2.2, we try to address this challenge by iteratively penalizing the weight vector of large Gaussian mixture in an EM-like procedure.

1.2 The Gaussian Mixture Model

We will now focus on the Gaussian Mixture Model (GMM), an important framework for clustering problems. Unlike the other previously seen methods, it is a probabilistic approach to clustering. One of the main advantages of model-based clustering is that the resulting partition can be interpreted statistically. It assumes that the observations are drawn from a mixture distribution the components of which are Gaussian with parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the density of the k-th mixture component is

$$\varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\Big). \tag{1.12}$$

Let $\boldsymbol{\theta}$ be the list containing all the unknown parameters of a Gaussian mixture model: the family of means $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K) \in (\mathbb{R}^p)^K$, the family of covariance matrices $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_k) \in (\mathcal{S}_{++}^p)^K$ and the vector of cluster probabilities $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K) \in [0,1]^K$ such that $\mathbf{1}_K^\top \boldsymbol{\pi} = 1$. The density of one observation $\boldsymbol{X}_1$ is then given by:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^p, \tag{1.13}$$

where $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$. This model can be interpreted from a latent variable perspective. Let Z be a discrete random variable taking its values in the set $[K]$ and such that $\mathbf{P}(Z = k) = \pi_k$ for every $k \in [K]$. The random variable Z indicates the cluster from which the observation $\boldsymbol{X}$ is drawn. Considering that all the conditional distributions $\boldsymbol{X}|Z = k$ are Gaussian, we get the following formula for the marginal density of $\boldsymbol{X}$:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathbf{P}(Z = k) p_\theta(\boldsymbol{x}|Z = k) = \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^p. \tag{1.14}$$

In the clustering problem, the goal is to assign $\boldsymbol{X}$ to a cluster or, equivalently, to predict the cluster Z of the vector $\boldsymbol{X}$. A prediction function in such a context is $g : \mathbb{R}^p \to [K]$ such that $g(\boldsymbol{X})$ is as close as possible to Z. If we measure the risk of a prediction function g in terms of misclassification error rate $R_{\boldsymbol{\theta}}(g) = \mathbf{P}_{\boldsymbol{\theta}}(g(\boldsymbol{X}) \neq Z)$, then it is well known that the optimal (Bayes) predictor $g_{\boldsymbol{\theta}}^* \in \arg\min_g R_{\boldsymbol{\theta}}(g)$ is provided by the rule

$$g_{\boldsymbol{\theta}}^*(\boldsymbol{x}) = \arg\max_{k \in [K]} \tau_k(\boldsymbol{x}, \boldsymbol{\theta}),$$

where $\tau_k(\boldsymbol{x}, \boldsymbol{\theta}) = p_{\boldsymbol{\theta}}(Z = k|\boldsymbol{X} = \boldsymbol{x})$ stands for the conditional probability of the latent variable Z given $\boldsymbol{X}$. In the Gaussian mixture model, Bayes's rule implies that

$$\tau_k(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|Z = k)\mathbf{P}(Z = k)}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} = \frac{\pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x})}{\sum_{k'=1}^{K} \pi_{k'} \varphi_{\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}}(\boldsymbol{x})} \tag{1.15}$$

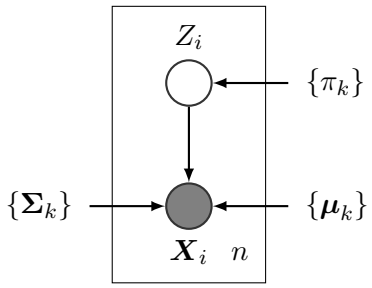

Figure 1.7: The Gaussian Mixture Model.

Since the true value of the parameter $\boldsymbol{\theta}$ is not available, formula (1.15) can not be directly used for solving the problem of clustering. Instead, a natural strategy is to estimate $\boldsymbol{\theta}$ by some vector $\widehat{\boldsymbol{\theta}}$, based on a sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ drawn from the density $p_{\boldsymbol{\theta}}$, and then to define the clustering rule by

$$\widehat{g}(\boldsymbol{x}) = g^*_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x}) = \arg\max_{k\in[K]} \tau_k(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}) = \arg\max_{k\in[K]} \widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k}(\boldsymbol{x}). \tag{1.16}$$

A common approach to estimating the parameter $\boldsymbol{\theta}$ is to rely on the likelihood maximization. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$ with $\boldsymbol{X}_i \in \mathbb{R}^p$ be a set of iid observations drawn from the density $p_{\boldsymbol{\theta}}$ given by (1.13). The graphical model in Figure 1.7 depicts the scheme of the observations. The log-likelihood of the Gaussian mixture model is

$$\ell_N(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}(\boldsymbol{x}_i) \right\}. \tag{1.17}$$

Because of the presence in this equation of the logarithm of a sum, the maximization of the log-likelihood is a difficult nonlinear and nonconvex problem. In particular, this is not a exponential family distribution yielding simple expressions. A commonly used approach for approximately maximizing (1.17) with respect to $\boldsymbol{\theta}$ is the Expectation-Maximization (EM) Algorithm [Dempster et al., 1977] that we recall below.

Summarizing the content of this section, we can describe the following natural approach to solving the clustering problem under Gaussian mixture modeling assumption:

1.2.1 EM Algorithm

The goal of the EM algorithm is to approximate a solution of the problem (1.18). Since this optimization problem contains a nonconvex cost function, it is impossible to design a polynomial time algorithm that provably converges to the global maximum point. Instead, the EM algorithm provides a sequence $\{\widehat{\boldsymbol{\theta}}(t)\}_{t\in\mathbb{N}}$ of parameter values such that the cost

Input: data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K

Output: function $\widehat{g} : \mathbb{R}^p \to [K]$

1: Estimate $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ by maximizing the log-likelihood:

$$\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ell(\boldsymbol{\theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \arg\max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) \right\}. \tag{1.18}$$

2: Output the clustering rule:

$$\widehat{g}(\cdot) = \arg\max_{k \in [K]} \widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k}(\cdot). \tag{1.19}$$

Figure 1.8: Clustering under Gaussian mixture modeling

function (*i.e.*, the log-likelihood) evaluated at these values forms an increasing sequence that converges to a local maximum.

The main idea underlying the EM algorithm is the following representation of the log-likelihood of one observation derived from the log-sum inequality:

$$\log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) \right\} = \max_{\boldsymbol{\tau} \in [0,1]^K \ \boldsymbol{\tau}^\top \mathbf{1}_K = 1} \sum_{k=1}^{K} \left\{ \tau_k \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_k \log(\pi_k / \tau_k) \right\}. \tag{1.20}$$

Let us denote by $\boldsymbol{\mathcal{T}} = (\tau_{i,k})$ a $N \times K$ matrix with nonnegative entries such that $\boldsymbol{\mathcal{T}} \mathbf{1}_K = \mathbf{1}_n$, that is each row of $\boldsymbol{\mathcal{T}}$ is a probability distribution on $[K]$. Combining (1.18) and (1.20), we get

$$\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \max_{\boldsymbol{\mathcal{T}}} \sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\}. \tag{1.21}$$

The great advantage of this new representation of the log-likelihood function is that the cost function in (1.21), considered as a function of $\boldsymbol{\theta}$ and $\boldsymbol{\mathcal{T}}$, is biconcave, *i.e.*, it is concave with respect to $\boldsymbol{\theta}$ for every fixed $\boldsymbol{\mathcal{T}}$ and concave with respect to $\boldsymbol{\mathcal{T}}$ for every fixed $\boldsymbol{\theta}$. In such a situation, one can apply the alternating maximization approach to sequentially improve on an initial point. In the present context, an additional attractive feature of the cost function in (1.21) is that the two optimization problems involved in the alternating maximization procedure admit explicit solutions.

Lemma 1. *Let us introduce the cost function*

$$F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\}. \tag{1.22}$$

Input: data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K
Output: parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k, \pi_k\}_{k\in[K]}$
1: Initialize $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
2: **Repeat**
3: Update the parameter $\boldsymbol{\mathcal{T}}$:

$$\tau_{i,k}^t = \frac{\pi_k^t \varphi_{\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t}(\boldsymbol{x}_i)}{\sum_{k'\in[K]} \pi_{k'}^t \varphi_{\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Sigma}_{k'}^t}(\boldsymbol{x}_i)}.$$

4: Update the parameter $\boldsymbol{\theta}$:

$$\pi_k^{t+1} = \frac{1}{N}\sum_{i=1}^N \tau_{i,k}^t, \qquad \boldsymbol{\mu}_k^{t+1} = \frac{1}{N\pi_k^{t+1}}\sum_{i=1}^N \tau_{i,k}^t \boldsymbol{x}_i,$$

$$\boldsymbol{\Sigma}_k^{t+1} = \frac{1}{N\pi_k^{t+1}}\sum_{i=1}^N \tau_{i,k}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})^\top.$$

5: increment t: $t = t + 1$.
6: **Until** stopping rule.
7: **Return** $\boldsymbol{\theta}^t$.

Figure 1.9: EM algorithm for Gaussian mixtures

Then, the following two optimization problems

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) \in \arg\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}), \qquad \widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\mathcal{T}}} F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) \tag{1.23}$$

has explicit solutions given by

$$\widehat{\pi}_k = \frac{1}{N}\sum_{i=1}^N \tau_{i,k}, \qquad \widehat{\boldsymbol{\mu}}_k = \frac{1}{N\widehat{\pi}_k}\sum_{i=1}^N \tau_{i,k}\boldsymbol{x}_i, \qquad \forall k \in [K], \tag{1.24}$$

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{N\widehat{\pi}_k}\sum_{i=1}^N \tau_{i,k}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top, \qquad \forall k \in [K], \tag{1.25}$$

$$\widehat{\tau}_{i,k} = \frac{\pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i)}{\sum_{k'\in[K]} \pi_{k'} \varphi_{\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}}(\boldsymbol{x}_i)}, \qquad \forall k \in [K],\ \forall i \in [N]. \tag{1.26}$$

Based on this result, the EM algorithm is defined as in Figure 1.9. The algorithm operates iteratively and needs a criterion to determine when the iterations should be stopped. There is no clear consensus on this point in the statistical literature, but it is a commonly used practice to stop when one of the following conditions is fulfilled:

i) The number of iterations t exceeds a pre-specified level $t_{\max}$.

ii) The increase of the log-likelihood over past t_0 iterations is not significantly different from zero: $\ell_N(\boldsymbol{\theta}^t) - \ell_N(\boldsymbol{\theta}^{t-t_0}) \leq \varepsilon$ for some pre-specified values $t_0 \in \mathbb{N}$ and $\varepsilon > 0$.

EM is conceptually easy and each iteration increases the log-likelihood:

$$\ell_N(\boldsymbol{\theta}^{t+1}) \geq \ell_N(\boldsymbol{\theta}^t), \qquad \forall t \in \mathbb{N}.$$

The complexity at each step of the EM algorithm is $O(KNp^2)$ and it usually requires many iterations to converge. In a high-dimensional setting when p is large, the quadratic dependence on p may result in prohibitively large running times. However, the computation of the elements of the covariance matrices $\boldsymbol{\Sigma}_k^t$ and the mean vectors $\boldsymbol{\mu}_k^t$ can be parallelized which may lead to considerable savings in the running time.

1.2.2 K-means from the EM angle

In this section, we will see that the K-means problem is closely related to the EM algorithm. We rewrite the minimization problem of K-means defined in eq. (1.1) as follows

$$\min_{\boldsymbol{c}_1,\dots,\boldsymbol{c}_K} \min_{\boldsymbol{R}\in\{0,1\}^{N\times K}} \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2, \tag{1.27}$$

where, in the matrix $\boldsymbol{R}$, the rows sum to 1. One can solve this problem by repeating two steps, the first one consists in minimizing the objective function with respect to $\boldsymbol{c}_1, \dots, \boldsymbol{c}_K$ with $\boldsymbol{R}$ fixed (Maximization step) and the second one consists in minimizing the objective function with respect to $\boldsymbol{R}$ with $\boldsymbol{c}_1, \dots, \boldsymbol{c}_K$ fixed (Expectation step). Consider the **E**-step and note that the objective function is linear with respect to $\boldsymbol{R}$. It consists for a data point $\boldsymbol{x}_i$, to find the cluster k such that $k = \arg\min_{j\in[K]} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|^2$. For the **M**-step, setting the gradient with respect to $\boldsymbol{c}_k$ to 0 gives us

$$2\sum_{i=1}^{N} r_{ik}(\boldsymbol{x}_i - \boldsymbol{c}_k) = 0, \tag{1.28}$$

which leads to

$$\boldsymbol{c}_k = \frac{\sum_{i=1}^{N} r_{ik}\boldsymbol{x}_i}{\sum_{i=1}^{N} r_{ik}}. \tag{1.29}$$

Since $\sum_{i=1}^{N} r_{ik}$ is the size of the cluster k, we recovered Lloyd's algorithm.

1.3 The curse of dimensionality

The expression 'Curse of dimensionality' introduced by R. Bellman in his book on dynamic programming [Bellman, 1957] refers to the problems related to high dimension. One can see that evaluating a function on the segment $(0, 1)$ with a step size of 0.1 is straightforward. However, evaluating the function in a grid of dimension 10 requires 10^{10} computations which can be intractable even today within a reasonable time. Many computational and statistical problems arise in this setting. Sometimes the literature refers to a 'high dimensional' setting when $p \gg n$ and more precisely when the model considered has more parameters or degrees of freedom than there are observations. In the sequel, we recall some classical phenomena that appear in this context and focus on the clustering with high dimensional data.

We saw previously different clustering methods that rely on a distance such as the Euclidean distance. It turns out that in a high dimensional setting, the notion of nearest point vanishes: the minimal distance increases but on the other hand the variance of the distance between points has a slower increase. Consider 2 p-dimensional random vectors $\boldsymbol{X}, \boldsymbol{X}'$ with i.i.d. entries and the Euclidean norm, the scaled deviation is then

$$\frac{\mathrm{sdev}[\|\boldsymbol{X} - \boldsymbol{X}'\|^2]}{\mathbb{E}[\|\boldsymbol{X} - \boldsymbol{X}'\|^2]} \approx \frac{1}{\sqrt{p}}, \tag{1.30}$$

and goes to 0 when $p \to \infty$. A direct consequence of such distance concentration phenomenon is the loss of relevance of the methods based on discriminating near and far neighbors such as those studied in the previous section (nearest center for K-means, agglomeration in hierarchical clustering or constructing adjacency graph for spectral clustering). In the clustering context, a strong assumption for ensuring the separation of clusters would be to consider the inter-cluster distance dominant compared to the variance within each clusters. Another phenomenon is the "error accumulation". Consider the classical linear regression setting $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$ with $\boldsymbol{X}$ an orthogonal matrix in $\mathbb{R}^{N \times p}$ and $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N$ i.i.d. centered with variance σ^2. The least-squares estimator $\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ has an estimation error given by

$$\mathbb{E}[\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|^2] = p\sigma^2. \tag{1.31}$$

Therefore we can see that the estimation error increases linearly with the dimension. Furthermore, an interesting phenomenon that occurs in high dimension is that spaces are mostly empty and the realizations of a p-dimensional random vector with a uniform probability distribution on the unit ball lie with high probability close to a hypersphere. Therefore, the data belong mostly to a $p - 1$ dimensional subspace. Interestingly, the ratio of

the volume of a unit ball and the volume of the unit hypercube goes to 0 as $p \to \infty$ (see section 2.3 of [Zimek et al., 2012]). This means that most of the volume lies in the corner of the hypercube. Therefore, any method based on a spherical distance such as the Euclidean norm is deficient in this context. One can consider a probabilistic approach to overcome the issues with high dimension, but the naïve model-based clustering suffers over-parametrization. In the Gaussian mixture model of K components in dimension p, the number of free parameters to estimate is

$$\nu = \underbrace{(K-1)}_{\text{Weights}} + \underbrace{Kp}_{\text{Means}} + \underbrace{Kp(p-1)}_{\text{Covariances Matrices}} , \tag{1.32}$$

which for $p = 100$ and $K = 5$ is 125704. Moreover, the evaluation of $\widehat{\tau}_{i,k}$ in eq. (1.26) requires the computation of the inverse of the covariance matrix $\widehat{\mathbf{\Sigma}}_k$ which is called the precision matrix. If $n < p$ the matrices $\widehat{\mathbf{\Sigma}}_k$ with $k \in [K]$ are ill-conditioned and the precision matrices are prone to large numerical errors or more often are singular and the problem can not be solved.

Several popular methods are used to overcome these issues. One can reasonably consider that several variables are correlated or that projections on many directions are irrelevant and, therefore, clusters may live on a lower-dimensional subspace. A first approach would be to perform a dimension reduction like Principal Component Analysis (PCA) but this leads to a decoupling of the dimension reduction task from the clustering task and may lead to a poor selection of the subspace [Bouveyron and Brunet, 2013], keeping information from irrelevant dimensions. Moreover, the resulting linearly transformed dimensions are difficult to interpret. Another approach called "feature selection" consists in selecting the most relevant features but fails when clusters live in different subspaces. This scenario leads to the development of "subspace clustering" techniques that go one step further by selecting the most relevant features for each cluster separately (see [Parsons et al., 2004] for a review on this topic).

In the rest of this chapter, we discuss some approaches based on the regularization technique and make sparsity assumption on the structure of the precision matrices in the Gaussian mixture model. The goal is to reduce the number of free parameters and tackle the problem of estimating the inverse of the covariance matrix. In section 2.1, we address this challenge by studying some nice structural properties of precision matrices.

The reader can find a more thorough overview of high dimensional statistics in Giraud [2014], Zimek et al. [2012], Bühlmann and van de Geer [2011]. For a survey of clustering in high dimension, see [Bouveyron and Brunet, 2013, Parsons et al., 2004].

Chapter 2

Partial contributions to clustering

In this chapter, we present some work carried out during this thesis which have unfortunately not been able to be the subject of an in-depth study that can be published. The first part deals with the sparsity hypothesis of the precision matrices within a high dimensional Gaussian mixture and adapts the single-component Graphical Lasso from [Friedman et al., 2007] to the mixture setting. In the second part, we assume that the weight vector of the mixture is sparse in order to obtain an estimator of the number of components in the mixture that is generally unknown.

2.1 Graphical Lasso for Gaussian mixtures

As we saw in the previous sections, the number of free parameters in a full GMM with K components in dimension p are $(K-1)+Kp+Kp(p+1)/2$ which means, for instance, that for $K = 5$ and $p = 100$ we have 125704 parameters to estimate. In this high dimensional setting, the EM algorithm experiences severe performance degradation. In particular, the inversion of the covariance matrices are one source of error. One way to circumvent these problems is to use regularization. To this end, we will make a structural assumption on the inverse of the covariance matrices, called precision or concentration matrices, of a component. The work presented in this chapter is inspired by [Friedman et al., 2007], [Banerjee et al., 2008], [Yuan and Lin, 2007] and [Meinshausen and Bühlmann, 2006] where it is suggested to penalize the off-diagonal entries of the precision matrix of a Gaussian graphical model. We do an attempt to generalize this work to the Gaussian mixture model.

We consider $\boldsymbol{X} = (X^{(1)}, \ldots, X^{(p)})$, a random vector admitting a p-dimensional normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a non-singular $\boldsymbol{\Sigma}$. One can construct an undirected graph $G = (V, E)$ with p vertices corresponding to each coordinate and, $E = (e_{i,j})_{1 \leq i < j \leq p}$, the edges

between the vertices describing the conditional independence relation among $X^{(1)}, \dots, X^{(p)}$. If in this graph, $e_{i,j}$ is absent in E if and only if $X^{(i)}$ and $X^{(j)}$ are independent conditionally to the other variables $\{X^{(l)}\}$ with $l \neq i, j$ (denoted $X^{(i)} \perp\!\!\!\perp X^{(j)} | X^{(l)}\, l \neq i, j$). Then G is called the Gaussian concentration graph model for the Gaussian random vector $\boldsymbol{X}$. This representation is particularly interesting in the study of the inverse of the covariance matrix. Let us denote $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Omega} = (\omega_{i,j})$ the precision matrix. The entries of this matrix satisfy $\omega_{i,j} = 0$ if and only if $X^{(i)} \perp\!\!\!\perp X^{(j)}$ conditionally to all the other variables. We recall in the following lemma this well-known result

Lemma 2.1.1 (Conditional independence in Gaussian concentration graph model)**.** *Consider $\boldsymbol{X} = (X^{(1)}, \dots, X^{(p)})$, a p-dimensional random vector with a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and set $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Omega} = (\omega_{i,j})$. Then $X^{(i)} \perp\!\!\!\perp X^{(j)} | \{X^{(l)} : l \notin \{i, j\}\} \iff \omega_{i,j} = 0$ with $l \neq i, j$*

Proof. This result can be found in [Edwards, 2000]. Consider the density of $\boldsymbol{X}$

$$\varphi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\Big). \tag{2.1}$$

It can be rewritten as

$$\varphi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \exp(\alpha + \beta^T \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Omega} \boldsymbol{x}), \tag{2.2}$$

with $\beta = \boldsymbol{\Omega}\boldsymbol{\mu}$ and $\alpha = \frac{1}{2}\log(|\boldsymbol{\Omega}|) - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Omega}\boldsymbol{\mu} - \frac{p}{2}\log(2\pi)$. Then, the previous equation can be rewritten as

$$\exp\big(\alpha + \sum_{j=1}^{p} \beta_j x^{(j)} - \frac{1}{2}\sum_{j=1}^{p}\sum_{i=1}^{p} \omega_{i,j} x^{(j)} x^{(i)}\big). \tag{2.3}$$

Now, for three random variables X, Y, Z, we have $X \perp\!\!\!\perp Y | Z$ if and only if the joint density can be factorized into two factors $f_{X,Y,Z}(x, y, z) = h(x, z)g(y, z)$, with h anf g two functions. Then, in the light of eq. (2.3), we have $X^{(i)} \perp\!\!\!\perp X^{(j)} | \{X^{(l)} : l \notin \{i, j\}\} \iff \omega_{i,j} = 0$. □

The first result available in the literature on gaussian graphical models focused on the estimation of the graph structure. In particular [Dempster, 1972] developed a greedy forward or backward search method to estimate the set of non-zero entries in the concentration matrix. The forward method relies on initializing an empty set and selecting iteratively an edge with an MLE fit for $\mathcal{O}(p^2)$ different parameters. The procedure stops according to a suitable selection criterion. The backward method proceeds in the same manner by starting with all edges and operating deletions. It is obvious that such methods are computationally intractable in high dimension. In [Meinshausen and Bühlmann, 2006], the authors studied a neighborhood selection procedure with lasso. The goal is to estimate

the neighborhood $ne_{X^{(i)}}$ of a node $X^{(i)}$ which is the smallest subset of $G \setminus \{X^{(i)}\}$ such that $X^{(i)} \perp\!\!\!\perp \left\{X^{(j)} : X^{(j)} \in G \setminus \{ne_{X^{(i)}}\}\right\} | X_{ne_{X^{(i)}}}$. The estimation of the neighborhood is cast as a sparse regression problem and tackled with a lasso penalization. The authors show that this procedure is consistent for sparse high dimensional graphs and computationally efficient. More precisely, let $\theta^{(i)} \in \mathbb{R}^p$ be the vector of coefficients of the optimal prediction,

$$\theta^{(i)} = \underset{\theta:\theta_i=0}{\arg\min}\, \mathbb{E}\Big[\big(X^{(i)} - \sum_{k=1}^{p} \theta_k X^{(k)}\big)^2\Big], \tag{2.4}$$

then the components of $\theta^{(i)}$ are determined by the precision matrix, $\theta_j^{(i)} = -\omega_{i,j}/\omega_{i,i}$. Therefore, the set of neighbors of $X^{(i)} \in G$ is given by

$$ne_{X^{(i)}} = \{X^{(j)}, j \in [p] : \omega_{i,j} \neq 0\}. \tag{2.5}$$

Now, let $\mathbb{X}$ be the $N \times p$-dimensional matrix such that the column $\mathbb{X}^{(i)}$ is the vector formed by N of $X^{(i)}$. Given a suitably chosen regularization parameter $\lambda \geq 0$, the Lasso estimate $\widehat{\theta}^{i,\lambda}$ of $\theta^{(i)}$ is defined as

$$\widehat{\theta}^{i,\lambda} = \underset{\theta:\theta_i=0}{\arg\min} \Big(\frac{1}{N}\|\mathbb{X}^{(i)} - \mathbb{X}\theta\|_2^2 + \lambda\|\theta\|_1\Big). \tag{2.6}$$

The authors prove under several assumptions that

$$P(\widehat{ne}^{\lambda}_{X^{(i)}} = ne_{X^{(i)}}) \to 1 \quad \text{when} \quad N \to \infty, \tag{2.7}$$

and for some $\epsilon > 0$,

$$P(\widehat{E}^{\lambda} = E) = 1 - \mathcal{O}(\exp(-cN^{\epsilon})) \quad \text{when} \quad N \to \infty, \tag{2.8}$$

where E^{λ} in a estimate of the edge set. Therefore, this method recovers the conditional independence structure of sparse high-dimensional Gaussian concentration graph at exponential rates. One can estimate the parameters of the model which has been selected by this method using, for instance ordinary least squares. Such a procedure often suffers from the instability of the estimator since small changes in the data change the selected model [Yuan and Lin, 2007, Breiman, 1996]. One difficulty of a method that would perform both tasks is to ensure that the estimator of the precision matrix is positive definite. [Yuan and Lin, 2007] proposed a penalized-likelihood method that performs model selection and parameter estimation simultaneously as well as ensures the positive definiteness of the precision matrix. Their approach is similar to [Meinshausen and Bühlmann, 2006] as they use

the ℓ_1 penalty, they additionally impose a positive definiteness constraint. Furthermore, they replace the residual sum of squares by the negative log-likelihood,

$$-\Big\{\frac{N}{2}\log(|\boldsymbol{\Omega}|)-\frac{1}{2}\sum_{i=1}^{N}\boldsymbol{X}_i^T\boldsymbol{\Omega}\boldsymbol{X}_i\Big\}. \tag{2.9}$$

The resulting constrained minimization problem over the set of positive definite matrices is

$$\min\Big\{-\log(|\boldsymbol{\Omega}|)+\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}_i^T\boldsymbol{\Omega}\boldsymbol{X}_i\Big\}\quad\text{subject to}\quad\sum_{i\neq j}|\omega_{i,j}|\leq t\quad\text{and}\quad\boldsymbol{\Omega}\succeq 0, \tag{2.10}$$

with $t\geq 0$ a tuning parameter. In these formulae we assume that the mean of the Gaussian distribution is known to be equal to 0. Consider the empirical covariance matrix $\boldsymbol{S}=1/N\sum_{i=1}^{N}\boldsymbol{X}_i\boldsymbol{X}_i^T$, eq. (2.10) can be rewritten as

$$\min\Big\{-\log(|\boldsymbol{\Omega}|)+\operatorname{tr}(\boldsymbol{S}\boldsymbol{\Omega})\Big\}\quad\text{subject to}\quad\sum_{i\neq j}|\omega_{i,j}|\leq t. \tag{2.11}$$

Since the whole problem is convex, the Lagrangian is given by

$$\mathcal{L}(\lambda,\boldsymbol{\Omega})=-\log(|\boldsymbol{\Omega}|)+\operatorname{tr}(\boldsymbol{S}\boldsymbol{\Omega})+\lambda\sum_{i\neq j}|\omega_{i,j}|, \tag{2.12}$$

where λ is a tuning parameter. A non-negative garrote-type estimator is provided in [Yuan and Lin, 2007] which requires a good initial estimator of $\boldsymbol{\Omega}$. The authors provided an asymptotic result:

Theorem 2.1.1 (Theorem 1 from [Yuan and Lin, 2007]). *If $\sqrt{N}\lambda\to\lambda_0\geq 0$ as $N\to\infty$, the lasso-type estimator satisfies*

$$\sqrt{N}(\widehat{\boldsymbol{\Omega}}-\boldsymbol{\Omega})\to\underset{\boldsymbol{U}=\boldsymbol{U}^T}{\arg\min}(\mathcal{V}(\boldsymbol{U})),$$

where the convergence is in distribution and $\mathcal{V}$ is defined by the formula

$$\mathcal{V}(\boldsymbol{U})=\operatorname{tr}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}\boldsymbol{\Sigma})+\operatorname{tr}(\boldsymbol{U}\boldsymbol{W})+\lambda_0\sum_{i\neq j}\big\{u_{i,j}\operatorname{sign}(\omega_{i,j})I(\omega_{i,j}\neq 0)+|u_{i,j}|I(\omega_{i,j}=0)\big\}$$

in which $\boldsymbol{W}$ is a random symmetric $p\times p$ matrix such that $\operatorname{vec}(\boldsymbol{W})\sim\mathcal{N}(0,\Lambda)$, and Λ is such that

$$\operatorname{cov}(w_{i,j},w_{i',j'})=\operatorname{cov}(X^{(i)}X^{(j)},X^{(i')}X^{(j')}).$$

Unfortunately, the computational complexity of interior point methods for maximizing eq. (2.12) is $\mathcal{O}(p^6)$ and at each step, we have to compute and store a Hessian matrix of size $\mathcal{O}(p^2)$. These prohibitively large complexities led the research on more specialized methods. [Banerjee et al., 2008] worked on the same approach, solving a maximum likelihood problem with an ℓ_1 penalty and focusing on the computational complexity by proposing an iterative block coordinate descent algorithm. The problem to maximize is similar to eq. (2.12)

$$\widehat{\boldsymbol{\Omega}} = \arg\max_{\boldsymbol{\Omega}\succ 0}\{\log(|\boldsymbol{\Omega}|) - \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Omega}) - \lambda\|\boldsymbol{\Omega}\|_1\}. \tag{2.13}$$

Note that the ℓ_1 norm of a matrix $\boldsymbol{\Omega}$ can be expressed as

$$\|\boldsymbol{\Omega}\|_1 = \max_{\|\boldsymbol{U}\|_\infty \leq 1} \operatorname{tr}(\boldsymbol{\Omega}\boldsymbol{U}), \tag{2.14}$$

injecting this in eq. (2.13) gives

$$\max_{\boldsymbol{\Omega}\succ 0} \min_{\|\boldsymbol{U}\|_\infty \leq \lambda} \big\{ \log(|\boldsymbol{\Omega}|) - \operatorname{tr}(\boldsymbol{\Omega}(\boldsymbol{S}+\boldsymbol{U}))\big\}. \tag{2.15}$$

After exchanging the min and the max, we solve the problem for $\boldsymbol{\Omega}$ by setting the gradient to 0. This gives $(\boldsymbol{\Omega}^{-1})^T - (\boldsymbol{S}+\boldsymbol{U})^T = 0$ yielding $\boldsymbol{\Omega} = (\boldsymbol{S}+\boldsymbol{U})^{-1}$. The dual problem is then

$$\min_{\|\boldsymbol{U}\|_\infty} \{-\log(|\boldsymbol{S}+\boldsymbol{U}|) - p\}, \tag{2.16}$$

or by setting $\boldsymbol{W} = \boldsymbol{S}+\boldsymbol{U}$,

$$\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Omega}^{-1}} = \arg\max \log(|\boldsymbol{W}|) \quad \text{s.t} \quad \|\boldsymbol{W}-\boldsymbol{S}\|_\infty \leq \lambda. \tag{2.17}$$

We observe the presence of a log-barrier adding the implicit constraint $(\boldsymbol{S}+\boldsymbol{U}) \succ 0$. Furthermore, the dual problem estimates the covariance matrix. To solve this maximization problem, the authors proposed a Block Coordinate Descent Algorithm that we describe below (see also Figure 2.1).

It can be proved that the Block Coordinate Descent algorithm converges [Banerjee et al., 2008], achieving an ε-suboptimal solution to eq. (2.17) and each iterate produces a strictly positive definite matrix. For a fixed number of sweeps K, the complexity of this algorithm is $\mathcal{O}(Kp^4)$. They provide also another algorithm using Nesterov's first order method which has a $\mathcal{O}(p^{4.5}/\epsilon)$ complexity for $\varepsilon > 0$, the desired accuracy. For any symmetric matrix $\boldsymbol{A}$, let $\boldsymbol{A}_{\backslash k \backslash j}$ be the matrix produced by removing column k and row j from $\boldsymbol{A}$. Let $\boldsymbol{A}_j$ the j^{th} column of $\boldsymbol{A}$ with the element $\boldsymbol{A}_{jj}$ removed. It is interesting to note that the dual problem of Line 6 in fig. 2.1 is

$$\min_{\boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{W}^{(j-1)}_{\backslash j \backslash j} \boldsymbol{x} - \boldsymbol{S}_j^T \boldsymbol{x} + \lambda\|\boldsymbol{x}\|_1, \tag{2.18}$$

Input: Matrix $\boldsymbol{S}$, parameter λ and threshold ε
Output: Estimate of $\boldsymbol{W}$
Initialize $\boldsymbol{W}^{(0)} := \boldsymbol{S} + \lambda I$
repeat
  for $j = 1, \ldots, p$ **do**
    (a) Let $\boldsymbol{W}^{(j-1)}$ denote the current iterate. Solve the quadratic program

$$\widehat{\mathbf{y}} := \arg\min_{\mathbf{y}} \{\mathbf{y}^T (\boldsymbol{W}^{(j-1)}_{\setminus j \setminus j})^{-1} \mathbf{y} : \|\mathbf{y} - \boldsymbol{S}_j\|_\infty \leq \lambda\}.$$

    (b) Update the rule: $\boldsymbol{W}^{(j)}$ is $\boldsymbol{W}^{(j-1)}$ with column/row $\boldsymbol{W}_j$ replaced by $\widehat{\mathbf{y}}$.
  end for
  Let $\widehat{\boldsymbol{W}}^{(0)} := \boldsymbol{W}^{(p)}$.
until convergence occurs when

$$\operatorname{tr}\big((\widehat{\boldsymbol{W}}^{(0)})^{-1}\boldsymbol{S}\big) - p + \lambda \big\|(\widehat{\boldsymbol{W}}^{(0)})^{-1}\big\|_1 \leq \varepsilon.$$

Figure 2.1: Block Coordinate Descent Algorithm

and strong duality holds, it can best casted as

$$\min_{\boldsymbol{x}} \|\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \|\boldsymbol{x}\|_1, \tag{2.19}$$

with $\boldsymbol{Q} = (\boldsymbol{W}^{(j-1)}_{\setminus j \setminus j})^{1/2}$ and $\boldsymbol{b} := \frac{1}{2}\boldsymbol{Q}^{-1}\boldsymbol{S}_j$. Therefore, we recover the Lasso problem. More precisely, the algorithm can be interpreted as a sequence of iterative Lasso problems. This approach is similar to another paper that we would like to mention [Friedman et al., 2007]. The authors proposed a faster algorithm based on the Block Coordinate Descent algorithm from [Banerjee et al., 2008] called Graphical Lasso. They estimate the matrix $\boldsymbol{W} = \boldsymbol{\Omega}^{-1}$ by performing iterative permutations of the columns of this matrix to make the target column the last for a coupled Lasso problem. The matrices $\boldsymbol{W}$ and $\boldsymbol{S}$ will be presented as following

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{11} & \boldsymbol{w}_{12} \\ \boldsymbol{w}_{21} & w_{22} \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_{11} & \boldsymbol{s}_{12} \\ \boldsymbol{s}_{21} & s_{22} \end{bmatrix}, \tag{2.20}$$

and the Graphical Lasso algorithm is described in Figure 2.2. The Lasso problem can be solved via a coordinate descent, the reader can refer to [Friedman et al., 2007] for the procedure. In this problem, the algorithm estimates $\widehat{\boldsymbol{\Sigma}}$ and returns also $\boldsymbol{B} = (\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(p)})$, the matrix where each column is the solution of the Lasso problem in eq. (2.19) for each

Input: Matrix $\boldsymbol{S}$, parameter λ and threshold ε
Output: Estimate of $\boldsymbol{W}$ and $\boldsymbol{B}$ a matrix of parameters.
Initialize $\boldsymbol{W}^{(0)} := \boldsymbol{S} + \lambda I$ and $\boldsymbol{B} = 0_{p\times p}$. The diagonal of $\boldsymbol{W}$ remained unchanged in what follows.
repeat
  for $j = 1, \ldots, p$ **do**
    (a) Let $\boldsymbol{W}^{(j-1)}$ denote the current iterate. Solve the Lasso problem in eq. (2.19)

$$\widehat{\boldsymbol{x}}^{(j-1)} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|(\boldsymbol{W}_{11}^{(j-1)})^{1/2}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{2.22}$$

    with $\boldsymbol{b} := (\boldsymbol{W}_{11}^{(j-1)})^{-1/2}\boldsymbol{s}_{12}$.
    (b) Update: $\boldsymbol{W}^{(j)}$ is $\boldsymbol{W}^{(j-1)}$ with $\boldsymbol{w}_{12} = \boldsymbol{W}_{11}^{(j-1)}\widehat{\boldsymbol{x}}^{(j-1)}$.
    (c) Save the parameter $\boldsymbol{x}^{(j-1)}$ in the j^{th} column of $\boldsymbol{B}$.
    (d) Permute the columns and rows of $\boldsymbol{W}^{(j-1)}$ such that the j^{th} column is $\boldsymbol{w}_{12}$, the next target.
  end for
  Let $\widehat{\boldsymbol{W}}^{(0)} := \boldsymbol{W}^{(p)}$.
until convergence occurs.

Figure 2.2: The Graphical Lasso from [Friedman et al., 2007].

column of $\boldsymbol{W}$. It is easy then to recover $\boldsymbol{\Omega}$ since

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{11} & \boldsymbol{w}_{12} \\ \boldsymbol{w}_{21}^T & w_{22} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\Omega}_{11} & \boldsymbol{\omega}_{12} \\ \boldsymbol{\omega}_{21}^T & \omega_{22} \end{bmatrix} = \begin{bmatrix} I_{p-1} & \mathbf{0} \\ \mathbf{0^T} & 1 \end{bmatrix}, \tag{2.21}$$

and

$$\begin{aligned} \boldsymbol{\omega}_{12} &= -\boldsymbol{W}_{11}^{-1}\boldsymbol{w}_{12}\omega_{22} \\ \omega_{22} &= 1/(w_{22} - \boldsymbol{w}_{12}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{w}_{12}). \end{aligned}$$

Therefore, for $j = 1, \ldots, p$, the permuted target components of $\boldsymbol{\Omega}$ are

$$\begin{aligned} \boldsymbol{\omega}_{12} &= -\boldsymbol{b}^{(j)}\widehat{\omega}_{22} \\ \omega_{22} &= 1/(w_{22} - \boldsymbol{w}_{12}^T\boldsymbol{b}^{(j)}). \end{aligned}$$

In what follows, we will adapt these methods to a Gaussian mixture model. More precisely, we will assume that each cluster is associated with a sparse Gaussian concentration graph. We will rely on the Graphical Lasso for estimating the precision matrix and will derive an EM algorithm for estimating the model parameters.

2.1.1 Graphical Lasso on Gaussian mixtures

In this part, we present our contribution. We consider a Gaussian mixture model of K components and our task is to estimate the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ with $\theta_k = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)$, where $\boldsymbol{\Omega}_k$ is the precision matrix of the k^{th} component of the mixture. We denote $\varphi_{(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)}$ the Gaussian density of mean $\boldsymbol{\mu}_k$ and precision matrix $\boldsymbol{\Omega}_k$. The penalized log-likelihood is

$$\ell_N^{pen}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - pen(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)}(\boldsymbol{x}_i) \right\} - pen(\boldsymbol{\theta}). \tag{2.23}$$

We suppose that each component of the mixture has a sparse Gaussian concentration graph. Therefore, in the spirit of [Banerjee et al., 2008] and [Friedman et al., 2007], we consider an ℓ_1 regularization $pen(\theta_k) = \sum_{k=1}^{K} \lambda_k ||\boldsymbol{\Omega}_k||_{1,1}$ with $\lambda_k > 0$. The penalization of the log-likelihood concerns only the precision matrices $\boldsymbol{\Omega}_k$. Regarding the other parameters $(\pi_k, \boldsymbol{\mu}_k)$, our algorithm is the same as EM and we can use the same iteration technique as in Lemma 1 to maximize the following cost function

$$F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) = \sum_{k=1}^{K} \left(\sum_{i=1}^{N} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\} - \lambda_k ||\boldsymbol{\Omega}_k||_{1,1} \right). \tag{2.24}$$

The maximization of this function over $\boldsymbol{\theta}$ and $\boldsymbol{\mathcal{T}}$ leads to the two following optimization problems:

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) \in \arg\max_{\boldsymbol{\theta}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}), \qquad \widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\mathcal{T}}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}). \tag{2.25}$$

For a given $\widehat{\boldsymbol{\mathcal{T}}}$, estimates of $(\pi_1, \ldots, \pi_K)$ and $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ obtained by the first optimization problem in eq. (2.25) are the same as in the EM algorithm:

$$\widehat{\pi}_k = \frac{1}{N} \sum_{i=1}^{N} \widehat{\tau}_{i,k}, \quad \text{and} \quad \widehat{\boldsymbol{\mu}}_k = \frac{1}{N\widehat{\pi}_k} \sum_{i=1}^{N} \widehat{\tau}_{i,k} \boldsymbol{x}_i \qquad \forall k \in [K], \tag{2.26}$$

and for a given $\widehat{\boldsymbol{\theta}}$, the estimate of $\boldsymbol{\mathcal{T}}$ obtained by the second optimization problem is

$$\widehat{\tau}_{i,k} = \mathrm{P}_{\widehat{\boldsymbol{\theta}}}(Z = k | \boldsymbol{X} = \boldsymbol{x}_i) = \frac{\widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k}(\boldsymbol{x}_i)}{\sum_{k' \in [K]} \widehat{\pi}_{k'} \varphi_{\widehat{\boldsymbol{\mu}}_{k'}, \widehat{\boldsymbol{\Omega}}_{k'}}(\boldsymbol{x}_i)}, \qquad \forall k \in [K],\ \forall i \in [N]. \tag{2.27}$$

However, due to the penalty $\lambda_k ||\boldsymbol{\Omega}_k||_{1,1}$, the estimation of $\boldsymbol{\Omega}_k$ is not straightforward. To overcome this problem, let us introduce the weighted empirical covariance matrix

$$\boldsymbol{\Sigma}_{N,k} = \frac{1}{N} \frac{\sum_{i=1}^{N} \tau_{i,k} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top}{\sum_{i=1}^{N} \tau_{i,k}}. \tag{2.28}$$

The penalized log-likelihood in equation (2.24) can therefore be expanded as follows

$$\begin{aligned} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) =& \sum_{k=1}^{K} \Bigg(\sum_{i=1}^{N} \Big\{ \tau_{i,k} \Big(-\frac{p}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Omega}_k| - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \Big) \\ & + \tau_{i,k}\log(\pi_k/\tau_{i,k}) \Big\} - \lambda_k ||\boldsymbol{\Omega}_k||_{1,1} \Bigg) \\ =& -\frac{Np}{2}\log(2\pi) + \sum_{k=1}^{K} \Bigg(\frac{N\pi_k}{2}\log|\boldsymbol{\Omega}_k| \\ & + \sum_{i=1}^{N} \Big\{ -\frac{\tau_{i,k}}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}_k (\boldsymbol{x}_i - \boldsymbol{\mu}_k) + \tau_{i,k}\log(\pi_k/\tau_{i,k}) \Big\} - \lambda_k ||\boldsymbol{\Omega}_k||_{1,1} \Bigg). \end{aligned}$$

Hence, the opposite minimization problem regarding each $\boldsymbol{\Omega}_k$ is

$$\boldsymbol{\Omega}_k \in \underset{\boldsymbol{\Omega} \succeq 0}{\arg\min} \Big\{ -\frac{N\pi_k}{2}\log|\boldsymbol{\Omega}| + \frac{1}{2}\sum_{i=1}^{N} \tau_{i,k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) + \lambda_k ||\boldsymbol{\Omega}||_{1,1} \Big\}. \tag{2.29}$$

And using the well-known commutativity property of the trace operator and dividing by $N\pi_k$ gives

$$\boldsymbol{\Omega}_k \in \underset{\boldsymbol{\Omega} \succeq 0}{\arg\min} \Big\{ -\frac{1}{2}\log|\boldsymbol{\Omega}| + \frac{1}{2} tr(\boldsymbol{\Sigma}_{N,k}\boldsymbol{\Omega}) + \frac{\lambda_k}{N\pi_k} ||\boldsymbol{\Omega}||_{1,1} \Big\}. \tag{2.30}$$

In the light of this equation, one can notice that we solve a graphical lasso problem within each cluster. This minimization problem is convex and can be solved with a block coordinate ascent algorithm as described in [Mazumder, 2012]. This results in an EM-like alternating minimization procedure summarized in Figure 2.3.

Experimental evaluation

We created a mixture of $K \in \{2, 4, 10, 20, 50\}$ Gaussian components with equally distributed weights. The centers of the Gaussian densities reside on the nodes of the p-dimensional unit hypercube. We considered two simple structures of the precision matrices:

1. A multiple of the identity matrix ($10^{-3} I_p$).
2. The sum of the identity matrix with a matrix the upper and lower parts of which has a diagonal of ones at the middle.

We sampled $N \in \{100, 500, 1000\}$ points from these densities in dimension $p \in \{2, 10, 50\}$ and compared the estimation error $\|\widehat{\boldsymbol{\Omega}}_k - \boldsymbol{\Omega}^*\|_F$ of EM and our algorithm, where $\|.\|_F$ is the Frobenius norm. We ran 100 simulations for each scenarios.

Input: Observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K.
Output: Parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k, \widehat{\pi}_k\}_{k\in[K]}$
1: Initialize $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
for $t = 1, \ldots,$ until convergence occurs, **do**
2: Update the parameter $\boldsymbol{\mathcal{T}}$:

$$\tau_{i,k}^t = \frac{\pi_k^t \varphi_{\boldsymbol{\mu}_k^t, \boldsymbol{\Omega}_k^t}(\boldsymbol{x}_i)}{\sum_{k'\in[K]} \pi_{k'}^t \varphi_{\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Omega}_{k'}^t}(\boldsymbol{x}_i)}.$$

3: Update the parameter $\boldsymbol{\theta}$ for each component:

$$\pi_k^{t+1} = \frac{1}{N}\sum_{i=1}^N \tau_{i,k}^t,$$

$$\boldsymbol{\mu}_k^{t+1} = \frac{1}{N\pi_k^{t+1}}\sum_{i=1}^N \tau_{i,k}^t \boldsymbol{x}_i$$

$$\boldsymbol{\Sigma}_{n,k} = \frac{1}{N^2\pi_k^{t+1}}\sum_{i=1}^N \tau_{i,k}^{t+1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k^{t+1})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k^{t+1})^\top$$

$$\boldsymbol{\Omega}_k^{t+1} \in \underset{\boldsymbol{\Omega}\succeq 0}{\arg\min} \Big\{ -\frac{1}{2}\log|\boldsymbol{\Omega}| + \frac{1}{2} tr(\boldsymbol{\Sigma}_{N,k}\boldsymbol{\Omega}) + \frac{\lambda_k}{n\pi_k^{t+1}} ||\boldsymbol{\Omega}||_{1,1} \Big\}$$

end for

Figure 2.3: Graphical lasso algorithm for Gaussian mixtures.

In the first scenario, with the scaled identity matrix as precision matrix, the estimation error $\|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}^*\|_F$ of our algorithm is slightly better than EM, as shown in Figure 2.4 and Figure 2.5. The increase of the dimension accentuates the gap between EM and our algorithm. However, the increase of the number of clusters does not have a large impact on the error.

In the second scenario, the estimation error of our algorithm is smaller than EM except in one case, in the middle graph of Figure 2.6. In this experiment we wanted to see the behavior of our algorithm in a non-high-dimensional regime when $p = 10$, $N = 500$, $K = 4$ and it turns out that EM behave better. But as the number of clusters increases (*i.e.* the complexity), the error of EM get worse. Finally, in a very high dimensional setting ($p = 50$, $N = 1000$, $K \in \{20, 50\}$), the estimation error of the graphical lasso on GMM algorithm is much better than EM.

2.2 Estimating the number of clusters

In this section, we will focus on the problem of estimating the number of clusters, K, in a Gaussian mixture model. Most of popular clustering methods such as K-Means, Expectation-Maximization with Gaussian mixture model or spectral clustering need this parameter in input. Various methods are used to perform a selection of the best model according to a given criterion. As we saw in Section 1.1.4,a common approach is to perform multiple clusterings with the parameter K ranging from say 2 to K_{max}, where K_{max} is the maximum number of clusters we assume are present in the dataset, and to select the best model according to some prescribed criterion. In this work, we seek to incorporate the model selection step into the estimation step by means of an "adaptive" sparse estimation of the weight vector of the Gaussian components.

2.2.1 First method: regularizing the posterior probabilities

Our first approach was to penalize the matrix of posteriors $\boldsymbol{\mathcal{T}}$ defined in Section 1.2.1. Given N p-dimensional observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, let us consider the EM algorithm with a given maximal number of clusters K_{max}. The idea of this method is to add a regularization term on the estimation of the $N \times K_{max}$ matrix $\boldsymbol{\mathcal{T}}$, the component $\tau_{i,j}$ of which, we recall, is defined as

$$\tau_{i,j} = p_{\boldsymbol{\theta}}(Z = j | \boldsymbol{X} = \boldsymbol{x}_i) = \frac{\pi_j \varphi_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)}{\sum_{k=1}^{K_{max}} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i)}. \tag{2.31}$$

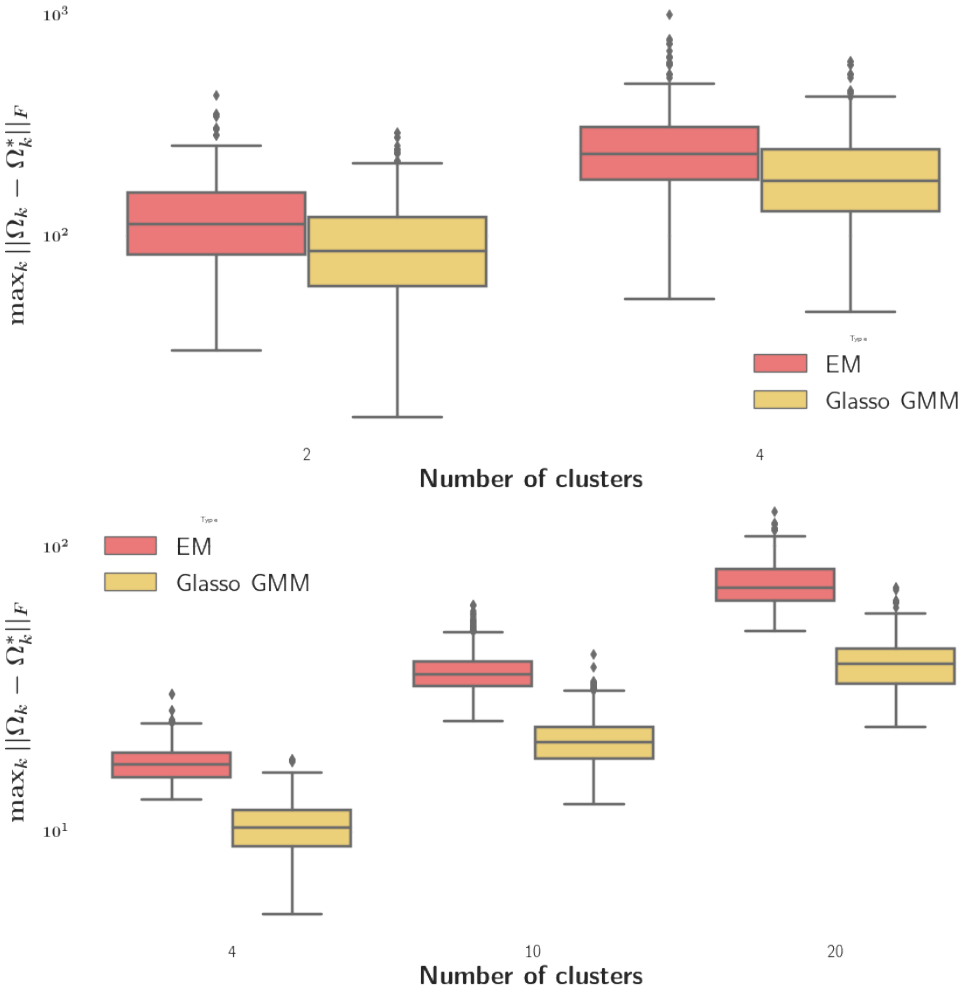

Figure 2.4: Estimation error $\max_k \|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^*\|_F$ in log scale for EM and graphical lasso on GMM where $\mathbf{\Omega}_k^* = 10^{-3} I_p$. With $p = 2$, $N = 100$ (upper graph) and $p = 5$, $N = 1000$ (lower graph).

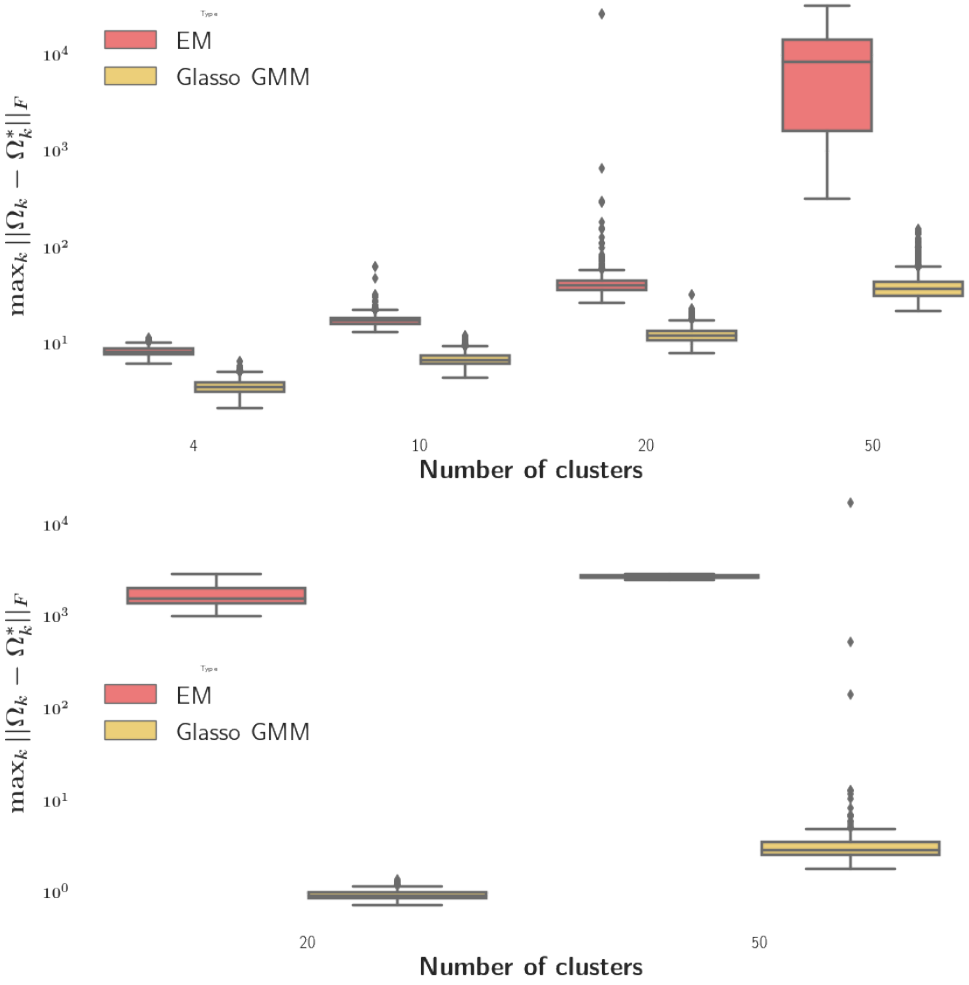

Figure 2.5: Estimation error $\max_k \|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^*\|_F$ in log scale for EM and graphical lasso on GMM where $\mathbf{\Omega}_k^* = 10^{-3} I_p$. With $p = 10$, $N = 1000$ (upper graph) and $p = 50$, $N = 1000$ (lower graph).

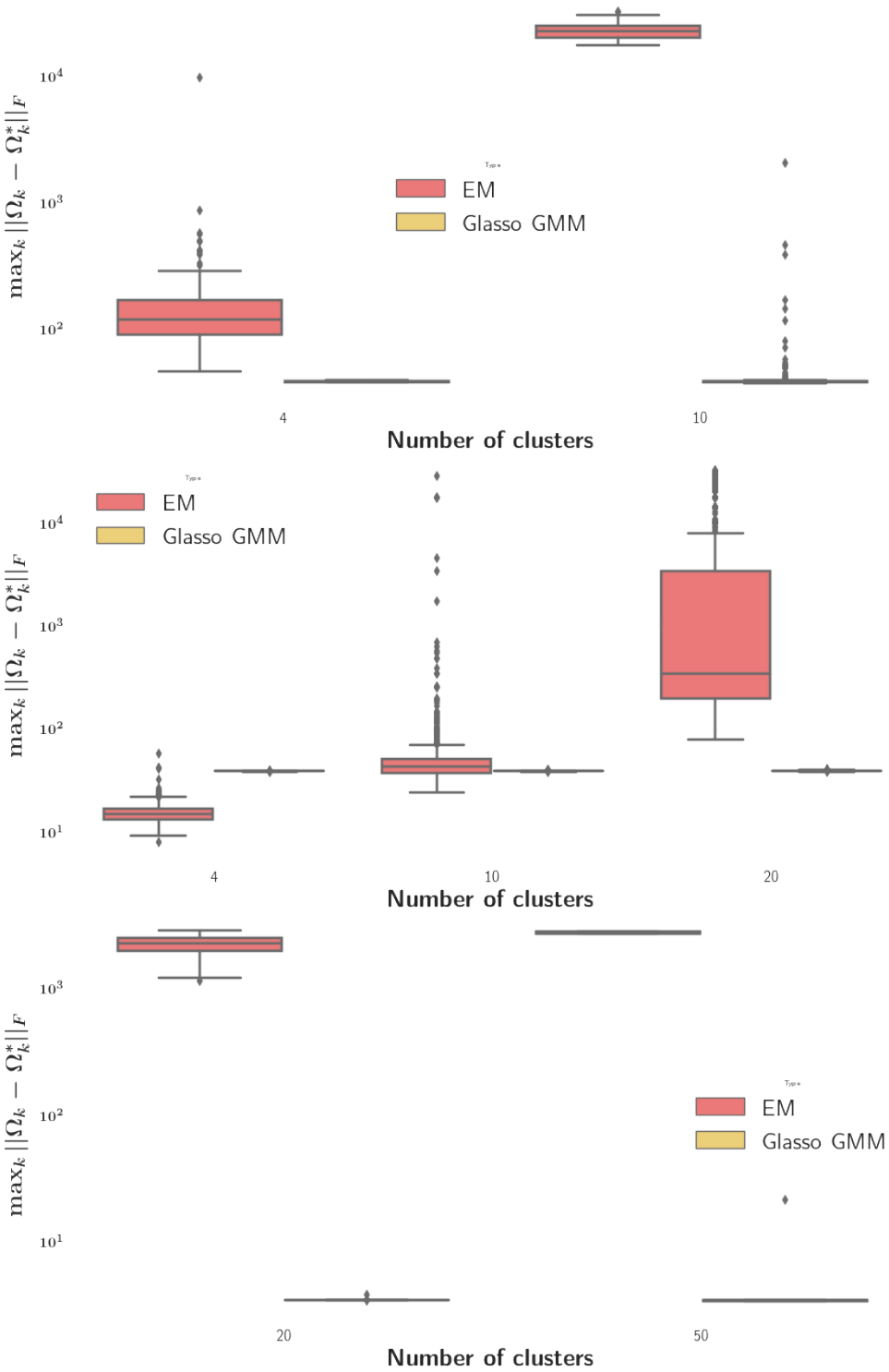

Figure 2.6: Estimation error $\max_k \|\widehat{\mathbf{\Omega}}_k - \mathbf{\Omega}_k^*\|_F$ in log scale for EM and graphical lasso on GMM where $\mathbf{\Omega}^*$ is the sum of the identity matrix with a matrix the upper and lower parts of which has a diagonal of ones at the middle. With $p = 10$, $N = 100$ (upper graph), $p = 10$, $N = 500$ (middle graph) and $p = 50$, $N = 1000$ (lower graph).

The estimated number of clusters $\widehat{K}$ will be the number of non-zero columns of $\boldsymbol{\mathcal{T}}$. Let us consider the probability simplex in $\mathbb{R}^{K_{max}}$, $\boldsymbol{\Pi} \coloneqq \{\boldsymbol{\tau} \in \mathbb{R}^{K_{max}} : \sum_{k=1}^{K_{max}} \tau_k = 1, \tau_k \geq 0 \quad \forall k \in [K_{max}]\}$ and the indicator function $\chi_{\boldsymbol{\Pi}}(.)$ defined as

$$\chi_{\boldsymbol{\Pi}}(\boldsymbol{\tau}) = \begin{cases} 0 & \text{if } \boldsymbol{\tau} \in \boldsymbol{\Pi}, \\ +\infty & \text{elsewhere.} \end{cases}$$

We note $\boldsymbol{\mathcal{T}}_{.,k}$ the k^{th} column and $\boldsymbol{\mathcal{T}}_{i,.}$ the i^{th} row of $\boldsymbol{\mathcal{T}}$. Using the cost function in Equation (1.22) and adding a penalization term composed of the L_2-norms of the columns of $\boldsymbol{\mathcal{T}}$ with a parameter $\lambda > 0$, we have the following cost function for our problem:

$$\begin{aligned} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) = &- \sum_{k=1}^{K} \left(\sum_{i=1}^{n} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\frac{\pi_k}{\tau_{i,k}}) \right\} \right) \\ &+ \lambda \sum_{k=1}^{K} \|\boldsymbol{\mathcal{T}}_{.,k}\|_2 + \sum_{i=1}^{n} \chi_{\boldsymbol{\Pi}}(\boldsymbol{\mathcal{T}}_{i,.}). \end{aligned}$$

The expectation step in an EM-like algorithm for this cost function leads to the following optimization problem:

$$\widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\min_{\boldsymbol{\mathcal{T}}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}). \tag{2.32}$$

A nice property of this problem is that it is convex. Unfortunately, the regularization term does not allow us to derive an explicit solution. Furthermore, the objective function is not decomposable since we optimize along columns and rows of $\boldsymbol{\mathcal{T}}$. The objective function $F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}})$ rewritten $F^{pen}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}})$ can be split into two terms:

$$F^{pen}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) = f(\boldsymbol{\mathcal{T}}) + g(\boldsymbol{\mathcal{T}}) \tag{2.33}$$

with

$$\begin{aligned} f(\boldsymbol{\mathcal{T}}) &= - \sum_{k=1}^{K} \left(\sum_{i=1}^{n} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\} + \sum_{k=1}^{K} \|\boldsymbol{\mathcal{T}}_{.,k}\|_2, \right. \\ g(\boldsymbol{\mathcal{T}}) &= \sum_{i=1}^{n} \chi_A(\boldsymbol{\mathcal{T}}_{i,.}). \end{aligned}$$

Note that f is convex and differentiable on its domain, g is also convex but not differentiable. We will tackle this problem by using a proximal method (see [Parikh and Boyd, 2014] for more details), proximal gradient descent. It is an iterative method the $(k+1)^{th}$

Input: Parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$
Output: Estimate $\widehat{\boldsymbol{\mathcal{T}}}$
`1: Initialize` $t_0 = 1$ `and` $\boldsymbol{\xi}^0$ `with`

$$\xi^0_{i,k} = \frac{\pi^0_k \varphi_{\boldsymbol{\mu}^0_k, \boldsymbol{\Omega}^0_k}(\boldsymbol{x}_i)}{\sum_{k' \in [K]} \pi^0_{k'} \varphi_{\boldsymbol{\mu}^0_{k'}, \boldsymbol{\Omega}^0_{k'}}(\boldsymbol{x}_i)}$$

for $k \geq 1$, until convergence occurs, **do**
(a) $\boldsymbol{\mathcal{T}}^k = \arg\min_{\boldsymbol{\mathcal{T}}: \forall i, \boldsymbol{\mathcal{T}}_{i,\cdot} \in \boldsymbol{\Pi}} \left(\|\boldsymbol{\mathcal{T}} - (\boldsymbol{\xi}^k - \eta \nabla f(\boldsymbol{\xi}^k))\|_2^2 \right)$,
(b) $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
(c) $\boldsymbol{\xi}_{k+1} = \boldsymbol{\mathcal{T}}^k + \left(\frac{t_k - 1}{t_{k+1}} \right) (\boldsymbol{\mathcal{T}}^k - \boldsymbol{\mathcal{T}}^{k-1})$.
end for

Figure 2.7: Expectation step, estimation of $\boldsymbol{\mathcal{T}}$ with FISTA.

step of which is given by

$$\begin{aligned} \boldsymbol{\mathcal{T}}^{k+1} &= \mathbf{prox}_g(\boldsymbol{\mathcal{T}}^k - \eta \nabla f(\boldsymbol{\mathcal{T}}^k)) \\ &= \mathrm{P}_{\boldsymbol{\Pi}}(\boldsymbol{\mathcal{T}}^k - \eta \nabla f(\boldsymbol{\mathcal{T}}^k)) \\ &= \underset{\boldsymbol{\mathcal{T}}: \forall i, \boldsymbol{\mathcal{T}}_{i,\cdot} \in \boldsymbol{\Pi}}{\arg\min} \left(\|\boldsymbol{\mathcal{T}} - (\boldsymbol{\mathcal{T}}^k - \eta \nabla f(\boldsymbol{\mathcal{T}}^k))\|_2^2 \right), \end{aligned}$$

where $\mathrm{P}_{\boldsymbol{\Pi}}$ is the projection function on $\boldsymbol{\Pi}$. The gradient of f on $\boldsymbol{\mathcal{T}}$ is given by

$$\left[\nabla_{\boldsymbol{\mathcal{T}}} f(\boldsymbol{\mathcal{T}}) \right]_{i,j} = 1 + \frac{\tau_{i,j}}{||\boldsymbol{\mathcal{T}}_{\cdot,j}||_2} - \log(\varphi_{\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j}(\boldsymbol{x}_i)) - \log(\frac{\pi_j}{\tau_{i,j}}) \tag{2.34}$$

We use the algorithm FISTA [Beck and Teboulle, 2009] for minimizing $f + g$. We provide more details on this method in Section 4.2. The implementation of this method is given in Figure 2.7, the whole EM-like method for the estimation of the parameters of the GMM is given in Figure 2.8.

A major drawback of this method is the computational cost of minimizing over the set of $N \times K_{max}$ matrices and we didn't manage to get interesting results for this algorithm.

2.2.2 Second method: penalizing the weight vector

The second approach that we took in order to have an estimate of the number of clusters was to penalize the weight vector of the Gaussian mixture. The idea is similar to the

Input: Observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and the number of clusters K
Output: Parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k, \pi_k\}_{k\in[K]}$
1: Initialize $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
2: **Repeat**
3: Update the parameter $\boldsymbol{\mathcal{T}}$ by using the procedure given in Figure 2.7.
4: Update the parameter $\boldsymbol{\theta}$:

$$\pi_k^{t+1} = \frac{1}{N}\sum_{i=1}^{N} \tau_{i,k}^t, \qquad \boldsymbol{\mu}_k^{t+1} = \frac{1}{N\pi_k^{t+1}}\sum_{i=1}^{N} \tau_{i,k}^t \boldsymbol{x}_i,$$

$$\boldsymbol{\Sigma}_k^{t+1} = \frac{1}{N\pi_k^{t+1}}\sum_{i=1}^{N} \tau_{i,k}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})^\top.$$

5: increment t: $t = t + 1$.
6: **Until** stopping rule.
7: **Return** $\boldsymbol{\theta}^t$.

Figure 2.8: EM algorithm with penalization on the columns of $\boldsymbol{\mathcal{T}}$.

previous method with an EM-like algorithm maximizing a penalized log-likelihood. Let $\lambda > 0$ be a tuning parameter and consider the following negative penalized log-likelihood:

$$\ell_N(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \log\left\{\sum_{j=1}^{K} \pi_k \varphi_{(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}(\boldsymbol{x}_i)\right\} + \lambda \sum_{j=1}^{K} \pi_j^{1/\gamma} \quad \gamma > 1, \tag{2.35}$$

such that:

$$\forall j \in [K], \pi_j \geq 0 \quad \text{and} \quad \sum_{j}^{K} \pi_j = 1. \tag{2.36}$$

One may wonder why choosing such a penalization. The aim is to promote the merging of similar clusters. For instance, let us consider 2 similar clusters (similar means and covariance matrices) with a weight vector $\boldsymbol{\pi} = (1/2, 1/2)$ and the merged cluster with weight $\boldsymbol{\pi}' = (1, 0)$. It is obvious that the log-likelihoods are the same. By choosing the L_1 norm, the penalty $\|\boldsymbol{\pi}\|_1$ is equal to 1 both for $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$. However, by taking the penalty $\pi_1^{1/2} + \pi_2^{1/2}$, we have a penalty for the clustering with two clusters equal to $2/\sqrt{2} \sim 1.4$. Hence, the method will favor the solution with one cluster. Let us consider the K-dimensional probability simplex $\mathbb{B}_K^+ = \{\boldsymbol{\pi} \in \mathbb{R}^K : \forall j \in [K], \pi_j \geq 0, \sum_j \pi_j = 1\}$. Our method is similar to EM in the expectation step and in the maximization step for estimating $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, it differs on the estimation of the weights vector $\boldsymbol{\pi}$ by solving the

following minimization problem:

$$\widehat{\boldsymbol{\pi}} = \underset{\boldsymbol{\pi}\in\mathbb{B}_K^+}{\arg\min} \left\{ -\frac{1}{N}\sum_{i=1}^{N} \log\Big(\sum_{j=1}^{K} \pi_j \varphi_{(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}(\boldsymbol{x}_i)\Big) + \lambda \sum_{j=1}^{K} \pi_j^{1/\gamma} \right\} \quad \gamma > 1. \tag{2.37}$$

Unfortunately, $\sum_j \pi_j^{1/\gamma}$ is neither convex nor smooth. We overcome this problem by making a change of variable. Let us note $\alpha_j = \pi_j^{1/\gamma}$ and consider the problem:

$$\widehat{\boldsymbol{\alpha}} \in \underset{\boldsymbol{\alpha}}{\arg\min} \left\{ -\frac{1}{N}\sum_{i=1}^{N} \log\Big\{\sum_{j=1}^{K} \alpha_j^{\gamma} \varphi_{(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}(\boldsymbol{x}_i)\Big\} + \lambda \sum_{j=1}^{K} \alpha_j \right\} \quad \gamma > 1, \tag{2.38}$$

subject to $\forall j \in [K]\, \alpha_j \geq 0$ and $\sum_j \alpha_j^{\gamma} = 1$. This is a smooth problem and we can recover an estimate $\widehat{\boldsymbol{\pi}} = (\widehat{\alpha}_1^{\gamma}, \ldots, \widehat{\alpha}_K^{\gamma})$. The objective function is differentiable with respect to $\boldsymbol{\alpha}$, we note it $f_{\boldsymbol{\theta}}(\boldsymbol{\alpha})$. As we saw in the previous section, we can use an iterative proximal method to solve this problem. Let us consider $\gamma > 1$ and define $A_\gamma = \{\boldsymbol{\alpha} \in \mathbb{R}^K : \forall j \in [K]\, \alpha_j \geq 0 \text{ and } \sum_j \alpha_j^{\gamma} = 1\}$. If we consider χ_{A_γ}, the indicator function of A_γ (0 in A_γ, ∞ elsewhere), the minimization problem can be rewritten as

$$\widehat{\boldsymbol{\alpha}} \in \underset{\boldsymbol{\alpha}\in\mathbb{R}^K}{\arg\min}\{f_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + \chi_{A_\gamma}(\boldsymbol{\alpha})\}, \tag{2.39}$$

and the $(t+1)^{th}$ step of the iterative proximal procedure is:

$$\begin{aligned}
\widehat{\boldsymbol{\alpha}}^{t+1} &= \text{prox}_{\chi_{A_\gamma}}(\boldsymbol{\alpha}^t - h\nabla f_{\boldsymbol{\theta}}(\boldsymbol{\alpha}^t)) && (2.40)\\
&= \underset{x\in\mathbb{R}^K}{\arg\min} \left\{ \chi_{A_\gamma}(x) + \frac{1}{2}\|x - (\boldsymbol{\alpha}^t - h\nabla f_{\boldsymbol{\theta}}(\boldsymbol{\alpha}^t))\|^2 \right\} && (2.41)\\
&= \text{P}_{A_\gamma}(\boldsymbol{\alpha}^t - h\nabla f_{\boldsymbol{\theta}}(\boldsymbol{\alpha}^t)), && (2.42)
\end{aligned}$$

with $h > 0$ a gradient step. For $j \in [K]$, the gradient of $f_{\boldsymbol{\theta}}$ w.r.t $\boldsymbol{\alpha}$ is

$$\Big[\nabla f_{\boldsymbol{\theta}}(\boldsymbol{\alpha})\Big]_j = -\frac{1}{N}\sum_{i=1}^{N} \frac{\gamma \alpha_j^{\gamma-1} \varphi_{\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j}(\boldsymbol{x}_i)}{\sum_{k=1}^{K} \alpha_k^{\gamma} \varphi_{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k}(\boldsymbol{x}_i)} + \lambda. \tag{2.43}$$

The FISTA procedure for estimating $\widehat{\boldsymbol{\alpha}}$ is given in Figure 2.9. Note that we relaxed the constraints by estimating the $K-1$ components of $\boldsymbol{\alpha}$ since $\alpha_K^{\gamma} = 1 - \sum_{k=1}^{K-1} \alpha_k^{\gamma}$. The set A_γ is redefined accordingly: $A_\gamma = \{\boldsymbol{\alpha} \in \mathbb{R}^K : \sum_{k=1}^{K-1} \alpha_k^{\gamma} \leq 1,\ \alpha_K^{\gamma} = 1 - \sum_{k=1}^{K-1} \alpha_k^{\gamma}\}$. The final EM-like procedure for estimating the parameters of a Gaussian mixture with penalization of the weight vector is given in Figure 2.10.

We generated different mixtures with K varying from 2 to 30 components in dimension 5 and draw $N = 1000$ observations. We compared our algorithm with EM and the BIC

Input: Parameter $\boldsymbol{\theta} = \{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\pi})_{k\in[K]}\}$.
Output: Parameter estimate $\widehat{\boldsymbol{\alpha}} = \big(\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{K-1}, (1 - \sum_{j=1}^{K-1} \boldsymbol{\alpha}_j^{\gamma})^{1/\gamma}\big)$.
1: Initialize $t_0 = 1$ and $\xi^0 = (\boldsymbol{\pi}_1^{1/\gamma}, ..., \boldsymbol{\pi}_{K-1}^{1/\gamma})$.
for $k \geq 1$, until convergence occurs, **do**
(a) $\boldsymbol{\alpha}^k = \mathrm{P}_A(\xi^k - h\nabla f_{\boldsymbol{\theta}}(\xi^k))$,
(b) $t_{k+1} = \frac{1+\sqrt{1+4t_{k+1}^2}}{2}$
(c) $\boldsymbol{\xi}^{k+1} = \boldsymbol{\alpha}^t + \left(\frac{t_k - 1}{t_{k+1}}\right)(\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k-1})$
end for

Figure 2.9: Estimation of $\boldsymbol{\alpha}$ via the FISTA method.

Input: Observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^p$ and a number of clusters K_{max}.
Output: parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k, \widehat{\pi}_k\}_{k\in[K]}$
1: Initialize $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
for $t = 1, \ldots,$ until convergence occurs, **do**
2: Update the parameter $\boldsymbol{\mathcal{T}}$:

$$\tau_{i,k}^t = \frac{\pi_k^t \varphi_{\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t}(\boldsymbol{x}_i)}{\sum_{k'\in[K]} \pi_{k'}^t \varphi_{\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Sigma}_{k'}^t}(\boldsymbol{x}_i)}.$$

3: Update the parameter $\widehat{\boldsymbol{\alpha}}^{t+1}$ with algorithm in Figure 2.9 and compute $\widehat{\boldsymbol{\pi}}^{t+1} = ((\alpha_1^{t+1})^\gamma), \ldots, (\alpha_K^{t+1})^\gamma)$.
4: Update parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k \in [K_{max}]$:

$$\boldsymbol{\mu}_k^{t+1} = \frac{1}{n\pi_k^{t+1}} \sum_{i=1}^{n} \tau_{i,k}^t \boldsymbol{x}_i,$$

$$\boldsymbol{\Sigma}_k^{t+1} = \frac{1}{n\pi_k^{t+1}} \sum_{i=1}^{n} \tau_{i,k}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1})^\top.$$

end for

Figure 2.10: Algorithm for estimating sparse weights vector on GMM.

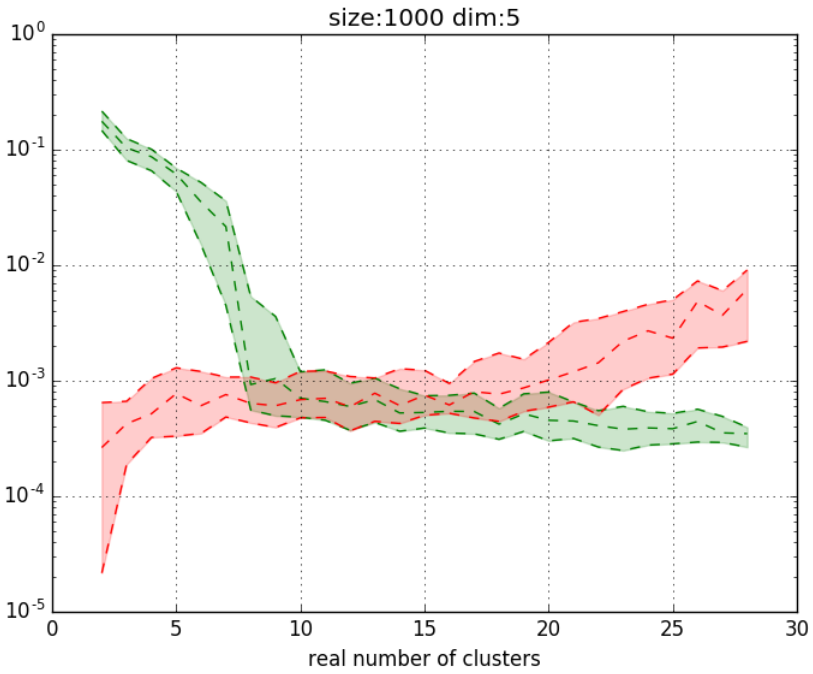

Figure 2.11: Estimation error $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$, for our algorithm (green) and EM-BIC (red). Vertical axis: error $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$ in log scale, horizontal axis: real number of clusters. First and third quartiles are shown as long as the median.

selection method with $K_{max} = 2 * K$. The resulting weight vectors are compared with the true weight $\boldsymbol{\pi}^*$ in L_1 norm. The plot of our simulations is given in Figure 2.11. The horizontal axis corresponds to the number of real clusters in the mixture and in the vertical axis corresponds to the error $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$. We ran 50 simulations, the first and third quartiles are shown as long as the median error. As we can see, our algorithm shows promising results. With a small number of clusters, the estimation error $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$ of our algorithm (green) is larger than with EM-BIC (red). However, when the number of clusters K increases, the estimation error of our algorithm decreases, whereas the estimation error of EM-BIC increases. Such phenomenon of decreasing error while the complexity increases is not natural. We believe that is caused by the choice of the parameter λ, a more clever choice of which would improve the error in the regime of small number of clusters.

2.3 Clustering and density estimation

We have tried several approaches to perform clustering using the GMM and various notions of sparsity. Unfortunately, the results were never as good as expected. One of the reasons is that all these approaches are too complex to be easily understood. In particular, they involve a number of parameters to be tuned, which turned out to be a difficult task. This led us to consider a slightly simpler task investigated in next chapters. It corresponds to choosing the weights of the components in a mixture assuming that the densities of the components (which can be interpreted as clusters) are known in advance. In practice, these densities can be furnished by some other algorithm or by an expert. This approach can be seen as an ensemble method applied to unsupervised learning.

Chapter 3

Optimal KL-Aggregation in Density Estimation

Contents

We study the maximum likelihood estimator of density of n independent observations, under the assumption that it is well approximated by a mixture with a large number of components. The main focus is on statistical properties with respect to the Kullback-Leibler loss. We establish risk bounds taking the form of sharp oracle inequalities both in deviation and in expectation. A simple consequence of these bounds is that the maximum likelihood estimator attains the optimal rate $((\log K)/n)^{1/2}$, up to a possible logarithmic correction, in the problem of convex aggregation when the number K of components is larger than $n^{1/2}$. More importantly, under the additional assumption that the Gram matrix of the components satisfies the compatibility condition, the obtained oracle inequalities yield the optimal rate in the sparsity scenario. That is, if the weight vector is (nearly) D-sparse, we get the rate $(D \log K)/n$. As a natural complement to our oracle inequalities, we introduce the notion of nearly-D-sparse aggregation and establish matching lower bounds for this type of aggregation.

3.1 Introduction

Assume that we observe n independent random vectors $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \in \mathcal{X}$ drawn from a probability distribution P^* that admits a density function f^* with respect to some reference measure ν. The goal is to estimate the unknown density by a mixture density. More precisely, we assume that for a given family of mixture components $f_1, \dots, f_K$, the unknown density of the observations f^* is well approximated by a convex combination $f_{\boldsymbol{\pi}}$ of these components, where

$$f_{\boldsymbol{\pi}}(\boldsymbol{x}) = \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}), \quad \boldsymbol{\pi} \in \mathbb{B}_+^K = \Big\{ \boldsymbol{\pi} \in [0,1]^K : \sum_{j=1}^{K} \pi_j = 1 \Big\}. \tag{3.1}$$

The assumption that the component densities $\mathcal{F} = \{f_j : j \in [K]\}$ are known essentially means that they are chosen from a dictionary obtained on the basis of previous experiments or expert knowledge.

We focus on the problem of estimation of the density function $f_{\boldsymbol{\pi}}$ and the weight vector $\boldsymbol{\pi}$ from the simplex $\mathbb{B}_+^K$ under the sparsity scenario: the ambient dimension K can be large, possibly larger than the sample size n, but most entries of $\boldsymbol{\pi}$ are either equal to zero or very small.

Our goal is to investigate the statistical properties of the Maximum Likelihood Estimator (MLE), defined by

$$\widehat{\boldsymbol{\pi}} \in \operatorname*{arg\,min}_{\boldsymbol{\pi} \in \Pi} \Big\{ -\frac{1}{n} \sum_{i=1}^{n} \log f_{\boldsymbol{\pi}}(\boldsymbol{X}_i) \Big\}, \tag{3.2}$$

where the minimum is computed over a suitably chosen subset Π of $\mathbb{B}_+^K$. In the present work, we will consider sets $\Pi = \Pi_n(\mu)$, depending on a parameter $\mu > 0$ and the sample $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_n\}$, defined by

$$\Pi_n(\mu) = \Big\{ \boldsymbol{\pi} \in \mathbb{B}_+^K : \min_{i \in [n]} \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{X}_i) \geq \mu \Big\}. \tag{3.3}$$

Note that the objective function in (3.2) is convex and the same is true for set (3.3). Therefore, the MLE $\widehat{\boldsymbol{\pi}}$ can be efficiently computed even for large K by solving a problem of convex programming. To ease notation, very often, we will omit the dependence of $\Pi_n(\mu)$ on μ and write Π_n instead of $\Pi_n(\mu)$.

The quality of an estimator $\widehat{\boldsymbol{\pi}}$ can be measured in various ways. For instance, one can consider the Kullback-Leibler divergence

$$\mathrm{KL}(f^* || f_{\widehat{\boldsymbol{\pi}}}) = \begin{cases} \int_{\mathcal{X}} f^*(\boldsymbol{x}) \log \frac{f^*(\boldsymbol{x})}{f_{\widehat{\boldsymbol{\pi}}}(\boldsymbol{x})} \, \nu(d\boldsymbol{x}), & \text{if } P^*\big(f^*(\boldsymbol{X})/f_{\widehat{\boldsymbol{\pi}}}(\boldsymbol{X}) = 0\big) = 0, \\ +\infty, & \text{otherwise,} \end{cases} \tag{3.4}$$

which has the advantage of bypassing identifiability issues. One can also consider the (well-specified) setting where $f^* = f_{\boldsymbol{\pi}^*}$ for some $\boldsymbol{\pi}^* \in \mathbb{B}_+^K$ and measure the quality of estimation through a distance between the vectors $\widehat{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}^*$ (such as the ℓ_1-norm $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$ or the Euclidean norm $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_2$).

The main contributions of the present work are the following:

(a) We demonstrate that in the mixture model there is no need to introduce sparsity favoring penalty in order to get optimal rates of estimation under the Kullback-Leibler loss in the sparsity scenario. In fact, the constraint that the weight vector belongs to the simplex acts as a sparsity inducing penalty. As a consequence, there is no need to tune a parameter accounting for the magnitude of the penalty.

(b) We show that the maximum likelihood estimator of the mixture density simultaneously attains the optimal rate of aggregation for the Kullback-Leibler loss for at least three types of aggregation: model-selection, convex and D-sparse aggregation.

(c) We introduce a new type of aggregation, termed *nearly D-sparse aggregation* that extends and unifies the notions of convex and D-sparse aggregation. We establish strong lower bounds for the nearly D-sparse aggregation and demonstrate that the maximum likelihood estimator attains this lower bound up to logarithmic factors.

3.1.1 Related work

The results developed in the present work aim to gain a better understanding (a) of the statistical properties of the maximum likelihood estimator over a high-dimensional simplex and (b) of the problem of aggregation of density estimators under the Kullback-Leibler loss. Various procedures of aggregation[1] for density estimation have been studied in the literature with respect to different loss functions. [Catoni, 1997, Yang, 2000, Juditsky et al., 2008] investigated different variants of the progressive mixture rules, also known as mirror averaging [Yuditskiĭ et al., 2005, Dalalyan and Tsybakov, 2012], with respect to the Kullback-Leibler loss and established model selection type oracle inequalities[2] in expectation. Same type of guarantees, but holding with high probability, were recently obtained in [Bellec, 2014, Butucea et al., 2016] for the procedure termed Q-aggregation, introduced in other contexts by [Dai et al., 2012, Rigollet, 2012].

[1] We refer the interested reader to [Tsybakov, 2014] for an up to date introduction into aggregation of statistical procedures.

[2] This means that they prove that the expected loss of the aggregate is almost as small as the loss of the best element of the dictionary $\{f_1, \dots, f_K\}$.

Aggregation of estimators of a probability density function under the L_2-loss was considered in [Rigollet and Tsybakov, 2007], where it was shown that a suitably chosen unbiased risk estimate minimizer is optimal both for convex and linear aggregation. The goal in the present work is to go beyond the settings of the aforementioned papers in that we want simultaneously to do as well as the best element of the dictionary, the best convex combination of the dictionary elements but also the best sparse convex combination. Note that the latter task was coined D-aggregation in [Lounici, 2007] (see also [Bunea et al., 2007]). In the present work, we rename it in D-sparse aggregation, in order to make explicit its relation to sparsity.

Key differences between the latter work and ours are that we do not assume the sparsity index to be known and we are analyzing an aggregation strategy that is computationally tractable even for large K. This is also the case of [Bunea et al., 2010, Bertin et al., 2011], which are perhaps the most relevant references to the present work. These papers deal with the L_2-loss and investigate the lasso and the Dantzig estimators, respectively, suitably adapted to the problem of density estimation. Their methods handle dictionary elements $\{f_j\}$ which are not necessarily probability density functions, but has the drawback of requiring the choice of a tuning parameter. This choice is a nontrivial problem in practice. Instead, we show here that the optimal rates of sparse aggregation with respect to the Kullback-Leibler loss can be attained by procedure which is tuning parameter free.

Risk bounds for the maximum likelihood and other related estimators in the mixture model have a long history [Li and Barron, 1999, Li, 1999, Rakhlin et al., 2005]. For the sake of comparison we recall here two elegant results providing non-asymptotic guarantees for the Kullback-Leibler loss.

Theorem 3.1.1 (Theorem 5.1 in [Li, 1999])**.** *Let $\mathcal{F}$ be a finite dictionary of cardinality K of density functions such that $\max_{f\in\mathcal{F}} \|f^*/f\|_\infty \le V$. Then, the maximum likelihood estimator over $\mathcal{F}$, $\widehat{f}^{\mathrm{ML}}_{\mathcal{F}} \in \arg\max_{f\in\mathcal{F}} \sum_{i=1}^n \log f(\boldsymbol{X}_i)$, satisfies the inequality*

$$\mathbf{E}_{f^*}\big[\mathrm{KL}\big(f^*||\widehat{f}^{\mathrm{ML}}_{\mathcal{F}}\big)\big] \le \big(2+\log V\big)\left(\min_{f\in\mathcal{F}} \mathrm{KL}(f^*||f) + \frac{2\log K}{n}\right). \tag{3.5}$$

Inequality (3.5) is an inexact oracle inequality in expectation that quantifies the ability of $\widehat{f}^{\mathrm{ML}}_{\mathcal{F}}$ to solve the problem of model-selection aggregation. The adjective inexact refers to the fact that the "bias term" $\min_{f\in\mathcal{F}} \mathrm{KL}(f^*||f)$ is multiplied by factor strictly larger than one. It is noteworthy that the remainder term $\frac{2\log K}{n}$ corresponds to the optimal rate of model-selection aggregation [Juditsky and Nemirovski, 2000, Tsybakov, 2003]. In relation with Theorem 3.1.1, it is worth mentioning a result of [Yang, 2000] and [Catoni, 1997], see also Theorem 5 in [Lecué, 2006] and Corollary 5.4 in [Juditsky et al., 2008], establishing

a risk bound similar to (3.5) without the extra factor $2 + \log V$ for the so called mirror averaging aggregate.

Theorem 3.1.2 (page 226 in [Rakhlin et al., 2005]). *Let $\mathcal{F}$ be a finite dictionary of cardinality K of density functions and let $\mathcal{C}_k = \{f_{\boldsymbol{\pi}} : \|\boldsymbol{\pi}\|_0 \leq k\}$ be the set of all the mixtures of at most k elements of $\mathcal{F}$ ($k \in [K]$). Assume that f^* and the densities f_k from $\mathcal{F}$ are bounded from below and above by some positive constants m and M, respectively. Then, there is a constant C depending only on m and M such that, for any tolerance level $\delta \in (0,1)$, the maximum likelihood estimator over $\mathcal{C}_k$, $\widehat{f}^{\mathrm{ML}}_{\mathcal{C}_k} \in \arg\max_{f\in\mathcal{C}_k} \sum_{i=1}^n \log f(\boldsymbol{X}_i)$, satisfies the inequality*

$$\mathrm{KL}\big(f^* || \widehat{f}^{\mathrm{ML}}_{\mathcal{C}_k}\big) \leq \min_{f\in\mathcal{C}_k} \mathrm{KL}(f^*||f) + C\Big(\frac{\log(K/\delta)}{n}\Big)^{1/2} \tag{3.6}$$

with probability at least $1-\delta$.

This result is remarkably elegant and can be seen as an exact oracle inequality in deviation for D-sparse aggregation (for $D = k$). Furthermore, if we choose $k = K$ in Theorem 3.1.2, then we get an exact oracle inequality for convex aggregation with a rate-optimal remainder term [Tsybakov, 2003]. However, it fails to provide the optimal rate for D-sparse aggregation.

Closing this section, we would like to mention the recent work [Xia and Koltchinskii, 2016], where oracle inequalities for estimators of low rank density matrices are obtained. They share a common feature with those obtained in this work: the adaptation to the unknown sparsity or rank is achieved without any additional penalty term. The constraint that the unknown parameter belongs to the simplex acts as a sparsity inducing penalty.

3.1.2 Additional notation

In what follows, for any $i \in [n]$, we denote by $\boldsymbol{Z}_i$ the K- dimensional vector $[f_1(\boldsymbol{X}_i), \ldots, f_K(\boldsymbol{X}_i)]^\top$ and by $\mathbf{Z}$ the $n \times K$ matrix $[\boldsymbol{Z}_1^\top, \ldots, \boldsymbol{Z}_n^\top]^\top$. We also define $\ell(u) = -\log u$, $u \in (0, +\infty)$, so that the MLE $\widehat{\boldsymbol{\pi}}$ is the minimizer of the function

$$L_n(\boldsymbol{\pi}) = \frac{1}{n}\sum_{i=1}^n \ell\big(\boldsymbol{Z}_i^\top \boldsymbol{\pi}\big). \tag{3.7}$$

For any set of indices $J \subseteq [K]$ and any $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top \in \mathbb{R}^K$, we define $\boldsymbol{\pi}_J$ as the K-dimensional vector whose j-th coordinate equals π_j if $j \in J$ and 0 otherwise. We denote the cardinality of any $J \subseteq [K]$ by $|J|$. For any set $J \subset \{1, \ldots, K\}$ and any constant $c \geq 0$,

we introduce the compatibility constants [van de Geer and Bühlmann, 2009] of a $K \times K$ positive semidefinite matrix $\mathbf{A}$,

$$\kappa_{\mathbf{A}}(J,c) = \inf\left\{\frac{c^2|J|\|\mathbf{A}^{1/2}\boldsymbol{v}\|_2^2}{(c\|\boldsymbol{v}_J\|_1 - \|\boldsymbol{v}_{J^c}\|_1)^2} : \boldsymbol{v}\in\mathbb{R}^K, \|\boldsymbol{v}_{J^c}\| < c\|\boldsymbol{v}_J\|_1\right\}, \tag{3.8}$$

$$\bar{\kappa}_{\mathbf{A}}(J,c) = \inf\left\{\frac{|J|\|\mathbf{A}^{1/2}\boldsymbol{v}\|_2^2}{\|\boldsymbol{v}_J\|_1^2} : \boldsymbol{v}\in\mathbb{R}^K, \|\boldsymbol{v}_{J^c}\|_1 < c\|\boldsymbol{v}_J\|_1\right\}. \tag{3.9}$$

The risk bounds established in the present work involve the factors $\kappa_{\mathbf{A}}(J,3)$ and $\bar{\kappa}_{\mathbf{A}}(J,1)$. One can easily check that $\bar{\kappa}_{\mathbf{A}}(J,3) \le \kappa_{\mathbf{A}}(J,3) \le \frac{9}{4}\bar{\kappa}_{\mathbf{A}}(J,1)$. We also recall that the compatibility constants of a matrix $\mathbf{A}$ are bounded from below by the smallest eigenvalue of $\mathbf{A}$.

Let us fix a function $f_0 : \mathcal{X} \to \mathbb{R}$ and denote $\bar{f}_k = f_k - f_0$ and

$$\bar{\boldsymbol{Z}}_i = [\bar{f}_1(\boldsymbol{X}_i), \dots, \bar{f}_K(\boldsymbol{X}_i)]^\top, \tag{3.10}$$

for $i \in [n]$. In the results of this work, the compatibility factors are used for the empirical and population Gram matrices of vectors $\bar{\boldsymbol{Z}}_k$, that is when $\mathbf{A} = \widehat{\boldsymbol{\Sigma}}_n$ and $\mathbf{A} = \boldsymbol{\Sigma}$ with

$$\widehat{\boldsymbol{\Sigma}}_n = \frac{1}{n}\sum_{i=1}^n \bar{\boldsymbol{Z}}_i\bar{\boldsymbol{Z}}_i^\top, \qquad \boldsymbol{\Sigma} = \mathbf{E}[\bar{\boldsymbol{Z}}_1\bar{\boldsymbol{Z}}_1^\top]. \tag{3.11}$$

The general entries of these matrices are $(\widehat{\boldsymbol{\Sigma}}_n)_{k,l} = {}^1\!/_n \sum_{i=1}^n \bar{f}_k(\boldsymbol{X}_i)\bar{f}_l(\boldsymbol{X}_i)$ and $(\boldsymbol{\Sigma})_{k,l} = \mathbf{E}[\bar{f}_k(\boldsymbol{X}_1)\bar{f}_l(\boldsymbol{X}_1)]$, respectively. We assume that there exist positive constants m and M such that for all densities f_k with $k \in [K]$, we have

$$\forall x \in \mathcal{X}, \quad m \le f_k(x) \le M. \tag{3.12}$$

We use the notation $V = M/m$. It is worth mentioning that the set of dictionaries satisfying simultaneously this boundedness assumption and the aforementioned compatibility condition is not empty. For instance, one can consider the functions $f_k(x) = 1 + {}^1\!/_2 \sin(2\pi k x)$ for $k \in [K]$. These functions are probability densities w.r.t. the Lebesgue measure on $\mathcal{X} = [0,1]$. They are bounded from below and from above by ${}^1\!/_2$ and ${}^3\!/_2$, respectively. Taking $f_0(x) = 1$, the corresponding Gram matrix is $\boldsymbol{\Sigma} = {}^1\!/_8\,\mathbf{I}_K$, which has all eigenvalues equal to ${}^1\!/_8$.

3.1.3 Agenda

The rest of the paper is organized as follows. In Section 3.2, we state our main theoretical contributions and discuss their consequences. Possible relaxations of the conditions, as

well as lower bounds showing the tightness of the established risk bounds, are considered in Section 3.3. A brief summary of the paper and some future directions of research are presented in Section 3.4. The proofs of all theoretical results are postponed to Section 3.5 and Section 3.6.

3.2 Oracle inequalities in deviation and in expectation

In this work, we prove several non-asymptotic risk bounds that imply, in particular, that the maximum likelihood estimator is optimal in model-selection aggregation, convex aggregation and D-sparse aggregation (up to log-factors). In all the results of this section we assume the parameter μ in (3.3) to be equal to 0.

Theorem 3.2.1. *Let $\mathcal{F}$ be a set of $K \geq 4$ densities satisfying the boundedness condition (3.12). Denote by $f_{\widehat{\boldsymbol{\pi}}}$ the mixture density corresponding to the maximum likelihood estimator $\widehat{\boldsymbol{\pi}}$ over Π_n defined in (3.7). There are constants $c_1 \leq 32V^3$, $c_2 \leq 288M^2V^6$ and $c_3 \leq 128M^2V^6$ such that, for any $\delta \in (0, 1/2)$, the following inequalities hold*

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \leq \inf_{\substack{J\subset[K]\\ \boldsymbol{\pi}\in\mathbb{B}_+^K}} \Big\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + c_1 \Big(\frac{\log(K/\delta)}{n}\Big)^{1/2} \|\boldsymbol{\pi}_{J^c}\|_1 + \frac{c_2|J|\log(K/\delta)}{n\kappa_{\widehat{\boldsymbol{\Sigma}}_n}(J,3)} \Big\}, \tag{3.13}$$

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \leq \inf_{J\subset[K]} \inf_{\substack{\boldsymbol{\pi}\in\mathbb{B}_+^K\\ \boldsymbol{\pi}_{J^c}=0}} \Big\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{c_3|J|\log(K/\delta)}{n\bar{\kappa}_{\widehat{\boldsymbol{\Sigma}}_n}(J,1)} \Big\} \tag{3.14}$$

with probability at least $1-\delta$.

The proof of this and the subsequent results stated in this section are postponed to Section 3.5. Comparing the two inequalities of the above theorem, one can notice two differences. First, the term proportional to $\|\boldsymbol{\pi}_{J^c}\|_1$ is absent in the second risk bound, which means that the risk of the MLE is compared to that of the best mixture with a weight sequences supported by J. Hence, this risk bound is weaker than the first one provided by (3.13). Second, the compatibility factor $\bar{\kappa}_{\widehat{\boldsymbol{\Sigma}}_n}(J,1)$ in (3.14) is larger that its counterpart $\kappa_{\widehat{\boldsymbol{\Sigma}}_n}(J,3)$ in (3.13). This entails that in the cases where the oracle is expected to be sparse, the remainder term of the bound in (3.13) is slightly looser than that of (3.14).

A first and simple consequence of Theorem 3.2.1 is obtained by taking $J = \varnothing$ in the right hand side of the first inequality. Then, $\|\boldsymbol{\pi}_{J^c}\|_1 = \|\boldsymbol{\pi}\|_1 = 1$ and we get

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \inf_{\boldsymbol{\pi}\in\mathbb{B}_+^K} \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + c_1\Big(\frac{\log(K/\delta)}{n}\Big)^{1/2}. \tag{3.15}$$

This implies that for every dictionary $\mathcal{F}$, without any assumption on the smallness of the coherence between its elements, the maximum likelihood estimator achieves the optimal rate of convex aggregation, up to a possible[3] logarithmic correction, in the high-dimensional regime $K \ge n^{1/2}$. In the case of regression with random design, an analogous result has been proved by Lecué and Mendelson [2013] and Lecué [2013]. One can also remark that the upper bound in (3.15) is of the same form as the one of Theorem 3.1.2 stated in section 3.1.1 above.

The main compelling feature of our results is that they show that the MLE adaptively achieves the optimal rate of aggregation not only in the case of convex aggregation, but also for the model-selection aggregation and D-(convex) aggregation. For handling these two cases, it is more convenient to get rid of the presence of the compatibility factor of the empirical Gram matrix $\widehat{\boldsymbol{\Sigma}}_n$. The latter can be replaced by the compatibility factor of the population Gram matrix, as stated in the next result.

Theorem 3.2.2. *Let $\mathcal{F}$ be a set of K densities satisfying the boundedness condition (3.12). Denote by $f_{\widehat{\boldsymbol{\pi}}}$ the mixture density corresponding to the maximum likelihood estimator $\widehat{\boldsymbol{\pi}}$ over Π_n defined in (3.7). There are constants $c_4 \le 32V^3 + 4$, $c_5 \le 4.5M^2(8\,V^3+1)^2$ and $c_6 \le 2M^2(8\,V^3+1)^2$ such that, for any $\delta \in (0, 1/2)$, the following inequalities hold*

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \inf_{\substack{J\subset[K]\\ \boldsymbol{\pi}\in\mathbb{B}_+^K}} \Bigg\{\mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + c_4\Big(\frac{\log(K/\delta)}{n}\Big)^{1/2}\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{c_5|J|\log(K/\delta)}{n\kappa_{\boldsymbol{\Sigma}}(J,3)}\Bigg\}, \tag{3.16}$$

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \inf_{J\subset[K]} \inf_{\substack{\boldsymbol{\pi}\in\mathbb{B}_+^K\\ \boldsymbol{\pi}_{J^c}=0}} \Big\{\mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{c_6|J|\log(K/\delta)}{n\bar{\kappa}_{\boldsymbol{\Sigma}}(J,1)}\Big\} \tag{3.17}$$

with probability at least $1-2\delta$.

The main advantage of the upper bounds provided by Theorem 3.2.2 as compared with those of Theorem 3.2.1 is that the former is deterministic, whereas the latter involves the

[3] In fact, the optimal rate of convex aggregation when $K \ge n^{1/2}$ is of order $\big(\log(K/n^{1/2})/n\big)^{1/2}$. Therefore, even the $\log K$ term is optimal whenever $K \ge Cn^{1/2+\alpha}$ for some $\alpha > 0$.

compatibility factor of the empirical Gram matrix which is random. The price to pay for getting rid of randomness in the risk bound is the increased values of the constants c_4, c_5 and c_6. Note, however, that this price is not too high, since obviously $1 \leq M \leq L$ and, therefore, $c_4 \leq 1.25c_1$, $c_5 \leq 1.56c_2$ and $c_6 \leq 1.56c_3$. In addition, the absence of randomness in the risk bound allows us to integrate it and to convert the bound in deviation into a bound in expectation.

Theorem 3.2.3 (Bound in Expectation)**.** *Let $\mathcal{F}$ be a set of K densities satisfying the boundedness condition (3.12). Denote by $f_{\widehat{\boldsymbol{\pi}}}$ the mixture density corresponding to the maximum likelihood estimator $\widehat{\boldsymbol{\pi}}$ over Π_n defined in (3.7). There are constants $c_7 \leq 20V^3 + 8$, $c_8 \leq M^2(22V^3+3)^2$ and $c_9 \leq M^2(15V^3+2)^2$ such that*

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \inf_{\substack{J\subset[K]\\ \boldsymbol{\pi}\in\mathbb{B}_+^K}} \left\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + c_7\Big(\frac{\log K}{n}\Big)^{1/2}\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{c_8|J|\log K}{n\kappa_{\boldsymbol{\Sigma}}(J,3)} \right\}, \tag{3.18}$$

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \inf_{J\subset[K]} \inf_{\substack{\boldsymbol{\pi}\in\mathbb{B}_+^K\\ \boldsymbol{\pi}_{J^c}=0}} \left\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{c_9|J|\log K}{n\bar{\kappa}_{\boldsymbol{\Sigma}}(J,1)} \right\}. \tag{3.19}$$

In inequality (3.19), upper bounding the infimum over all sets J by the infimum over the singletons, we get

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \inf_{j\in[K]} \left\{ \mathrm{KL}(f^*||f_j) + \frac{c_9\log K}{n\bar{\kappa}_{\boldsymbol{\Sigma}}(J,1)} \right\}. \tag{3.20}$$

This implies that the maximum likelihood estimator $f_{\widehat{\pi}}$ achieves the rate $\frac{\log K}{n}$ in model-selection type aggregation. This rate is known to be optimal in the model of regression [Rigollet, 2012]. If we compare this result with Theorem 3.1.1 stated in Section 3.1.1, we see that the remainder terms of these two oracle inequalities are of the same order (provided that the compatibility factor is bounded away from zero), but inequality (3.20) has the advantage of being exact.

We can also apply (3.19) to the problem of convex aggregation with small dictionary, that is for K smaller than $n^{1/2}$. Upper bounding $|J|$ by K, we get

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \inf_{\boldsymbol{\pi}\in\mathbb{B}_+^K} \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{c_9 K\log K}{n\bar{\kappa}_{\boldsymbol{\Sigma}}([K],1)}. \tag{3.21}$$

Assuming, for instance, the smallest eigenvalue of $\boldsymbol{\Sigma}$ bounded away from zero (which is a quite reasonable assumption in the context of low dimensionality), the above upper bound provides a rate of convex aggregation of the order of $\frac{K\log K}{n}$. Up to a logarithmic term, this rate is known to be optimal for convex aggregation in the model of regression.

Finally, considering all the sets J of cardinality smaller than D (with $D \leq K$) and setting $\bar{\kappa}_{\mathbf{\Sigma}}(D,1) = \inf_{J:|J|\leq D} \bar{\kappa}_{\mathbf{\Sigma}}(J,1)$, we deduce from (3.19) that

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \inf_{\boldsymbol{\pi}\in\mathbb{B}_+^K:\|\boldsymbol{\pi}\|_0\leq D} \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{c_9 D\log K}{n\bar{\kappa}_{\mathbf{\Sigma}}(D,1)}. \tag{3.22}$$

According to [Rigollet and Tsybakov, 2011, Theorem 5.3], in the regression model, the optimal rate of D-sparse aggregation is of order $(D/n)\log(K/D)$, whenever $D = o(n^{1/2})$. Inequality (3.22) shows that the maximum likelihood estimator over the simplex achieves this rate up to a logarithmic factor. Furthermore, this logarithmic inflation disappears when the sparsity D is such that, asymptotically, the ratio $\frac{\log D}{\log K}$ is bounded from above by a constant $\alpha < 1$. Indeed, in such a situation the optimal rate $\frac{D\log(K/D)}{n} = \frac{D\log K}{n}(1 - \frac{\log D}{\log K})$ is of the same order as the remainder term in (3.22), that is $\frac{D\log K}{n}$.

3.3 Discussion of the conditions and possible extensions

In this section, we start by announcing lower bounds for the Kullback-Leibler aggregation in the problem of density estimation. Then we discuss the implication of the risk bounds of the previous section to the case where the target is the weight vector $\boldsymbol{\pi}$ rather than the mixture density $f_{\boldsymbol{\pi}}$. Finally, we present some extensions to the case where the boundedness assumption is violated.

3.3.1 Lower bounds for nearly-D-sparse aggregation

As mentioned in previous section, the literature is replete with lower bounds on the minimax risk for various types of aggregation. However most of them concern the regression setting either with random or with deterministic design. Lower bounds of aggregation for density estimation were first established by Rigollet [2006] for the L_2-loss. In the case of Kullback-Leibler aggregation in density estimation, the only lower bounds we are aware are those established by Lecué [2006] for model-selection type aggregation. It is worth emphasizing here that the results of the aforementioned two papers provide weak lower bounds. Indeed, they establish the existence of a dictionary for which the minimax excess risk is lower bounded by the suitable quantity. In contrast with this, we establish here strong lower bounds that hold for every dictionary satisfying the boundedness and the compatibility conditions.

Let $\mathcal{F} = \{f_1, \dots, f_K\}$ be a dictionary of density functions on $\mathcal{X} = [0,1]$. We say that the dictionary $\mathcal{F}$ satisfies the boundedness and the compatibility assumptions if for some positive constants m, M and κ, we have $m \le f_j(x) \le M$ for all $j \in [K]$, $x \in \mathcal{X}$. In addition, we assume in this subsection that all the eigenvalues of the Gram matrix $\mathbf{\Sigma}$ belong to the interval $[\varkappa_*, \varkappa^*]$, with $\varkappa_* > 0$ and $\varkappa^* < \infty$.

For every $\gamma \in (0,1)$ and any $D \in [K]$, we define the set of nearly-D-sparse convex combinations of the dictionary elements $f_j \in \mathcal{F}$ by

$$\mathcal{H}_{\mathcal{F}}(\gamma, D) = \Big\{ f_{\boldsymbol{\pi}} : \boldsymbol{\pi} \in \mathbb{B}_+^K \text{ such that } \min_{J:|J|\le D} \|\boldsymbol{\pi}_{J^c}\|_1 \le \gamma \Big\}. \tag{3.23}$$

In simple words, $f_{\boldsymbol{\pi}}$ belongs to $\mathcal{H}_{\mathcal{F}}(\gamma, D)$ if it admits a γ-approximately D-sparse representation in the dictionary $\mathcal{F}$. We are interested in bounding from below the minimax excess risk

$$\mathcal{R}\big(\mathcal{H}_{\mathcal{F}}(\gamma, D)\big) = \inf_{\widehat{f}} \sup_{f^*} \Big\{ \mathbf{E}[\mathrm{KL}(f^* || \, \widehat{f}\,)] - \inf_{f_{\boldsymbol{\pi}} \in \mathcal{H}_{\mathcal{F}}(\gamma, D)} \mathrm{KL}(f^* || f_{\boldsymbol{\pi}}) \Big\}, \tag{3.24}$$

where the inf is over all possible estimators of f^* and the sup is over all density functions over $[0,1]$. Note that the estimator $\widehat{f}$ is not necessarily a convex combination of the dictionary elements. Furthermore, it is allowed to depend on the parameters γ and D characterizing the class $\mathcal{H}_{\mathcal{F}}(\gamma, D)$. It follows from (3.18), that if the dictionary satisfies the boundedness and the compatibility condition, then

$$\mathcal{R}\big(\mathcal{H}_{\mathcal{F}}(\gamma, D)\big) \le C \Big\{ \Big(\frac{\gamma^2 \log K}{n} \Big)^{1/2} + \frac{D \log K}{n} \Big\} \bigwedge \Big(\frac{\log K}{n} \Big)^{1/2}, \tag{3.25}$$

for some constant C depending only on m, M and $\varkappa_*$. Note that the last term accounts for the following phenomenon: If the sparsity index D is larger than a multiple of $\sqrt{n}$, then the sparsity bears no advantage as compared to the ℓ_1 constraint. The next result implies that this upper bound is optimal, at least up to logarithmic factors.

Theorem 3.3.1. *Assume that* $\log(1 + eK) \le n$. *Let* $\gamma \in (0,1)$ *and* $D \in [K]$ *be fixed. There exists a constant* A *depending only on* m, M, $\varkappa_*$ *and* $\varkappa^*$ *such that* $\mathcal{R}(\mathcal{H}_{\mathcal{F}}(\gamma, D))$ *is larger than*

$$A \Big\{ \Big[\frac{\gamma^2}{n} \log \Big(1 + \frac{K}{\gamma \sqrt{n}} \Big) \Big]^{1/2} + \frac{D \log(1 + K/D)}{n} \Big\} \bigwedge \Big[\frac{1}{n} \log \Big(1 + \frac{K}{\sqrt{n}} \Big) \Big]^{1/2}. \tag{3.26}$$

This is the first result providing lower bounds on the minimax risk of aggregation over nearly-D-sparse aggregates. To the best of our knowledge, even in the Gaussian sequence model, such a result has not been established to date. It has the advantage of unifying the results on convex and D-sparse aggregation, as well as extending them to a more general class. Let us also stress that the condition $\log(1 + eK) \le n$ is natural and unavoidable, since it ensures that the right hand side of (3.25) is smaller than the trivial bound $\log V$.

3.3.2 Weight vector estimation

The risk bounds carried out in the previous section for the problem of density estimation in the Kullback-Leibler loss imply risk bounds for the problem of weight vector estimation. Indeed, under the boundedness assumption (3.12), the Kullback-Leibler divergence between two mixture densities can be shown to be equivalent to the squared Mahalanobis distance between the weight vectors of these mixtures with respect to the Gram matrix. In order to go from the Mahalanobis distance to the Euclidean one, we make use of the restricted eigenvalue

$$\kappa^{\mathrm{RE}}_{\boldsymbol{\Sigma}}(s,c) = \inf_{\boldsymbol{v}\in\Delta(s,c)} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{v}\|_2^2, \tag{3.27}$$

with $\Delta(s,c) := \{\boldsymbol{v} : \exists\, J \subset [K] \text{ s.t. } |J| \le s,\ \|\boldsymbol{v}_{J^c}\|_1 \le c\|\boldsymbol{v}_J\|_1 \text{ and } \|\boldsymbol{v}_J\|_2 = 1\}$. This strategy leads to the next result.

Proposition 2. *Let $\mathcal{F}$ be a set of $K \ge 4$ densities satisfying condition (3.12). Denote by $f_{\widehat{\boldsymbol{\pi}}}$ the mixture density corresponding to the maximum likelihood estimator $\widehat{\boldsymbol{\pi}}$ over Π_n defined in (3.7). Let $\boldsymbol{\pi}^*$ the weight-vector of the best mixture density: $\boldsymbol{\pi}^* \in \arg\min_{\boldsymbol{\pi}} \mathrm{KL}(f^*||f_{\boldsymbol{\pi}})$, and let J^* be the support of $\boldsymbol{\pi}^*$. There are constants $c_{10} \le M^2(64V^3+8)$ and $c_{11} \le 4M^2(8V^3+1)$ such that, for any $\delta \in (0, 1/2)$, the following inequalities hold*

$$\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1 \le \frac{c_{10}|J^*|}{\bar{\kappa}_{\boldsymbol{\Sigma}}(J^*,1)} \left(\frac{\log(K/\delta)}{n}\right)^{1/2}, \tag{3.28}$$

$$\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_2 \le \frac{c_{11}}{\kappa^{\mathrm{RE}}_{\boldsymbol{\Sigma}}(|J^*|,1)} \left(\frac{2|J^*|\log(K/\delta)}{n}\right)^{1/2}, \tag{3.29}$$

$$\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_2^2 \le \frac{c_{11}}{\kappa^{\mathrm{RE}}_{\boldsymbol{\Sigma}}(|J^*|,1)} \left(\frac{2\log(K/\delta)}{n}\right)^{1/2} \tag{3.30}$$

with probability at least $1-2\delta$.

In simple words, this result tells us that the weight estimator $\widehat{\boldsymbol{\pi}}$ attains the minimax rate of estimation $|J^*|(\frac{\log(K)}{n})^{1/2}$ over the intersection of the ℓ_1 and ℓ_0 balls, when the error is measured by the ℓ_1-norm, provided that the compatibility factor of the dictionary $\mathcal{F}$ is bounded away from zero. The optimality of this rate—up to logarithmic factors—follows from the fact that the error of estimation of each nonzero coefficients of $\boldsymbol{\pi}^*$ is at least $cn^{-1/2}$ (for some $c > 0$), leading to a sum of the absolute values of the errors at least of the order $|J^*|n^{-1/2}$. The logarithmic inflation of the rate is the price to pay for not knowing the support J^*. It is clear that this reasoning is valid only when the sparsity $|J^*|$ is of smaller order than $n^{1/2}$. Indeed, in the case $|J^*| \ge cn^{1/2}$, the trivial bound $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1 \le 2$ is tighter than the one in (3.28).

Concerning the risk measured by the Euclidean norm, we underline that there are two regimes characterized by the order between upper bounds in (3.29) and (3.30). Roughly speaking, when the signal is highly sparse in the sense that $|J^*|$ is smaller than $(n/\log K)^{1/2}$, then the smallest bound is given by (3.29) and is of the order $\frac{|J^*|\log(K)}{n}$. This rate is can be compared to the rate $\frac{|J^*|\log(K/|J^*|)}{n}$, known to be optimal in the Gaussian sequence model. In the second regime corresponding to mild sparsity, $|J^*| > (n/\log K)^{1/2}$, the smallest bound is the one in (3.30). The latter is of order $(\frac{\log(K)}{n})^{1/2}$, which is known to be optimal in the Gaussian sequence model. For various results providing lower bounds in regression framework we refer the interested reader to [Raskutti et al., 2011, Rigollet and Tsybakov, 2011, Wang et al., 2014].

3.3.3 Extensions to the case of vanishing components

In the previous sections we have deliberately avoided any discussion of the role of the parameter μ, present in the search space $\Pi_n(\mu)$ of the problem (3.2)-(3.3). In fact, when all the dictionary elements are separated from zero by a constant m, a condition assumed throughout previous sections, choosing any value of $\mu \leq m$ is equivalent to choosing $\mu = 0$. Therefore, the choice of this parameter does not impact the quality of estimation. However, this parameter might have strong influence in practice both on statistical and computational complexity of the maximum likelihood estimator. A first step in understanding the influence of μ on the statistical complexity is made in the next paragraphs.

Let us consider the case where the condition $\min_x \min_j f_j(x) \geq m > 0$ fails, but the upper-boundedness condition $\max_x \max_j f_j(x) \leq M$ holds true. In such a situation, we replace the definition $V = M/m$ by $V = M/\mu$. We also define the set $\Pi^*(\mu) = \big\{\boldsymbol{\pi} \in \mathbb{B}_+^K : P^*\big(f_{\boldsymbol{\pi}}(\boldsymbol{X}) \geq \mu\big) = 1\big\}$. In order to keep mathematical formulae simple, we will only state the equivalent of (3.14) in the case of $m = 0$. All the other results of the previous section can be extended in a similar way.

Proposition 3. *Let $\mathcal{F}$ be a set of $K \geq 2$ densities satisfying the boundedness condition $\sup_{\boldsymbol{x}\in\mathcal{X}} f_j(\boldsymbol{x}) \leq M$. Denote by $f_{\widehat{\boldsymbol{\pi}}}$ the mixture density corresponding to the maximum likelihood estimator $\widehat{\boldsymbol{\pi}}$ over $\Pi_n(\mu)$ defined in (3.7). There is a constant $\bar{c} \leq 128M^2V^4$ such that, for any $\delta \in (0, 1/2)$,*

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \leq \inf_{J\subset[K]} \inf_{\substack{\boldsymbol{\pi}\in\Pi^*(\mu)\\ \boldsymbol{\pi}_{J^c}=0}} \left\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{\bar{c}|J|\log(K/\delta)}{n\bar{\kappa}_{\widehat{\boldsymbol{\Sigma}}_n}(J,1)} \right\} + \int_{\mathcal{X}} (\log\mu - \log f_{\widehat{\boldsymbol{\pi}}})_+ f^* d\nu \tag{3.31}$$

on an event of probability at least $1-\delta$. *Furthermore, if* $\inf_{\boldsymbol{x}\in\mathcal{X}} f^*(\boldsymbol{x}) \geq \mu$, *then, on the same event, we have*

$$\|f^* - f_{\widehat{\boldsymbol{\pi}}}\|^2_{L^2(P^*)} \leq 2M^2 \inf_{J\subset[K]} \inf_{\substack{\boldsymbol{\pi}\in\Pi^*(\mu)\\ \boldsymbol{\pi}_{J^c}=0}} \left\{ \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{\bar{c}|J|\log(K/\delta)}{n\bar{\kappa}_{\widehat{\boldsymbol{\Sigma}}_n}(J,1)} \right\}. \tag{3.32}$$

The last term present in the first upper bound, $\int_{\mathcal{X}}(\log\mu - \log f_{\widehat{\boldsymbol{\pi}}})_+ f^* d\nu$ is the price we pay for considering densities that are not lower bounded by a given constant. A simple, non-random upper bound on this term is $\int_{\mathcal{X}} \max_{k\in[K]}(\log\mu - \log f_k)_+ f^* d\nu$. Providing a tight upper bound on this kind or remainder terms is an important problem which lies beyond the scope of the present work.

3.4 Conclusion

In this paper, we have established exact oracle inequalities for the maximum likelihood estimator of a mixture density. This oracle inequality clearly highlights the interplay of three sources of error: misspecification of the model of mixture, departure from D-sparsity and stochastic error of estimating D nonzero coefficients. We have also proved a lower bound that show that the remainder terms of our upper bounds are optimal, up to logarithmic terms. This lower bound is valid not only for the maximum likelihood estimator, but for any estimator of the density function. As a consequence, the maximum likelihood estimator has a nearly optimal excess risk in the minimax sense.

In all the results of the present paper, we have assumed that the components of the mixture model are deterministic. From a practical point of view, it might be reasonable to choose these components in a data driven way, using, for instance, a hold-out sample. This question, as well as the problem of tuning the parameter μ , constitute interesting and challenging avenues for future research.

3.5 Proofs of results stated in previous sections

This section collects the proofs of the theorems and claims stated in previous sections.

3.5.1 Proof of Theorem 3.2.1

The main technical ingredients of the proof are a strong convexity argument and a control of the maximum of an empirical process. The corresponding results are stated in Lemma 3.5.2

and Proposition 3.5.1, respectively, deferred to Section 3.5.6. We denote by $\bar{\mathbf{Z}}$ the $n \times K$ matrix $[\bar{\boldsymbol{Z}}_1, \ldots, \bar{\boldsymbol{Z}}_K]$.

Since $\widehat{\boldsymbol{\pi}}$ is a minimizer of $L_n(\cdot)$, see (3.2) and (3.7), we know that $L_n(\widehat{\boldsymbol{\pi}}) \leq L_n(\boldsymbol{\pi})$ for every π. However, this inequality can be made sharper using the (local) strong convexity of the function $\ell(u) = -\log(u)$. Indeed, Lemma 3.5.2 below shows that

$$\frac{1}{n}\sum_{i=1}^{n} \ell(f_{\widehat{\boldsymbol{\pi}}}(\boldsymbol{X}_i)) \leq \frac{1}{n}\sum_{i=1}^{n} \ell(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)) - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2. \tag{3.33}$$

On the other hand, if we set $\varphi(\pi, \boldsymbol{x}) = \int (\log f_{\boldsymbol{\pi}}) f^* d\nu - \log f_{\boldsymbol{\pi}}(\boldsymbol{x})$, we have $\mathbf{E}_{f^*}[\varphi(\boldsymbol{\pi}, \boldsymbol{X}_i)] = 0$ and

$$\ell(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)) = \mathrm{KL}(f^* || f_{\boldsymbol{\pi}}) - \int_{\mathcal{X}} f^* \log f^* d\nu + \varphi(\pi, \boldsymbol{X}_i). \tag{3.34}$$

Combining inequalities (3.33) and (3.34), we get

$$\mathrm{KL}(f^* || f_{\widehat{\boldsymbol{\pi}}}) \leq \mathrm{KL}(f^* || f_{\boldsymbol{\pi}}) - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} \big(\varphi(\boldsymbol{\pi}, \boldsymbol{X}_i) - \varphi(\widehat{\boldsymbol{\pi}}, \boldsymbol{X}_i)\big). \tag{3.35}$$

The next step of the proof consists in establishing a suitable upper bound on the noise term $\Phi_n(\boldsymbol{\pi}) - \Phi_n(\widehat{\boldsymbol{\pi}})$ where

$$\Phi_n(\boldsymbol{\pi}) = \frac{1}{n}\sum_{i=1}^{n} \varphi(\boldsymbol{\pi}, \boldsymbol{X}_i). \tag{3.36}$$

According to the mean value theorem, setting $\zeta_n := \sup_{\bar{\boldsymbol{\pi}} \in \boldsymbol{\Pi}_n} \big\|\nabla \Phi_n(\bar{\boldsymbol{\pi}})\big\|_\infty$, for every vector $\boldsymbol{\pi} \in \boldsymbol{\Pi}_n$, it holds that

$$|\Phi_n(\widehat{\boldsymbol{\pi}}) - \Phi_n(\boldsymbol{\pi})| \leq \sup_{\bar{\boldsymbol{\pi}} \in \boldsymbol{\Pi}_n} \big\|\nabla \Phi_n(\bar{\boldsymbol{\pi}})\big\|_\infty \|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1 = \zeta_n \|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1. \tag{3.37}$$

This inequality, combined with (3.35), yields

$$\mathrm{KL}(f^* || f_{\widehat{\boldsymbol{\pi}}}) \leq \mathrm{KL}(f^* || f_{\boldsymbol{\pi}}) - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 + \zeta_n \|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1. \tag{3.38}$$

Using the Gram matrix $\widehat{\boldsymbol{\Sigma}}_n = {}^1\!/_n \bar{\mathbf{Z}}^\top \bar{\mathbf{Z}}$, the quantity $\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2$ can be rewritten as

$$\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 = n\|\widehat{\boldsymbol{\Sigma}}_n^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2. \tag{3.39}$$

We proceed with applying the following result [Bellec et al., 2016, Lemma 2].

Lemma 3.5.1 (Bellec et al. [2016], Lemma 2)**.** *For any pair of vectors $\boldsymbol{\pi}, \boldsymbol{\pi}' \in \mathbb{R}^K$, for any pair of scalars $\mu > 0$ and $\gamma > 1$, for any $K \times K$ symmetric matrix $\mathbf{A}$ and for any set $J \subset [p]$, the following inequality is true*

$$2\mu\gamma^{-1}(\|\boldsymbol{\pi} - \boldsymbol{\pi}'\|_1 + \gamma\|\boldsymbol{\pi}\|_1 - \gamma\|\boldsymbol{\pi}'\|_1) - \|\mathbf{A}(\boldsymbol{\pi} - \boldsymbol{\pi}')\|_2^2 \leq 4\mu\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{(\gamma+1)^2\mu^2|J|}{\gamma^2 \kappa_{\mathbf{A}^2}(J, c_\gamma)}, \tag{3.40}$$

where $c_\gamma = (\gamma+1)/(\gamma-1)$.

Choosing $\mathbf{A} = \widehat{\mathbf{\Sigma}}_n^{1/2}/(\sqrt{2}\,M)$, $\mu = \zeta_n$ and $\gamma = 2$ (thus $c_\gamma = 3$) we get the inequality

$$\zeta_n\|\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}\|_1 - \|\mathbf{A}(\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}})\|_2^2 \le 4\zeta_n\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{9\zeta_n^2|J|}{4\kappa_{\mathbf{A}^2}(J,3)}, \qquad \forall J \in \{1,\dots,p\}. \tag{3.41}$$

One can check that $\kappa_{\mathbf{A}^2}(J,3) = \kappa_{\widehat{\mathbf{\Sigma}}_n}(J,3)/(2M^2)$. Combining the last inequality with (3.38), we arrive at

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + 4\zeta_n\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{9M^2\zeta_n^2|J|}{2\kappa_{\widehat{\mathbf{\Sigma}}_n}(J,3)}. \tag{3.42}$$

Since the last inequality holds for every $\boldsymbol{\pi}$, we can insert an $\inf_{\boldsymbol{\pi}}$ in the right hand side. Furthermore, in view of Proposition 3.5.1 below, with probability larger than $1-\delta$, ζ_n is bounded from above by $8V^3(\frac{\log(K/\delta)}{n})^{1/2}$. This completes the proof of (3.13).

To prove (3.14), we follow the same steps as above up to inequality (3.38). Then, we remark that for every $\boldsymbol{\pi}$ in the simplex satisfying $\boldsymbol{\pi}_{J^c} = 0$, it holds

$$\|(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})_{J^c}\|_1 = \|\widehat{\boldsymbol{\pi}}_{J^c}\|_1 = 1 - \|\widehat{\boldsymbol{\pi}}_J\|_1 = \|\boldsymbol{\pi}_J\|_1 - \|\widehat{\boldsymbol{\pi}}_J\|_1 \le \|(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})_J\|_1. \tag{3.43}$$

Therefore, $\|\widehat{\mathbf{\Sigma}}_n^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \ge \frac{\bar\kappa_{\widehat{\mathbf{\Sigma}}_n}(J,1)\|(\boldsymbol{\pi}-\widehat{\boldsymbol{\pi}})_J\|_1^2}{|J|}$, we have with probability at least $1-\delta$

$$\begin{aligned}
\zeta_n\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1 - \frac{1}{2M^2n}\|\mathbf{Z}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 &\le 2\zeta_n\|(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})_J\|_1 - \frac{1}{2M^2}\|\widehat{\mathbf{\Sigma}}_n^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \\
&\le 2\zeta_n\|(\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}})_J\|_1 - \frac{\bar\kappa_{\widehat{\mathbf{\Sigma}}_n}(J,1)\|(\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}})_J\|_1^2}{2M^2|J|} \\
&\le \frac{2\zeta_n^2M^2|J|}{\bar\kappa_{\widehat{\mathbf{\Sigma}}_n}(J,1)}.
\end{aligned} \tag{3.44}$$

Replacing the right hand term in (3.38) and taking the infimum, we get the claim of the corollary. Since, in view of Proposition 3.5.1 below, with probability larger than $1-\delta$, ζ_n is bounded from above by $8V^3(\frac{\log(K/\delta)}{n})^{1/2}$, we get the claim of (3.14).

3.5.2 Proof of Theorem 3.2.2

Let us denote $\boldsymbol{v} = \widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}$. According to (3.38) and (3.39), we have

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \zeta_n\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1 - \frac{1}{2M^2}\|\widehat{\mathbf{\Sigma}}_n^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \tag{3.45}$$

$$\le \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \zeta_n\|\boldsymbol{v}\|_1 - \frac{1}{2M^2}\|\mathbf{\Sigma}^{1/2}\boldsymbol{v}\|_2^2 + \frac{1}{2M^2}\,\boldsymbol{v}^\top(\mathbf{\Sigma} - \widehat{\mathbf{\Sigma}}_n)\boldsymbol{v}. \tag{3.46}$$

As $\boldsymbol{v}$ is the difference of two vectors lying on the simplex, we have $\|\boldsymbol{v}\|_1 \leq 2$. Let $\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty = \max_{j,j'} |(\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n)_{j,j'}|$ stand for the largest (in absolute values) element of the matrix $\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n$. We have

$$\boldsymbol{v}^\top(\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n)\boldsymbol{v} \leq \|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty \|\boldsymbol{v}\|_1^2 \leq 2\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty \|\boldsymbol{v}\|_1. \tag{3.47}$$

Setting $\bar{\zeta}_n = \zeta_n + M^{-2}\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty$, we get

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \leq \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \bar{\zeta}_n\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1 - \frac{1}{2M^2}\|\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2. \tag{3.48}$$

Following the same steps as those used for obtaining (3.42), we arrive at

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \leq \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + 4\bar{\zeta}_n\|\boldsymbol{\pi}_{J^c}\|_1 + \frac{9\bar{\zeta}_n^2 M^2|J|}{2\kappa_{\boldsymbol{\Sigma}}(J,3)}. \tag{3.49}$$

The last step consists in evaluating the quantiles of the random variable $\bar{\zeta}_n$. To this end, one checks that the Hoeffding inequality combined with the union bound yields

$$\mathbf{P}\Big\{\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty > t\Big\} \leq K(K-1)\exp(-2nt^2/M^4), \qquad \forall t > 0. \tag{3.50}$$

In other terms, for every $\delta \in (0,1)$, we have

$$\mathbf{P}\Big\{\|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}_n\|_\infty \leq M^2\Big(\frac{\log(K^2/\delta)}{2n}\Big)^{1/2}\Big\} \geq 1 - \delta. \tag{3.51}$$

Note that for $\delta \leq 1$, we have $\log(K^2/\delta) \leq 2\log(K/\delta)$. Combining with Proposition 3.5.1, this implies that $\bar{\zeta}_n \leq (8V^3 + 1)\big(\frac{\log(K/\delta)}{n}\big)^{1/2}$ with probability larger than $1 - 2\delta$. This completes the proof of (3.16). The proof of (3.17) is omitted since it repeats the same arguments as those used for proving (3.14).

3.5.3 Proof of Theorem 3.2.3

According to (3.49), for any $\boldsymbol{\pi} \in \Pi$ and any $J \subset \{1,\dots,K\}$, we have

$$\mathbf{E}[\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}})] \leq \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + 4\|\boldsymbol{\pi}_{J^c}\|_1\mathbf{E}[\bar{\zeta}_n] + \frac{9M^2|J|}{2\kappa_{\boldsymbol{\Sigma}}(J,3)}\,\mathbf{E}[\bar{\zeta}_n^2]. \tag{3.52}$$

Recall now that $\bar{\zeta}_n = \zeta_n + M^{-2}\|\widehat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}\|_\infty$ and, according to Proposition 3.5.1, we have

$$\mathbf{E}[\zeta_n] \leq 4V^3\Big(\frac{2\log(2K^2)}{n}\Big)^{1/2} \qquad \text{and} \qquad \mathbf{Var}[\zeta_n] \leq \frac{V^2}{2n}. \tag{3.53}$$

Using Theorem 3.6.2, one easily checks that

$$\mathbf{E}[\|\widehat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}\|_\infty] \leq M^2\Big(\frac{\log(2K^2)}{2n}\Big)^{1/2}. \tag{3.54}$$

This implies that

$$\mathbf{E}[\bar{\zeta}_n] \leq (8V^3+1)\Big(\frac{\log(2K^2)}{2n}\Big)^{1/2}. \tag{3.55}$$

Similarly, in view of the Efron-Stein inequality, we have $\mathbf{Var}[\|\widehat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}\|_\infty] \leq \frac{M^4}{2n}$. This implies that

$$\begin{aligned}
\mathbf{E}[\bar{\zeta}_n^2] &\leq (\mathbf{E}[\bar{\zeta}_n])^2 + \big\{(\mathbf{Var}[\zeta_n])^{1/2} + M^{-2}(\mathbf{Var}[\|\widehat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}\|_\infty])^{1/2}\big\}^2 && (3.56)\\
&\leq (8V^3+1)^2\frac{\log(2K^2)}{2n} + \frac{(V+1)^2}{2n} && (3.57)\\
&\leq 1.615(8V^3+1)^2\frac{\log K}{n}. && (3.58)
\end{aligned}$$

Combining (3.55), (3.58) and (3.52), we get the desired result.

3.5.4 Proof of Proposition 2

Using the strong convexity of the function $u \mapsto -\log u$ over the interval $[m, M]$ and the fact that $\boldsymbol{\pi}^*$ minimizes the convex function $\boldsymbol{\pi} \mapsto \mathrm{KL}(f^*||f_{\boldsymbol{\pi}})$, we get

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \geq \mathrm{KL}(f^*||f_{\boldsymbol{\pi}^*}) + \frac{1}{2M^2}\|\widehat{\boldsymbol{\Sigma}}_n^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*)\|_2^2. \tag{3.59}$$

Combining with (3.48), in which we replace $\boldsymbol{\pi}$ by $\boldsymbol{\pi}^*$, we get

$$\|\boldsymbol{\Sigma}^{1/2}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*)\|_2^2 \leq 2M^2\bar{\zeta}_n\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1. \tag{3.60}$$

Let us set $\boldsymbol{v} = \widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*$. If $\boldsymbol{v} = 0$, then the claims are trivial. In the rest of this proof, we assume $\|\boldsymbol{v}\|_1 > 0$. In view of (3.43), we have $\|\boldsymbol{v}\|_1 \leq 2\|\boldsymbol{v}_{J^*}\|_1$. Therefore, using the definition of the compatibility factor, we get

$$\|\boldsymbol{v}\|_1^2 \leq 4\|\boldsymbol{v}_{J^*}\|_1^2 \leq \frac{4|J^*|\,\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{v}\|_2^2}{\bar{\kappa}(J^*,1)} \leq \frac{8|J^*|\,M^2\bar{\zeta}_n\|\boldsymbol{v}\|_1}{\bar{\kappa}(J^*,1)}. \tag{3.61}$$

We have already checked that $\bar{\zeta}_n \leq (8V^3+1)\big(\frac{\log(K/\delta)}{n}\big)^{1/2}$ with probability larger than $1-2\delta$. Dividing both sides of inequality (3.61) by $\|\boldsymbol{v}\|_1$ and using the aforementioned upper bound on $\bar{\zeta}_n$, we get the desired bound on $\|\boldsymbol{v}\|_1 = \|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*\|_1$.

In order to bound the error $\boldsymbol{v} = \widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^*$ in the Euclidean norm, we denote by $\widehat{J}$ the set of $D = |J^*|$ indices corresponding to D largest entries of the vector $(|v_1|, \ldots, |v_K|)$. Since

$\|\boldsymbol{v}\|_1 \le 2\|\boldsymbol{v}_{J^*}\|_1$, we clearly have $\|\boldsymbol{v}\|_1 \le 2\|\boldsymbol{v}_{\widehat{J}}\|_1$. Therefore,

$$\begin{aligned}
\|\boldsymbol{v}\|_2^2 &= \|\boldsymbol{v}_{\widehat{J}}\|_2^2 + \|\boldsymbol{v}_{\widehat{J}^c}\|_2^2 && (3.62)\\
&\le \|\boldsymbol{v}_{\widehat{J}}\|_2^2 + \|\boldsymbol{v}_{\widehat{J}^c}\|_\infty \|\boldsymbol{v}_{\widehat{J}^c}\|_1 && (3.63)\\
&\le \|\boldsymbol{v}_{\widehat{J}}\|_2^2 + \frac{\|\boldsymbol{v}_{\widehat{J}}\|_1}{D}\|\boldsymbol{v}_{\widehat{J}^c}\|_1 && (3.64)\\
&\le \|\boldsymbol{v}_{\widehat{J}}\|_2^2 + \frac{1}{D}\|\boldsymbol{v}_{\widehat{J}}\|_1^2 \le 2\|\boldsymbol{v}_{\widehat{J}}\|_2^2. && (3.65)
\end{aligned}$$

Combining this inequality with the definition of the restricted eigenvalue and inequality (3.60) above, we arrive at

$$\begin{aligned}
\|\boldsymbol{v}_{\widehat{J}}\|_2^2 &\le \frac{\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{v}\|_2^2}{\kappa^{\mathrm{RE}}(D,1)} \le \frac{2M^2\bar{\zeta}_n\|\boldsymbol{v}\|_1}{\kappa^{\mathrm{RE}}(D,1)} && (3.66)\\
&\le \frac{4M^2\bar{\zeta}_n(\|\boldsymbol{v}_{\widehat{J}}\|_1 \wedge 1)}{\kappa^{\mathrm{RE}}(D,1)} \le \frac{4M^2\,\bar{\zeta}_n(\sqrt{D}\|\boldsymbol{v}_{\widehat{J}}\|_2 \wedge 1)}{\kappa^{\mathrm{RE}}(D,1)}. && (3.67)
\end{aligned}$$

Dividing both sides by $\|\boldsymbol{v}_{\widehat{J}}\|_2$, taking the square and using (3.65), we get

$$\|\boldsymbol{v}\|_2 \le \sqrt{2}\,\|\boldsymbol{v}_{\widehat{J}}\|_2 \le \frac{4\sqrt{2}M^2|J^*|^{1/2}\,\bar{\zeta}_n}{\kappa^{\mathrm{RE}}(|J^*|,1)} \bigwedge \frac{2\sqrt{2}M\bar{\zeta}_n^{1/2}}{\kappa^{\mathrm{RE}}(|J^*|,1)^{1/2}}. \tag{3.68}$$

This inequality, in conjunction with the upper bound on $\bar{\zeta}_n$ used above, completes the proof of the second claim.

3.5.5 Proof of Proposition 3

We repeat the proof of Theorem 3.2.1 with some small modifications. First of all, we replace the function $\ell(u) = -\log(u)$ by the function

$$\bar{\ell}(u) = \begin{cases} -\log(u/\mu), & \text{if } u \ge \mu, \\ (1-\frac{u}{\mu}) + \frac{1}{2}(1-\frac{u}{\mu})^2, & \text{if } u \in (0,\mu). \end{cases} \tag{3.69}$$

One easily checks that this function is twice continuously differentiable with a second derivative satisfying $M^{-2} \le \bar{\ell}''(u) \le \mu^{-2}$ for every $u \in (0, M)$. Furthermore, since $\bar{\ell}(u) = \ell(u/\mu)$ for every $u \ge \mu$, we have $\bar{L}_n(\widehat{\boldsymbol{\pi}}) = L_n(\widehat{\boldsymbol{\pi}})$, where we have used the notation $\bar{L}_n(\boldsymbol{\pi}) = \frac{1}{n}\sum_{i=1}^n \bar{\ell}(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i))$. Therefore, similarly to (3.33), we get

$$\frac{1}{n}\sum_{i=1}^n \bar{\ell}(f_{\widehat{\boldsymbol{\pi}}}(\boldsymbol{X}_i)) \le \frac{1}{n}\sum_{i=1}^n \bar{\ell}(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)) - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2, \tag{3.70}$$

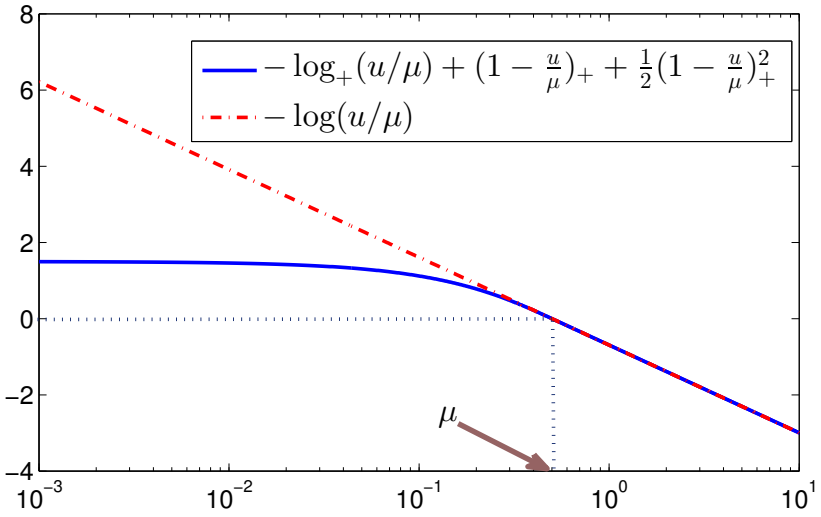

Figure 3.1: The plot of the function $u \mapsto \bar{\ell}(u)$, used in the proof of Proposition 3, superposed on the plot of the function $u \mapsto \ell(u) = -\log u$. We see that the former is a strongly convex surrogate of the latter.

for every $\boldsymbol{\pi} \in \Pi^*(\mu)$. Let us define $\bar{\varphi}(\boldsymbol{\pi}, \boldsymbol{x}) = \bar{\ell}(f_{\boldsymbol{\pi}}(\boldsymbol{x})) - \int \bar{\ell}(f_{\boldsymbol{\pi}}) f^* d\nu$ and $\bar{\Phi}_n(\boldsymbol{\pi}) = \frac{1}{n}\sum_{i=1}^n \bar{\varphi}(\boldsymbol{\pi}, \boldsymbol{X}_i)$. We have

$$\begin{aligned}
\int \bar{\ell}(f_{\widehat{\boldsymbol{\pi}}})\, f^* d\nu &\le \int \bar{\ell}(f_{\boldsymbol{\pi}})\, f^* d\nu - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \\
&\quad + \frac{1}{n}\sum_{i=1}^n \big(\varphi(\boldsymbol{\pi}, \boldsymbol{X}_i) - \varphi(\widehat{\boldsymbol{\pi}}, \boldsymbol{X}_i)\big) \qquad (3.71) \\
&\le \int \bar{\ell}(f_{\boldsymbol{\pi}})\, f^* d\nu - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \\
&\quad + \underbrace{\sup_{\boldsymbol{\pi} \in \Pi_n(0)} \|\nabla \bar{\Phi}_n(\boldsymbol{\pi})\|_\infty}_{:=\xi_n} \|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}\|_1. \qquad (3.72)
\end{aligned}$$

Notice that $\boldsymbol{\pi} \in \Pi^*(\mu)$ implies that $\bar{\ell}(f_{\boldsymbol{\pi}}) = \log\mu - \log f_{\boldsymbol{\pi}}$ and that $\bar{\ell}(f_{\widehat{\boldsymbol{\pi}}}) \ge \log\mu - \log f_{\widehat{\boldsymbol{\pi}}} - (\log\mu - \log f_{\widehat{\boldsymbol{\pi}}})_+$. Therefore, along the lines of the proof of (3.14) (see, namely, (3.44)), we get

$$\mathrm{KL}(f^*||f_{\widehat{\boldsymbol{\pi}}}) \le \mathrm{KL}(f^*||f_{\boldsymbol{\pi}}) + \frac{2\xi_n^2 M^2 |J|}{\bar{\kappa}_{\widehat{\boldsymbol{\Sigma}}_n}(J, 1)} + \int_{\mathcal{X}} (\log\mu - \log f_{\widehat{\boldsymbol{\pi}}})_+ f^* d\nu. \qquad (3.73)$$

We can repeat now the arguments of Proposition 3.5.1 with some minor modifications. First of all, we rewrite ξ_n as $\xi_n = \max_{l=1,\dots,K} \xi_{l,n}$ with $\xi_{l,n} = \sup_{\boldsymbol{\pi} \in \Pi_n(0)} |\partial_l \bar{\Phi}_n(\boldsymbol{\pi})|$. One checks

that the bounded difference inequality and the Efron-Stein inequality can be applied with an additional factor 2, since for $F_l(\mathbf{X}) = \sup_{\boldsymbol{\pi}\in\Pi_n(0)} |\partial_l \bar{\Phi}_n(\boldsymbol{\pi})|$, we have

$$|F_l(\mathbf{X}) - F_l(\mathbf{X}')| \leq \frac{2M}{n\mu} = \frac{2V}{n}. \tag{3.74}$$

Therefore, for every $l \in [K]$, with probability larger than $1-(\delta/K)$, we have $\xi_{l,n} \leq \mathbf{E}[\xi_{l,n}] + V(\frac{2\log(K/\delta)}{n})^{1/2}$ and $\mathbf{Var}[\xi_n] \leq (2V)^2/n$. By the union bound, we obtain that with probability larger than $1-\delta$, $\xi_n \leq \max_l \mathbf{E}[\xi_{l,n}] + V(\frac{2\log(K/\delta)}{n})^{1/2}$. Thus, to upper bound $\mathbf{E}[\xi_{l,n}]$, we use the symmetrization argument:

$$\mathbf{E}[\xi_{l,n}] \leq 2\mathbf{E}\left[\sup_{\boldsymbol{\pi}\in\Pi_n(0)} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i \bar{\ell}'(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)) f_l(\boldsymbol{X}_i)\right|\right] \tag{3.75}$$

$$\leq 2M\mathbf{E}\left[\sup_{\boldsymbol{\pi}\in\Pi_n(0)} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i \bar{\ell}'(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i))\right|\right] \tag{3.76}$$

$$\leq \frac{2M}{\mu}\mathbf{E}\left[\left|\frac{1}{n}\sum_{i=1}^n \epsilon_i\right|\right] + 2M\mathbf{E}\left[\sup_{\boldsymbol{\pi}\in\Pi_n(0)} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i [\bar{\ell}'(f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)) - \bar{\ell}'(0)]\right|\right], \tag{3.77}$$

where the second inequality comes from [Boucheron et al., 2013, Th. 11.5]. Note that the function $\bar{\ell}'$, the derivative of $\bar{\ell}$ defined in (3.69), is by construction Lipschitz with constant $1/\mu^2$. Therefore, in view of the contraction principle,

$$\mathbf{E}[\xi_{l,n}] \leq \frac{2M}{\mu}\mathbf{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \epsilon_i\right)^2\right]^{1/2} + \frac{4M}{\mu^2}\mathbf{E}\left[\sup_{\boldsymbol{\pi}\in\Pi_n(0)} \frac{1}{n}\sum_{i=1}^n \epsilon_i f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)\right] \tag{3.78}$$

$$\leq \frac{2M}{\mu\sqrt{n}} + \frac{4M}{\mu^2}\mathbf{E}\left[\sup_{k\in[K]} \frac{1}{n}\sum_{i=1}^n \epsilon_i f_k(\boldsymbol{X}_i)\right] \tag{3.79}$$

$$\leq \frac{2M}{\mu\sqrt{n}} + \frac{8M^2}{\mu^2}\left(\frac{\log K}{2n}\right)^{1/2} \leq \frac{2V^2(1+2\sqrt{2\log K})}{\sqrt{n}}. \tag{3.80}$$

As a consequence, we proved that with probability larger than $1-\delta$, we have $\xi_n \leq 8V^2(\frac{\log K}{n})^{1/2}$. This completes the proof of the first inequality. In order to prove the second one, we simply change the way we have evaluated the term $\int \bar{\ell}(f_{\widehat{\boldsymbol{\pi}}})f^*$ in the left hand side of (3.71). Since $\bar{\ell}$ is strongly convex with a second order derivative bounded from below by $1/M^2$, we have $\bar{\ell}(f_{\widehat{\boldsymbol{\pi}}}) \geq \bar{\ell}(f^*) + \bar{\ell}'(f^*)(f_{\widehat{\boldsymbol{\pi}}} - f^*) + \frac{1}{2M^2}(f_{\widehat{\boldsymbol{\pi}}} - f^*)^2$. Since f^* is always larger than μ, the derivative $\bar{\ell}'(f^*)$ equals $1/f^*$. Integrating over $\mathcal{X}$, we get the second inequality of the proposition.

3.5.6 Auxiliary results

We start by a general convex result based on the strong convexity of the $-log$ function to derive a bound on the estimated log-likelihood.

Lemma 3.5.2. *Let us assume that* $M = \max_{j\in[K]} \|f_j\|_\infty < \infty$. *Then, for any* $\boldsymbol{\pi} \in \mathbb{B}_+^K$, *it holds that*

$$L_n(\widehat{\boldsymbol{\pi}}) \le L_n(\boldsymbol{\pi}) - \frac{1}{2M^2 n}\|\bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2. \tag{3.81}$$

Proof. Recall that $\widehat{\boldsymbol{\pi}}$ minimizes the function L_n defined in (3.7) over Π_n. Furthermore, the function $u \mapsto \ell(u)$ is clearly strongly convex with a second order derivative bounded from below by $1/M^2$ over the set $u \in (0, M]$. Therefore, for every $\widehat{u} \in (0, M]$, the function $\widetilde{\ell}$ given by:

$$\widetilde{\ell}(u) = \ell(u) - \frac{1}{2M^2}(\widehat{u} - u)^2, \quad u \in (0, M], \tag{3.82}$$

is convex. This implies that the mapping

$$\boldsymbol{\pi} \mapsto \widetilde{L}_n(\boldsymbol{\pi}) = L_n(\boldsymbol{\pi}) - \frac{1}{2M^2 n}\|\mathbf{Z}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})\|_2^2 \tag{3.83}$$

is convex over the set $\boldsymbol{\pi} \in \mathbb{B}_+^K$. This yields[4]

$$\widetilde{L}_n(\boldsymbol{\pi}) - \widetilde{L}_n(\widehat{\boldsymbol{\pi}}) \ge \sup_{\boldsymbol{v}\in\partial\, \widetilde{L}_n(\widehat{\boldsymbol{\pi}})} \boldsymbol{v}^\top(\boldsymbol{\pi} - \widehat{\boldsymbol{\pi}}), \qquad \forall \boldsymbol{\pi} \in \mathbb{B}_+^K. \tag{3.84}$$

Using the Karush-Kuhn-Tucker conditions and the fact that $\widehat{\boldsymbol{\pi}}$ minimizes L_n, we get $\mathbf{0}_K \in \partial\, L_n(\widehat{\boldsymbol{\pi}}) = \partial\, \widetilde{L}_n(\widehat{\boldsymbol{\pi}})$. This readily gives $\widetilde{L}_n(\boldsymbol{\pi}) - \widetilde{L}_n(\widehat{\boldsymbol{\pi}}) \ge 0$, for any $\boldsymbol{\pi} \in \mathbb{B}_+^K$. The last step is to remark that $\mathbf{Z}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}) = \bar{\mathbf{Z}}(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi})$, since both $\widehat{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}$ have entries summing to one. □

The core of our results lies in the following proposition which bound the deviations of the empirical process part.

Proposition 3.5.1 (Supremum of Empirical Process)**.** *For any* $\boldsymbol{\pi} \in \mathbb{B}_+^K$ *and* $\boldsymbol{x} \in \mathcal{X}$, *define* $\varphi(\pi, \boldsymbol{x}) = \int (\log f_{\boldsymbol{\pi}}) f^* - \log f_{\boldsymbol{\pi}}(\boldsymbol{x})$ *and consider* $\Phi_n(\boldsymbol{\pi}) = \frac{1}{n}\sum_{i=1}^n \varphi(\boldsymbol{\pi}, \boldsymbol{X}_i)$. *If* $K \ge 2$, *then for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$, *we have*

$$\zeta_n = \sup_{\boldsymbol{\pi}\in\Pi_n} \big\|\nabla\Phi_n(\boldsymbol{\pi})\big\|_\infty \le 8V^3\Big(\frac{\log(K/\delta)}{n}\Big)^{1/2}. \tag{3.85}$$

Furthermore, we have $\mathbf{E}[\zeta_n] \le 4V^3\big(\frac{2\log(2K^2)}{n}\big)^{1/2}$ *and* $\mathbf{Var}[\zeta_n] \le V^2/(2n)$.

Proof. To ease notation, let us denote $g_{\boldsymbol{\pi},l}(x) = \frac{f_l(x)}{f_{\boldsymbol{\pi}}(x)} - \mathbf{E}\big[\frac{f_l(\boldsymbol{X})}{f_{\boldsymbol{\pi}}(\boldsymbol{X})}\big]$ and

$$F(\mathbf{X}) = \sup_{\boldsymbol{\pi}\in\Pi_n} \big\|\nabla\Phi_n(\boldsymbol{\pi})\big\|_\infty = \sup_{(\boldsymbol{\pi},l)\in\Pi_n\times[K]} \Big|\frac{1}{n}\sum_{i=1}^n g_{\boldsymbol{\pi},l}(\boldsymbol{X}_i)\Big|, \tag{3.86}$$

[4] We denote by ∂g the sub-differential of a convex function g.

where $\mathbf{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$. To derive a bound on F, we will use the McDiarmid concentration inequality that requires the bounded difference condition to hold for F. For some $i_0 \in [n]$, let $\mathbf{X}' = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}'_{i_0}, \ldots, \boldsymbol{X}_n)$ be a new sample obtained from $\mathbf{X}$ by modifying the i_0-th element $\boldsymbol{X}_i$ and by leaving all the others unchanged. Then, we have

$$F(\mathbf{X}) - F(\mathbf{X}') = \sup_{(\boldsymbol{\pi},l)\in\Pi_n\times[K]} \left|\frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{\pi},l}(\boldsymbol{X}_i)\right| - \sup_{(\boldsymbol{\pi},l)\in\Pi\times[K]} \left|\frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{\pi},l}(\boldsymbol{X}'_i)\right| \tag{3.87}$$

$$\leq \sup_{(\boldsymbol{\pi},l)\in\Pi_n\times[K]} \left|\frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{\pi},l}(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^{n} g_{\boldsymbol{\pi},l}(\boldsymbol{X}'_i)\right| \tag{3.88}$$

$$= \sup_{(\boldsymbol{\pi},l)\in\Pi_n\times[K]} \left|\frac{1}{n}\Big(g_{\boldsymbol{\pi},l}(\boldsymbol{X}_{i_0}) - g_{\boldsymbol{\pi},l}(\boldsymbol{X}'_{i_0})\Big)\right| \leq \frac{V}{n}, \tag{3.89}$$

where the last inequality is a direct consequence of assumption (3.12). Therefore, using the McDiarmid concentration inequality recalled in Theorem 3.6.3 below, we check that the inequality

$$F(\mathbf{X}) \leq \mathbf{E}(F(\mathbf{X})) + V\sqrt{\frac{\log(1/\delta)}{2n}} \tag{3.90}$$

holds with probability at least $1-\delta$. Furthermore, in view of the Efron-Stein inequality, we have

$$\mathbf{Var}[\zeta_n] = \mathbf{Var}[F(\mathbf{X})] \leq \frac{V^2}{2n}. \tag{3.91}$$

Let us denote $\mathcal{G} := \{(f_l/f_{\boldsymbol{\pi}}) - 1, (\boldsymbol{\pi}, l) \in \Pi_n \times [K]\}$ and $\mathfrak{R}_{n,q}(\mathcal{G})$ the Rademacher complexity of $\mathcal{G}$ given by

$$\mathfrak{R}_n(\mathcal{G}) = \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)\in\Pi_n\times[K]} \left|\frac{1}{n}\sum_{i=1}^{n} \epsilon_i\Big(\frac{f_l(\boldsymbol{X}_i)}{f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)} - 1\Big)\right|\right], \tag{3.92}$$

with $\epsilon_1, \ldots, \epsilon_n$ independent and identically distributed Rademacher random variables independent of $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. Using the symmetrization inequality (see, for instance, Theorem 2.1 in Koltchinskii [2011]) we have

$$\mathbf{E}[F(\mathbf{X})] = \mathbf{E}[\zeta_n] \leq 2\mathbf{E}[\mathfrak{R}_n(\mathcal{G})]. \tag{3.93}$$

Lemma 3.5.3. *The Rademacher complexity defined in (3.92) satisfies*

$$\mathfrak{R}_n(\mathcal{G}) \leq 4V^3\sqrt{\frac{\log K}{n}}. \tag{3.94}$$

Proof. The proof relies on the contraction principle of Ledoux and Talagrand [1991] that we recall in Section 3.6.3 for the convenience. We apply this principle to the random

variables $X_{i,(\boldsymbol{\pi},l)} = f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)/f_l(\boldsymbol{X}_i) - 1$ and to the function $\psi(x) = (1+x)^{-1} - 1$. Clearly ψ is Lipschitz on $[\frac{1}{V} - 1, V - 1]$ with the Lipschitz constant equal to V^2 and $\psi(0) = 0$. Therefore

$$\begin{aligned}
\mathfrak{R}_n(\mathcal{G}) &\le \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)} \frac{1}{n}\sum_{i=1}^n \epsilon_i \psi(\boldsymbol{X}_{i,(\boldsymbol{\pi},l)})\right] + \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)} \frac{1}{n}\sum_{i=1}^n \epsilon_i (-\psi)(\boldsymbol{X}_{i,(\boldsymbol{\pi},l)})\right] \\
&\le 2V^2 \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)\in \Pi_n\times[K]} \frac{1}{n}\sum_{i=1}^n \epsilon_i \boldsymbol{X}_{i,(\boldsymbol{\pi},l)}\right] \\
&= 2V^2 \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)\in \Pi_n\times[K]} \frac{1}{n}\sum_{i=1}^n \epsilon_i \left(\frac{f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\right)\right]. \qquad (3.95)
\end{aligned}$$

Expanding $f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)$ we obtain

$$\begin{aligned}
\mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)} \frac{1}{n}\sum_{i=1}^n \epsilon_i \left(\frac{f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\right)\right] &= \mathbf{E}_\epsilon\left[\sup_{(\boldsymbol{\pi},l)} \sum_{k=1}^K \frac{\pi_k}{n}\sum_{i=1}^n \epsilon_i \left(\frac{f_k(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\right)\right] \\
&= \mathbf{E}_\epsilon\left[\max_{k,l\in[K]} \frac{1}{n}\sum_{i=1}^n \epsilon_i \left(\frac{f_k(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\right)\right]. \qquad (3.96)
\end{aligned}$$

We apply now Theorem 3.6.2 with $s = (k,l)$, $N = K^2$, $a = -V$, $b = V$ and $Y_{i,s} = \epsilon_i\big(\frac{f_k(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\big)$. This yields

$$\mathbf{E}_\epsilon\left[\max_{k,l\in[K]} \frac{1}{n}\sum_{i=1}^n \epsilon_i \left(\frac{f_k(\boldsymbol{X}_i)}{f_l(\boldsymbol{X}_i)} - 1\right)\right] \le 2V\Big(\frac{\log K^2}{2n}\Big)^{1/2}. \qquad (3.97)$$

This completes the proof of the lemma. □

Combining inequalities (3.90,3.93) and Lemma 3.5.3, we get that the inequality

$$F(\mathbf{X}) \le 8V^3\Big(\frac{\log K}{n}\Big)^{1/2} + V\Big(\frac{\log(1/\delta)}{2n}\Big)^{1/2} \qquad (3.98)$$

holds with probability at least $1-\delta$. Noticing that $V \ge 1$ and, for $K \ge 2$, $\delta \in (0, K^{-1/31})$ we have $8\sqrt{\log K} + \sqrt{(1/2)\log(1/\delta)} \le 8\sqrt{\log(K/\delta)}$, we get the first claim of the proposition. The second claim is a direct consequence of Lemma 3.5.3 and (3.93). □

3.6 Proof of the lower bound for nearly-D-sparse aggregation

We prove the minimax lower bound for estimation in Kullback-Leibler risk using the following slightly adapted version of Theorem 2.5 from Tsybakov [2009]. Throughout this

section, we denote by $\lambda_{\min,\mathbf{\Sigma}}(k)$ and $\lambda_{\max,\mathbf{\Sigma}}(k)$, respectively, the smallest and the largest eigenvalue of all $k \times k$ principal minors of the matrix $\mathbf{\Sigma}$.

Theorem 3.6.1. *For some integer $L \geq 4$ assume that $\mathcal{H}_{\mathcal{F}}(\gamma, D)$ contains L elements $f_{\boldsymbol{\pi}^{(1)}}, \ldots, f_{\boldsymbol{\pi}^{(L)}}$ satisfying the following two conditions.*

(i) $\mathrm{KL}(f_{\boldsymbol{\pi}^{(j)}} || f_{\boldsymbol{\pi}^{(k)}}) \geq 2s > 0$, *for all pairs (j,k) such that $1 \leq j < k \leq L$.*

(ii) For product densities f_ℓ^n defined on $\mathcal{X}^n$ by $f_\ell^n(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = f_{\boldsymbol{\pi}^{(\ell)}}(\boldsymbol{x}_1) \times \ldots \times f_{\boldsymbol{\pi}^{(\ell)}}(\boldsymbol{x}_n)$ it holds

$$\max_{\ell \in [L]} \mathrm{KL}(f_\ell^n || f_1^n) \leq \frac{\log L}{16}. \tag{3.99}$$

Then

$$\inf_{\widehat{f}} \sup_{f \in \mathcal{H}_{\mathcal{F}}(\gamma, D)} \mathbf{P}_f\big(\mathrm{KL}(f||\widehat{f}) \geq s\big) \geq 0.17. \tag{3.100}$$

To establish the bound claimed in Theorem 3.3.1, we will split the problem into two parts, corresponding to the following two subsets of $\mathcal{H}_{\mathcal{F}}(\gamma, D)$

$$\begin{aligned} \mathcal{H}_{\mathcal{F}}(0, D) &= \big\{ f_{\boldsymbol{\pi}} : \boldsymbol{\pi} \in \mathbb{B}_+^K \text{ s.t. } \exists\, J \subset [K] \text{ with } \|\boldsymbol{\pi}_{J^c}\|_1 = 0 \text{ and } |J| \leq D \big\}, \\ \mathcal{H}_{\mathcal{F}}(\gamma, 1) &= \big\{ f_{\boldsymbol{\pi}} : \boldsymbol{\pi} \in \mathbb{B}_+^K \text{ s.t. } \pi_1 = 1 - \gamma \text{ and } \textstyle\sum_{j=2}^K \pi_j = \gamma \big\}. \end{aligned} \tag{3.101}$$

We will show that over $\mathcal{H}_{\mathcal{F}}(0, D)$, we have a lower bound of order $\log(1 + K/D)/n$ while over $\mathcal{H}_{\mathcal{F}}(\gamma, 1)$, a lower bound of order $\big[\frac{\gamma^2}{n} \log\big(1 + K/(\gamma\sqrt{n})\big)\big]^{1/2}$ holds true. Therefore, the lower bound over $\mathcal{H}_{\mathcal{F}}(\gamma, D)$ is larger than the average of these bounds.

For any $M \geq 1$ and $k \in [M-1]$, let Ω_k^M be the subset of $\{0,1\}^M$ defined by

$$\Omega_k^M := \Big\{ \boldsymbol{\omega} \in \{0,1\}^M : \|\boldsymbol{\omega}\|_1 = k \Big\}. \tag{3.102}$$

Before starting, we remind here a version of the Varshamov-Gilbert lemma (see, for instance, [Rigollet and Tsybakov, 2011, Lemma 8.3]) which will be helpful for deriving our lower bounds.

Lemma 3.6.1. *Let $M \geq 4$ and $k \in [M/2]$ be two integers. Then there exist a subset $\Omega \subset \Omega_k^M$ and an absolute constant C_1 such that*

$$\|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_1 \geq \frac{k+1}{4} \quad \forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega \text{ s.t. } \boldsymbol{\omega} \neq \boldsymbol{\omega}' \tag{3.103}$$

and $L = |\Omega|$ satisfies $L \geq 4$ and

$$\log L \geq C_1 k \log\Big(1 + \frac{eM}{k}\Big). \tag{3.104}$$

We will also use the following lemma that allows us to relate the KL-divergence $\mathrm{KL}(f_{\boldsymbol{\pi}} || f_{\boldsymbol{\pi}'})$ to the Euclidean distance between the weight vectors $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$.

Lemma 3.6.2. *If the dictionary $\mathcal{F}$ satisfies the boundedness assumption (3.12), then for any $f_{\boldsymbol{\pi}}, f_{\boldsymbol{\pi}'} \in \mathcal{H}_{\mathcal{F}}(\gamma, D)$ we have*

$$\frac{1}{2V^2M}\,\|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\pi}'-\boldsymbol{\pi})\|_2^2 \le \mathrm{KL}(f_{\boldsymbol{\pi}}||f_{\boldsymbol{\pi}'}) \le \frac{V^2}{2m}\,\|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\pi}'-\boldsymbol{\pi})\|_2^2. \tag{3.105}$$

Proof. Using the Taylor expansion, one can check that for any $u \in [1/V, V]$, we have $(1-u)+\frac{1}{2V^2}(u-1)^2 \le -\log u \le (1-u)+\frac{V^2}{2}(u-1)^2$. Therefore,

$$\frac{1}{2V^2}\int_{\mathcal{X}}\Big(\frac{f_{\boldsymbol{\pi}'}}{f_{\boldsymbol{\pi}}}-1\Big)^2 f_{\boldsymbol{\pi}}\,d\nu \le \mathrm{KL}(f_{\boldsymbol{\pi}}||f_{\boldsymbol{\pi}'}) \le \frac{V^2}{2}\int_{\mathcal{X}}\Big(\frac{f_{\boldsymbol{\pi}'}}{f_{\boldsymbol{\pi}}}-1\Big)^2 f_{\boldsymbol{\pi}}\,d\nu. \tag{3.106}$$

Since $\mathcal{F}$ satisfies the boundedness assumption, we get

$$\frac{1}{2MV^2}\int_{\mathcal{X}}\big(f_{\boldsymbol{\pi}'}-f_{\boldsymbol{\pi}}\big)^2 d\nu \le \mathrm{KL}(f_{\boldsymbol{\pi}}||f_{\boldsymbol{\pi}'}) \le \frac{V^2}{2m}\int_{\mathcal{X}}\big(f_{\boldsymbol{\pi}'}-f_{\boldsymbol{\pi}}\big)^2 d\nu. \tag{3.107}$$

The claim of the lemma follows from these inequalities and the fact that $\int_{\mathcal{X}}\big(f_{\boldsymbol{\pi}'}-f_{\boldsymbol{\pi}}\big)^2 d\nu = \|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\pi}'-\boldsymbol{\pi})\|_2^2$. □

3.6.1 Lower bound on $\mathcal{H}_{\mathcal{F}}(0, D)$

We show that the lower bound $(D/n)\log(1+eK/D) \wedge \big((1/n)\log(1+K/\sqrt{n})\big)^{1/2}$ holds when we consider the worst case error for f^* belonging to the set $\mathcal{H}_{\mathcal{F}}(0, D)$.

Proposition 4. *If $\log(1+eK) \le n$ then, for the constant*

$$C_2 = \frac{C_1 m \bar{\kappa}_{\boldsymbol{\Sigma}}(2D, 0)}{2^9 V^2 M(C_1 m \vee 4V^2\lambda_{\max,\boldsymbol{\Sigma}}(2D))} \ge \frac{C_1 m \varkappa_*}{2^9 V^2 M(C_1 m \vee 4V^2\varkappa^*)}, \tag{3.108}$$

we have

$$\inf_{\widehat{f}}\sup_{f\in\mathcal{H}_{\mathcal{F}}(0,D)} \mathbf{P}_f\left(\mathrm{KL}(f||\widehat{f}) \ge C_2\,\frac{D\log(1+\frac{K}{D})}{n} \bigwedge \left(\frac{\log\big(1+\frac{K}{\sqrt{n}}\big)}{n}\right)^{1/2}\right) \ge 0.17. \tag{3.109}$$

Proof. We assume that $D \le K/2$. The case $D > K/2$ can be reduced to the case $D = K/2$ by using the inclusion $\mathcal{H}_{\mathcal{F}}(0, K/2) \subset \mathcal{H}_{\mathcal{F}}(0, D)$. Let us set $A_1 = 4 \vee 16V^2\lambda_{\max,\boldsymbol{\Sigma}}(2D)/(C_1 m)$ and denote by d the largest integer such that

$$d \le D \quad \text{and} \quad d^2\log\Big(1+\frac{eK}{d}\Big) \le A_1 n. \tag{3.110}$$

According to Lemma 3.6.1, there exists a subset $\Omega = \{\boldsymbol{\omega}^{(\ell)} : \ell \in [L]\}$ of Ω_d^K of cardinality $L \ge 4$ satisfying $\log L \ge C_1 d\log(1+eK/d)$ such that for any pair of distinct elements $\boldsymbol{\omega}^{(\ell)}$,

$\boldsymbol{\omega}^{(\ell')} \in \Omega$ we have $\|\boldsymbol{\omega}^{(\ell)} - \boldsymbol{\omega}^{(\ell')}\|_1 \geq d/4$. Using these binary vectors $\boldsymbol{\omega}^{(\ell)}$, we define the set $\mathcal{D} = \{\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(L)}\} \subset \mathbb{B}_+^K$ as follows:

$$\boldsymbol{\pi}^{(1)} = \boldsymbol{\omega}^{(1)}/d, \quad \boldsymbol{\pi}^{(\ell)} = (1-\varepsilon)\boldsymbol{\pi}^{(1)} + \varepsilon\boldsymbol{\omega}^{(\ell)}/d, \quad \ell = 2, \ldots, L. \tag{3.111}$$

Clearly, for every $\varepsilon \in [0,1]$, the vectors $\boldsymbol{\pi}^{(\ell)}$ belong to $\mathbb{B}_+^K$. Furthermore, for any pair of distinct values $\ell, \ell' \in [L]$, we have $\|\boldsymbol{\pi}^{(\ell)} - \boldsymbol{\pi}^{(\ell')}\|_q^q = (\varepsilon/d)^q \|\boldsymbol{\omega}^{(\ell)} - \boldsymbol{\omega}^{(\ell')}\|_1 \geq (\varepsilon/d)^q d/4$. In view of Lemma 3.6.2, this yields

$$\mathrm{KL}(f_{\boldsymbol{\pi}^{(\ell)}} || f_{\boldsymbol{\pi}^{(\ell')}}) \geq \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)}{4V^2 M d} \left\|\boldsymbol{\pi}^{(\ell)} - \boldsymbol{\pi}^{(\ell')}\right\|_1^2 \geq \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2D,0)}{64V^2 M} \times \frac{\varepsilon^2}{d}. \tag{3.112}$$

Let us choose

$$\varepsilon^2 = \frac{d^2 \log(1 + eK/d)}{nA_1}. \tag{3.113}$$

It follows from (3.110) that $\varepsilon \leq 1$. Inserting this value of ε in (3.112), we get

$$\mathrm{KL}(f_{\boldsymbol{\pi}^{(\ell)}} || f_{\boldsymbol{\pi}^{(\ell')}}) \geq 2C_2 \frac{d \log(1 + eK/d)}{n}. \tag{3.114}$$

This inequality shows that condition (i) of Theorem 3.6.1 is satisfied with $s = C_2 \, (d/n) \log(1 + eK/d)$. For the second condition of the same theorem, we have

$$\begin{aligned}
\max_{\ell \in [L]} \mathrm{KL}(f_\ell^n || f_1^n) &= n \max_\ell \mathrm{KL}(f_{\boldsymbol{\pi}^{(\ell)}} || f_{\boldsymbol{\pi}^{(1)}}) && (3.115)\\
&\leq \frac{nV^2 \lambda_{\max,\boldsymbol{\Sigma}}(2d)}{2m} \max_\ell \|\boldsymbol{\pi}^{(\ell)} - \boldsymbol{\pi}^{(1)}\|_2^2 && (3.116)\\
&\leq \frac{nV^2 \lambda_{\max,\boldsymbol{\Sigma}}(2D)}{m} \times \frac{\varepsilon^2}{d}, && (3.117)
\end{aligned}$$

since one can check that $\|\boldsymbol{\pi}^{(\ell)} - \boldsymbol{\pi}^{(1)}\|_2^2 \leq (\varepsilon/d)^2 \|\boldsymbol{\omega}^{(\ell)} - \boldsymbol{\omega}^{(1)}\|_1 \leq 2\varepsilon^2/d$. Therefore, using the definition of ε, we get

$$\begin{aligned}
\max_{\ell \in [L]} \mathrm{KL}(f_\ell^n || f_1^n) &\leq \frac{nV^2 \lambda_{\max,\boldsymbol{\Sigma}}(2D)}{m} \times \frac{C_1 dm \log(1 + eK/d)}{16 nV^2 \lambda_{\max,\boldsymbol{\Sigma}}(2D)} && (3.118)\\
&= \frac{C_1 d \log(1 + eK/d)}{16} \leq \frac{\log L}{16}. && (3.119)
\end{aligned}$$

Theorem 3.6.1 implies that

$$\inf_{\widehat{f}} \sup_{f \in \mathcal{H}_{\mathcal{F}}(0,D)} \mathbf{P}_f \left(\mathrm{KL}(f || \widehat{f}) \geq C_2 \frac{d \log(1 + eK/d)}{n} \right) \geq 0.17. \tag{3.120}$$

We use the fact that d is the largest integer satisfying (3.110). Therefore, either $d+1 > D$ or

$$(d+1)^2 \log\left(1 + \frac{eK}{d+1}\right) \geq A_1 n. \tag{3.121}$$

If $d \geq D$, then the claim of the proposition follows from (3.120), since $d\log(1+eK/d) \geq D\log(1+eK/D)$. On the other hand, if (3.121) is true, then

$$\begin{aligned} d\log(1+eK/d) &\geq \frac{1}{2}(d+1)\log(1+eK/(d+1)) \\ &\geq \frac{1}{2}\big(A_1 n\log(1+eK/(d+1))\big)^{1/2}. \end{aligned} \tag{3.122}$$

In addition, $d^2\log(1+eK/d) \leq A_1 n$ implies that $(d+1)^2 \leq A_1 n$. Combining the last two inequalities, we get the inequality $d\log(1+eK/d) \geq 1/2\big(A_1 n\log(1+eK/\sqrt{A_1 n})\big)^{1/2} \geq \big(n\log(1+eK/\sqrt{n})\big)^{1/2}$. Therefore, in view of (3.120), we get the claim of the proposition. □

3.6.2 Lower bound on $\mathcal{H}_{\mathcal{F}}(\gamma, 1)$

Next result shows that the lower bound $\frac{\gamma^2}{n}\log\left(1+\frac{K}{\gamma\sqrt{n}}\right)$ holds for the worst case error when f^* belongs to the set $\mathcal{H}_{\mathcal{F}}(\gamma,1)$.

Proposition 5. *Assume that*

$$\left(\frac{\log(1+eK)}{n}\right)^{1/2} \leq 2\gamma. \tag{3.123}$$

Then, for the constant $C_3 = \frac{C_1 m\bar{\kappa}_{\mathbf{\Sigma}}(2D,0)}{2^{12}V^4 M\lambda_{\max,\mathbf{\Sigma}}(2D)}$, *it holds that*

$$\inf_{\widehat{f}}\sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,1)} \mathbf{P}_f\left(\mathrm{KL}(f||\widehat{f}) \geq C_3\left\{\frac{\gamma^2}{n}\log\left(1+\frac{K}{\gamma\sqrt{n}}\right)\right\}^{1/2}\right) \geq 0.17. \tag{3.124}$$

Proof. Let $C > 2$ be a constant the precise value of which will be specified later. Denote by d the largest integer satisfying

$$d\sqrt{\log(1+eK/d)} \leq C\gamma\sqrt{n}. \tag{3.125}$$

Note that $d \geq 1$ in view of the condition $(\frac{\log(1+eK)}{n})^{1/2} \leq 2\gamma$ of the proposition. This readily implies that $d \leq C\gamma\sqrt{n}$ and, therefore,

$$\frac{\gamma}{d} \geq C^{-1}\left\{\frac{1}{n}\log\left(1+\frac{eK}{C\gamma\sqrt{n}}\right)\right\}^{1/2} \geq 2C^{-2}\left\{\frac{1}{n}\log\left(1+\frac{K}{\gamma\sqrt{n}}\right)\right\}^{1/2}. \tag{3.126}$$

Let us first consider the case $d \leq (K-1)/2$. According to Lemma 3.6.1, there exists a subset $\Omega \subset \Omega_d^{K-1}$ of cardinality L satisfying $\log L \geq C_1 d\log\left(1+\frac{e(K-1)}{d}\right)$ and $\|\boldsymbol{\omega}^{(\ell)} - \boldsymbol{\omega}^{(\ell')}\|_1 \geq d/4$ for any pair of distinct elements $\boldsymbol{\omega}, \boldsymbol{\omega}'$ taken from Ω. With these binary vectors in hand, we define the set $\mathcal{D} \subset \mathbb{B}_+^K$ of cardinality L as follows:

$$\mathcal{D} = \Big\{\boldsymbol{\pi} = \big(1-\gamma, \gamma\boldsymbol{\omega}/d\big) : \quad \omega \in \Omega\Big\}. \tag{3.127}$$

It is clear that all the vectors of $\mathcal{D}$ belong to $\mathcal{H}_{\mathcal{F}}(\gamma, 1)$. Let us fix now an element of $\mathcal{D}$ and denote it by $\boldsymbol{\pi}^1$, the corresponding element of Ω being denoted by $\boldsymbol{\omega}^1$. We have

$$\max_{\boldsymbol{\pi}\in\mathcal{D}} \mathrm{KL}(f^n_{\boldsymbol{\pi}}||f^n_{\boldsymbol{\pi}^1}) \leq \frac{nV^2}{2m} \max_{\boldsymbol{\pi}\in\mathcal{D}} \|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\pi}-\boldsymbol{\pi}^1)\|_2^2 \tag{3.128}$$

$$\leq \frac{nV^2\lambda_{\max,\boldsymbol{\Sigma}}(2d)\gamma^2}{2md^2} \max_{\boldsymbol{\omega}\in\Omega} \|\boldsymbol{\omega}-\boldsymbol{\omega}^1\|_2^2 \tag{3.129}$$

$$\leq \frac{nV^2\lambda_{\max,\boldsymbol{\Sigma}}(2d)\gamma^2}{md}. \tag{3.130}$$

The definition of d yields $(d+1)\sqrt{\log(1+eK/(d+1))} > C\gamma\sqrt{n}$, which implies that

$$\begin{aligned}\frac{\gamma^2}{d} &\leq 2(d+1)\frac{\gamma^2}{(d+1)^2}\\ &\leq 2(d+1)\frac{\log\big(1+eK/(d+1)\big)}{nC^2}\\ &\leq \frac{4d\log\big(1+e(K-1)/d\big)}{nC^2}.\end{aligned} \tag{3.131}$$

Combined with eq. (3.130), this implies that

$$\max_{\boldsymbol{\pi}\in\mathcal{D}} \mathrm{KL}(f^n_{\boldsymbol{\pi}}||f^n_{\boldsymbol{\pi}^1}) \leq \frac{nV^2\lambda_{\max,\boldsymbol{\Sigma}}(2d)}{m} \times \frac{4d\log\big(1+e(K-1)/d\big)}{nC^2} \tag{3.132}$$

$$= \frac{4V^2\lambda_{\max,\boldsymbol{\Sigma}}(2d)}{mC^2} \times d\log\big(1+e(K-1)/d\big). \tag{3.133}$$

Choosing

$$C^2 = 2 \vee \frac{64V^2\lambda_{\max,\boldsymbol{\Sigma}}(2d)}{C_1 m}$$

we get that $\max_{\boldsymbol{\pi}\in\mathcal{D}} \mathrm{KL}(f^n_{\boldsymbol{\pi}}||f^n_{\boldsymbol{\pi}^1}) \leq \frac{1}{16}C_1 d\log\big(1+e(K-1)/d\big) \leq \frac{\log L}{16}$.

Furthermore, for any $\boldsymbol{\pi}, \boldsymbol{\pi}' \in \mathcal{D}$, in view of Lemma 3.6.2 and (3.126), we have

$$\mathrm{KL}(f_{\boldsymbol{\pi}}||f_{\boldsymbol{\pi}'}) \geq \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)}{4V^2Md}\big\|\boldsymbol{\pi}-\boldsymbol{\pi}'\big\|_1^2 = \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)\gamma^2}{4V^2Md^3}\|\boldsymbol{\omega}-\boldsymbol{\omega}'\|_1^2 \tag{3.134}$$

$$\geq \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)}{64V^2M} \times \frac{\gamma^2}{d} \tag{3.135}$$

$$\geq \frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)}{32V^2MC^2} \times \left\{\frac{\gamma^2}{n}\log\left(1+\frac{K}{\gamma\sqrt{n}}\right)\right\}^{1/2}. \tag{3.136}$$

Since $\frac{\bar{\kappa}_{\boldsymbol{\Sigma}}(2d,0)}{32V^2MC^2} = 2C_3$, this implies that Theorem 3.6.1 can be applied, which leads to the inequality

$$\inf_{\widehat{f}} \sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,1)} \mathbf{P}_f\left(\mathrm{KL}(f||\widehat{f}) \geq C_3\left\{\frac{\gamma^2}{n}\log\left(1+\frac{K}{\gamma\sqrt{n}}\right)\right\}^{1/2}\right) \geq 0.17. \tag{3.137}$$

To complete the proof of the proposition, we have to consider the case $d > (K-1)/2$. In this case, we can repeat all the previous arguments for $d = K/2$ and get the desired inequality. □

3.6.3 Lower bound holding for all densities

Now that we have lower bounds in probability for $\mathcal{H}_{\mathcal{F}}(0, D)$ and $\mathcal{H}_{\mathcal{F}}(\gamma, 1)$, we can derive a lower bound in expectation for $\mathcal{H}_{\mathcal{F}}(\gamma, D)$. In particular, to prove Theorem 3.3.1, we will use the inequality

$$\mathcal{R}\big(\mathcal{H}_{\mathcal{F}}(\gamma, D)\big) \geq \inf_{\widehat{f}} \sup_{f^* \in \mathcal{H}_{\mathcal{F}}(0,D) \cup \mathcal{H}_{\mathcal{F}}(\gamma,1)} \mathbf{E}[\mathrm{KL}(f^* || \widehat{f})]. \tag{3.138}$$

Proof of Theorem 3.3.1. To ease notation, let us define

$$r(n, K, \gamma, D) = \left[\frac{\gamma^2}{n} \log\left(1 + \frac{K}{\gamma\sqrt{n}}\right)\right]^{1/2} + \frac{D \log(1 + K/D)}{n} \bigwedge \left(\frac{\log(1 + K/\sqrt{n})}{n}\right)^{1/2}. \tag{3.139}$$

We first consider the case where the dominating term is the first one, that is

$$\left[\frac{\gamma^2}{n} \log\left(1 + \frac{K}{\gamma\sqrt{n}}\right)\right]^{1/2} \geq \frac{3D \log(1 + K/D)}{n}. \tag{3.140}$$

On the one hand, since $D \geq 1$, we have

$$\frac{3D \log(1 + K/D)}{n} \geq \frac{\log(1 + eK)}{n}. \tag{3.141}$$

On the other hand, using the inequality $\log(1 + x) \leq x$, we get

$$\left[\frac{\gamma^2}{n} \log\left(1 + \frac{K}{\gamma\sqrt{n}}\right)\right]^{1/2} \leq \frac{\gamma}{\sqrt{n}} \left[\log(1 + eK) + \log\left(1 + \frac{1}{e^2\gamma^2 n}\right)\right]^{1/2} \tag{3.142}$$

$$\leq \gamma \left[\frac{\log(1 + eK)}{n}\right]^{1/2} + \frac{\gamma}{\sqrt{n}} \left[\frac{1}{e^2\gamma^2 n}\right]^{1/2} \tag{3.143}$$

$$\leq \gamma \left[\frac{\log(1 + eK)}{n}\right]^{1/2} + \frac{\log(1 + eK)}{2n}. \tag{3.144}$$

Combining (3.140), (3.141) and (3.144), we get

$$\left(\frac{\log(1 + eK)}{n}\right)^{1/2} \leq 2\gamma. \tag{3.145}$$

This implies that we can apply Proposition 5, which yields

$$\inf_{\widehat{f}} \sup_{f \in \mathcal{H}_{\mathcal{F}}(\gamma,D)} \mathbf{P}_f\left(\mathrm{KL}(f||\widehat{f}) \geq C_3 \left\{\frac{\gamma^2}{n} \log\left(1 + \frac{K}{\gamma\sqrt{n}}\right)\right\}^{1/2}\right) \geq 0.17. \tag{3.146}$$

In view of (3.140), this implies that

$$\inf_{\widehat{f}} \sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,D)} \mathbf{P}_f\Big(\mathrm{KL}(f||\widehat{f}) \geq \frac{3}{4}C_3\, r(n,K,\gamma,D)\Big) \geq 0.17. \tag{3.147}$$

We now consider the second case, where the dominating term in the rate is the second one, that is

$$\Big[\frac{\gamma^2}{n}\log\Big(1+\frac{K}{\gamma\sqrt{n}}\Big)\Big]^{1/2} \leq \frac{3D\log(1+K/D)}{n} \bigwedge \Big(\frac{\log(1+K/\sqrt{n})}{n}\Big)^{1/2}. \tag{3.148}$$

In view of Proposition 4, we have

$$\inf_{\widehat{f}} \sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,D)} \mathbf{P}_f\left(\mathrm{KL}(f||\widehat{f}) \geq C_2\,\frac{D\log(1+\frac{K}{D})}{n} \bigwedge \left(\frac{\log\big(1+\frac{K}{\sqrt{n}}\big)}{n}\right)^{1/2}\right) \geq 0.17. \tag{3.149}$$

In view of (3.148), we get

$$\inf_{\widehat{f}} \sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,D)} \mathbf{P}_f\Big(\mathrm{KL}(f||\widehat{f}) \geq \frac{1}{4}C_2\, r(n,K,\gamma,D)\Big) \geq 0.17. \tag{3.150}$$

Thus, we have proved that $\log(1+eK) \leq n$ implies that

$$\inf_{\widehat{f}} \sup_{f\in\mathcal{H}_{\mathcal{F}}(\gamma,D)} \mathbf{P}_f\big(\mathrm{KL}(f||\widehat{f}) \geq C_4\, r(n,K,\gamma,D)\big) \geq 0.17, \tag{3.151}$$

for some constant $C_4 > 0$, whatever the relation between γ and D. The desired lower bound follows now from the Tchebychev inequality $\mathbf{E}\big[\mathrm{KL}(f||\widehat{f})\big] \geq C_4\, r(n,K,\gamma,D)\mathbf{P}_f\big(\mathrm{KL}(f||\widehat{f}) \geq C_4\, r(n,K,\gamma,D)\big)$. □

Appendix A: Concentration inequalities

This section contains some well-known results, which are recalled here for the sake of the self-containedness of the paper.

Theorem 3.6.2. *For each $s = 1,\dots,N$, let $Y_{1,s},\dots,Y_{n,s}$ be n independent and zero mean random variables such that for some real numbers a, b we have $\mathbf{P}(Y_{i,s} \in [a,b]) = 1$ for all $i \in [n]$ and $s \in [N]$. Then, we have*

$$\mathbf{E}\Big[\max_{s\in[N]} \frac{1}{n}\sum_{i=1}^{n} Y_{i,s}\Big] \leq (b-a)\Big(\frac{\log N}{2n}\Big)^{1/2}, \tag{3.152}$$

$$\mathbf{E}\Big[\max_{s\in[N]} \Big|\frac{1}{n}\sum_{i=1}^{n} Y_{i,s}\Big|\Big] \leq (b-a)\Big(\frac{\log(2N)}{2n}\Big)^{1/2}. \tag{3.153}$$

Proof. We denote $Z_s = \frac{1}{n}\sum_{i=1}^n Y_{i,s}$ for $s = 1, \dots, N$ and $Z_s = -\frac{1}{n}\sum_{i=1}^n Y_{i,s}$ for $s = N+1, \dots, 2N$. For every $s \in [2N]$, the logarithmic moment generating function $\psi_s(\lambda) = \log \mathbf{E}[e^{\lambda Z_s}]$ satisfies

$$\psi_s(\lambda) = \log\Big(\prod_i \mathbf{E}[e^{\lambda Y_{i,s}/n}]\Big) = \sum_{i=1}^n \log \mathbf{E}[e^{\lambda Y_{i,s}/n}] \le \frac{\lambda^2 (b-a)^2}{8n}, \tag{3.154}$$

where the last inequality is a consequence of the Hoeffding lemma (see, for instance, Lemma 2.2 in [Boucheron et al., 2013]). This means that Z_s is sub-Gaussian with variance-factor $\nu = (b-a)^2/4n$. Therefore, Theorem 2.5 from [Boucheron et al., 2013] yields $\mathbf{E}[\max_s Z_s] \le \sqrt{2\nu \log(2N)}$, which completes the proof. □

We group and state together the bounded differences and the Efron-Stein inequalities (Boucheron et al. [2013], Theorems 6.2 and 3.1, respectively).

Theorem 3.6.3. *Assume that a function f satisfies the bounded difference condition: there exist constants c_i, $i = 1, \dots, n$ such that for all $i = 1, \dots, n$, all $X = (X_1, \dots, X_i, \dots, X_n)$ and $X' = (X_1, \dots, X_i', \dots, X_n)$ where only the i^{th} vector is changed*

$$|f(X) - f(X')| \le c_i. \tag{3.155}$$

Denote

$$\nu = \sum_{i=1}^n c_i^2. \tag{3.156}$$

Let $Z = f(X_1, \dots, X_n)$ where X_i are independent. Then, for every $\delta \in (0,1)$,

$$\mathbf{P}\Big\{Z \le \mathbf{E}Z + \Big(\frac{\nu \log(1/\delta)}{2}\Big)^{1/2}\Big\} \ge 1 - \delta, \qquad \textit{and} \qquad \mathbf{Var}[Z] \le \frac{\nu}{2}. \tag{3.157}$$

Next we state the contraction principle of [Ledoux and Talagrand, 1991]; a proof can be found in (Boucheron et al. [2013], Theorem 11.6).

Theorem 3.6.4. *Let $x_1, \dots, x_n$ be vectors whose real-valued components are indexed by $\boldsymbol{\mathcal{T}}$, that is, $x_i = (x_{i,s})_{s \in \boldsymbol{\mathcal{T}}}$. For each $i = 1, \dots, n$ let $\varphi_i : \mathbb{R} \to \mathbb{R}$ be a 1-Lipschitz function such that $\varphi_i(0) = 0$. Let $\epsilon_1, \dots, \epsilon_n$ be independent Rademacher random variables, and let $\Psi : [0, \infty) \to \mathbb{R}$ be a non-decreasing convex function. Then*

$$\mathbf{E}\Big[\Psi\Big(\frac{1}{2}\sup_{s \in \boldsymbol{\mathcal{T}}}\Big|\sum_{i=1}^n \epsilon_i \varphi_i(x_{i,s})\Big|\Big)\Big] \le \mathbf{E}\Big[\Psi\Big(\sup_{s \in \boldsymbol{\mathcal{T}}}\Big|\sum_{i=1}^n \epsilon_i x_{i,s}\Big|\Big)\Big] \tag{3.158}$$

$$\mathbf{E}\Big[\Psi\Big(\sup_{s \in \boldsymbol{\mathcal{T}}}\sum_{i=1}^n \epsilon_i \varphi_i(x_{i,s})\Big)\Big] \le \mathbf{E}\Big[\Psi\Big(\sup_{s \in \boldsymbol{\mathcal{T}}}\sum_{i=1}^n \epsilon_i x_{i,s}\Big)\Big]. \tag{3.159}$$

Chapter 4

Experimental Results for the KL-aggregation

Contents

In this section we propose an efficient algorithm for performing the KL-aggregation (see Chapter 3) and describe its implementation. We also compare its performance with different alternative methods. For the sake of simplicity, the comparison with the other methods is done in the univariate case only. The implementation of our algorithm and its behavior are the same in the multivariate setting.

4.1 Introduction

Before anything else, we remind the reader the problem setting and the estimator considered. We observe n independent random vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathcal{X}$ drawn from a probability distribution P^* that admits a density function f^* with respect to the Lebesgue measure. Given a family of mixture components $f_1, \ldots, f_K$, we assumed that this unknown density

is well approximated by a convex combination $f_{\boldsymbol{\pi}}$ of these components:

$$f_{\boldsymbol{\pi}}(\boldsymbol{x}) = \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}), \quad \boldsymbol{\pi} \in \mathbb{B}_+^K = \Big\{ \boldsymbol{\pi} \in [0,1]^K : \sum_{j=1}^{K} \pi_j = 1 \Big\}. \tag{4.1}$$

The component densities $\mathcal{F} = \{f_j : j \in [K]\}$ are assumed to be given by previous experiments or expert knowledge. The problem of construction of this family is an open problem that we try to address in Section 4.5. The objective of this chapter is to expose and study experimentally the algorithm implemented for computing the Maximum Likelihood Estimator (MLE), defined by

$$\widehat{\boldsymbol{\pi}} \in \operatorname*{arg\,min}_{\boldsymbol{\pi} \in \mathbb{B}_+^K} \Big\{ -\frac{1}{n} \sum_{i=1}^{n} \log f_{\boldsymbol{\pi}}(\boldsymbol{X}_i) \Big\}. \tag{4.2}$$

One can note that this problem is convex as the composition of $-\log$ and a linear function is convex. Furthermore, the feasible space is also convex. This problem can be solved via a Primal-Dual interior point method. But we opted for an approach based on the accelerated proximal gradient descent method because of its suitability to the problems in high-dimensions with sparsity assumption [Beck and Teboulle, 2009].

4.2 Implementation

Input: $\boldsymbol{\pi} \in \mathbb{R}^p$.
Output: The projection $\boldsymbol{\pi}^{proj}$ of $\boldsymbol{\pi}$ onto the probability simplex.
1: `Sort` $\boldsymbol{\pi}$ `into` $\boldsymbol{u}$: $u_1 \geq u_2 \geq \cdots \geq u_p$.
2: `Find` $\rho = \max\{1 \leq j \leq p : u_j + \frac{1}{j}(1 - \sum_{i=1}^{j} u_i) > 0\}$.
3: `Define` $\lambda = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} u_i)$.
4: `Construct` $\boldsymbol{\pi}^{proj}$ s.t. $\boldsymbol{\pi}_i^{proj} = \max\{\pi_i + \lambda, 0\}$, $i = 1, \ldots, p$.

Figure 4.1: Projection procedure onto the probability simplex

We can see that eq. (4.2) is equivalent to

$$\operatorname*{arg\,min}_{\boldsymbol{\pi} \in \mathbb{R}^K} \Big\{ -\frac{1}{n} \sum_{i=1}^{n} \log f_{\boldsymbol{\pi}}(\boldsymbol{X}_i) + \chi_{\mathbb{B}_+^K}(\boldsymbol{\pi}) \Big\}, \tag{4.3}$$

where $\chi_{\mathbb{B}_+^K}$ is the indicator function

$$\chi_{\mathbb{B}_+^K}(\boldsymbol{\pi}) = \begin{cases} 0, & \text{if } \boldsymbol{\pi} \in \mathbb{B}_+^K, \\ +\infty, & \text{otherwise.} \end{cases}$$

This problem can be decomposed into

$$\min_{\boldsymbol{\pi}} \big\{\ell(\boldsymbol{\pi}) + g(\boldsymbol{\pi})\big\}, \tag{4.4}$$

where $\ell(\boldsymbol{\pi}) = -\frac{1}{n}\sum_{i=1}^{n} \log f_{\boldsymbol{\pi}}(\boldsymbol{X}_i)$ and $g(\boldsymbol{\pi}) = \chi_{\mathbb{B}_+^K}(\boldsymbol{\pi})$. One can note that this problem is convex but not smooth since ℓ is differentiable but g is not. One way to tackle this minimization is to consider the proximal operator

$$\text{prox}_{\lambda g}(\boldsymbol{\pi}) = \arg\min_{\boldsymbol{u}} \big\{g(\boldsymbol{u}) + \frac{1}{2\lambda}\|\boldsymbol{u} - \boldsymbol{\pi}\|_2^2\big\}, \tag{4.5}$$

where $\lambda > 0$ is a scale parameter for the function g. One can interpret $\text{prox}_{\lambda g}(\boldsymbol{\pi})$ as a point that compromises between minimizing g and being near to $\boldsymbol{\pi}$. Note that in our context, $g(.) = \chi_{\mathbb{B}_+^K}(.)$, therefore

$$\begin{aligned}
\text{prox}_{\lambda g}(\boldsymbol{\pi}) &= \arg\min_{\boldsymbol{u}} \big\{\chi_{\mathbb{B}_+^K}(\boldsymbol{u}) + \frac{1}{2\lambda}\|\boldsymbol{u} - \boldsymbol{\pi}\|_2^2\big\}, \\
&= \arg\min_{\boldsymbol{u} \in \mathbb{B}_+^K} \big\{\|\boldsymbol{u} - \boldsymbol{\pi}\|_2^2\big\}, \\
&= \boldsymbol{\Pi}_{\mathbb{B}_+^K}(\boldsymbol{\pi})
\end{aligned}$$

where $\boldsymbol{\Pi}_{\mathbb{B}_+^K}(\boldsymbol{\pi})$ is the Euclidean projection of $\boldsymbol{\pi}$ into the probability simplex. The reader can find in [Parikh and Boyd, 2014] a detailed study of proximal algorithms. A particularly interesting procedure for our problem is the proximal gradient method that solves eq. (4.4). This method is iterative and the $(k+1)^{th}$ step is

$$\boldsymbol{\pi}^{k+1} := \text{prox}_{\lambda^k g}(\boldsymbol{\pi}^k - \lambda^k \nabla f(\boldsymbol{\pi}^k)), \tag{4.6}$$

where $\lambda^k > 0$ is a step size. This step size can be found via a line-search method [Parikh and Boyd, 2014]. However, if ∇f is L-Lipschitz, we can chose a fixed $\lambda^k \in (0, 1/L)$. In this setting, one can show that this method converges with a rate of $\mathcal{O}(1/k)$. This rate is known to be sub-optimal. To improve this slow rate, accelerated versions of the proximal gradient method have been developed [Nesterov, 2007, Beck and Teboulle, 2009] that achieve optimal $\mathcal{O}(1/k^2)$ rate under the L-Lipschitz condition on ∇f. These optimization methods rely on the proximal operator and Nesterov's accelerated gradient method [Nesterov, 1983]. A version of this accelerated method is

$$\begin{cases}
\boldsymbol{\xi}^k & := \boldsymbol{\pi}^k + \omega^k(\boldsymbol{\pi}^k - \boldsymbol{\pi}^{k-1}), \\
\boldsymbol{\pi}^{k+1} & := \text{prox}_{\lambda^k g}(\boldsymbol{\xi}^k - \lambda^k \nabla f(\boldsymbol{\xi}^k)),
\end{cases}$$

where ω^k is defined by $\omega^1 := 1$ and

$$\omega^k := \frac{2(\omega^{k-1} - 1)}{1 + \sqrt{1 + (\omega^{k-1})^2}}.$$

This method has been coined Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) in [Beck and Teboulle, 2009]. Our procedure is a special case of this algorithm that can be called "Accelerated projected gradient descent" since the proximal is the projection into $\mathbb{B}_+^K$. A procedure for the projection onto the probability simplex can be found in [Duchi et al., 2008] and a simple proof in [Wang et al., 2013]. The procedure for this projector is given in Figure 4.1. Finally, the complete procedure for our algorithm is given in Figure 4.2.

1: **Input:** A gradient step λ.
2: **Output:** parameter estimate $\widehat{\boldsymbol{\pi}}$.
3: `1:` `Initialize` $t_0 = 1$ `and` $\boldsymbol{\pi}_0 = (1/K, \ldots, 1/K)$,
4: **for** $k \geq 1$, until convergence occurs, **do**
5: (a) $\boldsymbol{\pi}_k = \boldsymbol{\Pi}_{\mathbb{B}_+^K}\big(\xi_k - \lambda \nabla f_{\xi_k}(\xi_k)\big)$,
6: (b) $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
7: (c) $\xi_{k+1} = \boldsymbol{\pi}_k + \big(\frac{t_k - 1}{t_{k+1}}\big)(\boldsymbol{\pi}_k - \boldsymbol{\pi}_{k-1})$.
8: **end for**.

Figure 4.2: FISTA for the estimation of $\boldsymbol{\pi}$.

A nice property of this method is that it provides a sparse solution of this minimization problem which fits with our goal of selecting elements of the dictionary. General Primal-Dual interior points methods do not offer this feature.

4.3 Alternative methods considered

In this section we briefly describe several estimators of the density which are compared to our estimator. Note that although we used the algorithm EM in our experiments, we do not described it in this section since it is already done in Chapter 1.

4.3.1 SPADES

A method combining the dictionary approach and the ℓ_1-penalty (and, therefore, very close in spirit to our method) have been proposed by [Bunea et al., 2010]. They studied the

linear combinations (as opposed to convex combinations studied in the previous chapter) of functions $\{f_1, \ldots, f_M\}$ with $f_j \in L_2(\mathbb{R}^d)$, $j = 1, \ldots, M$:

$$f_\lambda(x) = \sum_{j=1}^{M} \lambda_j f_j(x), \quad \lambda = (\lambda_1, \ldots, \lambda_M) \in \mathbb{R}^M. \tag{4.7}$$

They suggested the following estimator $\widehat{\lambda}$ called SPADES:

$$\widehat{\lambda} = \underset{\lambda \in \mathbb{R}^M}{\arg\min} \Big\{ -\frac{2}{n} \sum_{i=1}^{n} f_\lambda(\boldsymbol{X}_i) + \|f_\lambda\|^2 + 2 \sum_{j=1}^{M} \omega_j |\lambda_j| \Big\}. \tag{4.8}$$

It could be interesting to include SPADES in our experimental evaluation, but we did not manage to find an easy-to-use implementation of it, and it turned out that our implementation was quite slow. Furthermore, the SPADES is conceptually close to the Adaptive Dantzig (AD) [Bertin et al., 2011] procedure described in the next subsection. Therefore, we opted for excluding SPADES from our experiments but including AD.

4.3.2 Adaptive Dantzig density estimation

The Adaptive Dantzig estimator of a density has been introduced in [Bertin et al., 2011]. This method is similar to ours as it constructs an estimator of the unknown density from a linear mixture of functions taken from a dictionary. The key idea of this estimator is to minimize the ℓ_1-norm of the weight vector of the linear combination under an adaptive Dantzig constraint. This constraint comes from concentration inequalities. We recall here some material about the Dantzig selector. It has been introduced by [Candes and Tao, 2007] in the linear regression model

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\lambda}_0 + \boldsymbol{\epsilon} \tag{4.9}$$

where $\boldsymbol{Y} \in \mathbb{R}^n$, $\boldsymbol{A}$ is a n by M matrix, $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is the noise vector and $\boldsymbol{\lambda}_0 \in \mathbb{R}^M$ the unknown regression parameter to estimate. The Dantzig estimator is then defined as the solution of the problem

$$\text{minimize } \|\boldsymbol{\lambda}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\lambda} - \boldsymbol{Y})\|_\infty \leq \eta, \tag{4.10}$$

where η is a regularization parameter. Statistical properties of this estimator were established in [Bickel et al., 2009]. They considered the non-parametric regression framework

$$Y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \ldots, n \tag{4.11}$$

where f is an unknown function, the design points $(x_i)_{i=1,\dots,n}$ are known and $(\epsilon_i)_{i=1,\dots,n}$ is a noise vector. One can estimate f_0 as a weighted sum $f_{\boldsymbol{\lambda}_0}$ of elements of a dictionary $D = (\varphi_m)_{m=1,\dots,M}$

$$f_{\boldsymbol{\lambda}_0} = \sum_{i=1}^{M} \lambda_{0,m}\varphi_m. \tag{4.12}$$

One easily checks that the model in 4.11 coincides with model in eq. (4.9) if we choose as design matrix $\boldsymbol{A} = (\varphi_m(x_i))$. The goal of [Bertin et al., 2011] was to estimate an unknown density f_0 with respect to a known measure dx on $\mathbb{R}$ by using the observation of n-sample $X_1, \dots, X_n$ and to build a linear combination f_λ of elements of the dictionary D as in eq. (4.12). It follows from the strong law of large numbers that

$$\widehat{\beta}_m = \frac{1}{n}\sum_{i=1}^{n} \varphi_m(X_i)$$

converges almost surely to the scalar product of f_0 and φ_m:

$$\int \varphi_m(x) f_0(x) dx = \beta_{0,m}, \tag{4.13}$$

and the Gram matrix associated to the dictionary D

$$G_{m,m'} = \int \varphi_m(x)\varphi_{m'}(x)dx \quad \text{with} \quad 1 \le m, m' \le M. \tag{4.14}$$

The scalar product of $f_{\boldsymbol{\lambda}}$ and φ_m is therefore

$$\int \varphi_m(x) f_{\boldsymbol{\lambda}}(x)dx = \sum_{m'=1}^{M} \lambda_{m'} \int \varphi_{m'}(x)\varphi_m(x)dx = (\boldsymbol{G}\boldsymbol{\lambda})_m. \tag{4.15}$$

The Dantzig estimate $\widehat{\boldsymbol{\lambda}}^D$ is then obtained by solving the following constrained minimization problem

$$\begin{cases} \text{minimize} & \|\boldsymbol{\lambda}\|_1 \\ \text{subject to} & |(\boldsymbol{G}\boldsymbol{\lambda})_m - \widehat{\beta}_m| \le \eta_{\gamma,m} \quad m \in \{1, \dots, M\}, \end{cases}$$

where, for a constant $\gamma > 0$,

$$\eta_{\gamma,m} = \sqrt{\frac{2\tilde{\sigma}_m^2 \gamma \log M}{n}} + \frac{2\|\varphi_m\|_\infty \gamma \log M}{3n}, \tag{4.16}$$

with

$$\tilde{\sigma}_m^2 = \widehat{\sigma}_m^2 + 2\|\varphi_m\|_\infty \sqrt{\frac{2\widehat{\sigma}_m^2 \gamma \log M}{n}} + \frac{8\|\varphi_m\|_\infty^2 \gamma \log M}{n}, \tag{4.17}$$

and

$$\widehat{\sigma}_m^2 = \frac{1}{n(n-1)} \sum_{i=2}^{n} \sum_{j=1}^{i-1} (\varphi_m(X_i) - \varphi_m(X_j)). \tag{4.18}$$

Note that $\eta_{\gamma,m}$ depends on the data which explains the name *Adaptive Dantzig*. [Bertin et al., 2011] derived the form of $\eta_{\gamma,m}$ from sharp concentration inequalities (see Theorem 1 of [Bertin et al., 2011]). More precisely, if we consider $\boldsymbol{\lambda}_0 = (\lambda_{0,m})_{m=1,\dots,M}$ such that the projection of f_0 on the space spanned by D is

$$\mathbf{P}_D f_0 = \sum_{m=1}^{M} \lambda_{0,m} \varphi_m, \tag{4.19}$$

then $(\boldsymbol{G}\boldsymbol{\lambda}_0)_m = \beta_{0,m}$ and the parameter $\eta_{\gamma,m}$ can be seen as the smallest quantity such that, for $\gamma > 1$, we have $|\beta_{0,m} - \widehat{\beta}_m| \leq \eta_{\gamma,m}$ with high probability. Note that the assumption $\gamma > 1$ is an almost necessary condition to have a theoretical control on the quadratic error $\mathbf{E}\|\widehat{f}^D - f_0\|_2^2$. Therefore, we will follow the choice of $\gamma = 1.01$ made by the authors in our experiments. The pseudo code of the procedure is given in Figure 4.3. In what follows, the Adaptive Dantzig density estimator is noted $\widehat{f}^{AD}$ and the abbreviation AD is used in the plots.

4.3.3 Kernel density estimation

The kernel density estimator (KDE) is a well established non-parametric way of estimating the probability density function of a random variable. We will recall in this section some material about KDE.

Let $X_1, \dots, X_n$ be i.i.d. random variables drawn from an unknown probability density f with respect to the Lebesgue measure on $\mathbb{R}$. The kernel density estimator $\widehat{f}_h$ is given by

$$\widehat{f}_h(x) \triangleq \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{X_i - x}{h}\Big) \tag{4.22}$$

where $K : \mathbb{R} \to \mathbb{R}$ and $\int K(u)du = 1$ is called a kernel and h is the bandwidth. We used Gaussian kernel and three methods to select the bandwidth: Cross Validation, Scott's rule of thumb which is the default method in Scipy [Jones et al., 2001–] and the Sheather and Jones bandwidth selection procedure [Sheather and Jones, 1991].

1: **Input:** A sample $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n \in \mathbb{R}^p$ and the dictionary $D = (\varphi_m)_{m=1,\dots,M}$.
2: **Output:** Dantzig density estimate $\widehat{f}^{AD} = f_{\widehat{\boldsymbol{\lambda}}^D}$.
3: **Init:** Set $\gamma = 1.01$.
4: Compute $\widehat{\beta}_m = \frac{1}{n}\sum_{i=1}^N \varphi_m(\boldsymbol{X}_i)$.
5: Compute $\widehat{\sigma}_m^2 = \frac{1}{n(n-1)}\sum_{i=2}^n \sum_{j=1}^{i-1}(\varphi_m(\boldsymbol{X}_i) - \varphi_m(\boldsymbol{X}_j))^2$.
6: Compute $\tilde{\sigma}_m^2$.

$$\tilde{\sigma}_m^2 = \widehat{\sigma}_m^2 + 2\|\varphi_m\|_\infty \sqrt{\frac{2\widehat{\sigma}_m^2 \gamma \log M}{n}} + \frac{8\|\varphi_m\|_\infty^2 \gamma \log M}{n}. \tag{4.20}$$

7: Compute $\eta_{\gamma,m}$

$$\eta_{\gamma,m} = \sqrt{\frac{2\tilde{\sigma}_m^2 \gamma \log M}{n}} + \frac{2\|\varphi_m\|_\infty \gamma \log M}{3n}.$$

8: Compute the coefficients $\widehat{\boldsymbol{\lambda}}^{D,\gamma}$ of the Dantzig estimate, $\widehat{\boldsymbol{\lambda}}^{D,\gamma} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^M} \|\boldsymbol{\lambda}\|_1$ such that $\boldsymbol{\lambda}$ satisfies the Dantzig constraint

$$\forall m \in \{1, \dots, m\}, \quad |(\boldsymbol{G}\boldsymbol{\lambda})_m - \widehat{\beta}_m| \le \eta_{\gamma,m}. \tag{4.21}$$

9: Compute the mixture density $f_{\widehat{\boldsymbol{\lambda}}^D} = \sum_{m=1}^M \widehat{\lambda}_m^D \varphi_m$.

Figure 4.3: Adaptive Dantzig density estimation procedure

Methods based on minimizing the AMISE

The most natural way to derive an estimator of the bandwidth would be to minimize the Mean Integrated Squared Error (MISE)

$$\mathrm{MISE}(h) := \mathbb{E}\Big[\int (\widehat{f}_h(x) - f(x))^2 dx\Big]. \tag{4.23}$$

Unfortunately, we can not rely on this quantity since f is unavailable. However, we can derive the first two terms of the asymptotic expansion of the MISE (AMISE). When $n \to \infty$ and $h = h(n) \to 0$, and under regularity assumptions on f and K, we have

$$\mathrm{AMISE}(h) = \frac{1}{nh} R(K) + \frac{h^4 \sigma_K^4}{4} R(f''), \tag{4.24}$$

where for an appropriate function g,

$$R(g) = \int g^2(x)dx \qquad \text{and} \qquad \sigma_g^2 = \int x^2 g(x)dx.$$

The reader can refer to the appendix of [Tsybakov, 2009] for a proof of this expansion. Setting the derivative w.r.t. h of the right hand side of eq. (4.24) to 0, we see that a suitable estimate of the bandwidth would be the solution of

$$h = \left(\frac{R(K)}{\sigma_K^4 R(f'')} \right)^{1/5} n^{-1/5}. \tag{4.25}$$

However, this cannot be done directly since we do not know $R(f'')$. In the special case where we consider that the kernels are Gaussian and the target density to be estimated is also a Gaussian with density $\phi_{(0,\sigma^2)}$, we have $R(\phi''_{(0,\sigma^2)}(x)) = 3/(8\sqrt{\pi}\sigma^5)$ and we can derive the Scott's rule of thumb in univariate case [Scott, 2015]

$$\widehat{h} = (4/3)^{1/5} \sigma n^{-1/5} \approx 1.06 \widehat{\sigma} n^{-1/5}. \tag{4.26}$$

Without this assumption on the target density, we have to look deeper into the study of $R(f'')$. Several estimators of this quantity has been developed to circumvent this issue [Hall and Marron, 1987, Jones and Sheather, 1991, Sheather and Jones, 1991]. We will focus on a popular method from [Sheather and Jones, 1991]. The authors constructed a kernel density estimator of $R(f'')$

$$\widehat{S}(\widehat{\alpha}_2(h)) = \frac{1}{n(n-1)} (\widehat{\alpha}_2(h))^{-5} \sum_{i=1}^{n} \sum_{j=1}^{n} \Phi^{(4)}\Big(\frac{X_i - X_j}{\widehat{\alpha}_2(h)} \Big), \tag{4.27}$$

where $\Phi^{(i)}$ is the i^{th} derivative of the standard normal density. Note that $\widehat{\alpha}_2(h)$ depends on h. An estimator of $\widehat{\alpha}_2(h)$ can be built with specific properties on the diagonal elements of eq. (4.27)

$$\widehat{\alpha}_2(h) = 1.357 \big(\widehat{S}(a) / \widehat{T}(b) \big)^{1/7} h^{5/7}, \tag{4.28}$$

with

$$\widehat{T}(b) = -\frac{1}{n(n-1)} b^{-7} \sum_{i=1}^{n} \sum_{j=1}^{n} \Phi^{(6)}\Big(\frac{X_i - X_j}{b} \Big), \tag{4.29}$$

and

$$a = 0.920 \widehat{\lambda} n^{-1/7}, \quad b = 0.912 \widehat{\lambda} n^{-1/9}, \tag{4.30}$$

where $\widehat{\lambda}$ is the sample interquartile range. We will not go into the details of these expressions but it is worth mentioning that $\widehat{T}(b)$ is a kernel density estimator of $R(f''')$. Therefore combining eq. (4.27), eq. (4.28) and eq. (4.29), we can solve eq. (4.25) over h via a Newton-Raphson procedure. The algorithm is given in Figure 4.4.

1: **Input:** A sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{R}$.
2: **Output:** A bandwidth estimator $\widehat{h}$.
3: **Init:** Set $a = 0.920\widehat{\lambda}n^{-1/7}$ and $b = 0.912\widehat{\lambda}n^{-1/9}$.
4: Compute

$$\widehat{T}(b) = -\frac{1}{n(n-1)}b^{-7}\sum_{i=1}^{n}\sum_{j=1}^{n}\Phi^{(6)}\Big(\frac{X_i - X_j}{b}\Big). \tag{4.31}$$

5: Compute

$$\widehat{S}(a) = \frac{1}{n(n-1)}a^{-5}\sum_{i=1}^{n}\sum_{j=1}^{n}\Phi^{(4)}\Big(\frac{X_i - X_j}{a}\Big) \tag{4.32}$$

6: Define the function $\widehat{\alpha}_2(h) = 1.357\big(\widehat{S}(a)/\widehat{T}(b)\big)^{1/7}h^{5/7}$.
7: Solve over h

$$h - \left(\frac{R(K)}{\sigma_K^4\widehat{S}(\widehat{\alpha}_2(h))}\right)^{1/5} = 0. \tag{4.33}$$

Figure 4.4: Sheather and Jones bandwidth selection method.

Behavior of KDE in high dimension

It is well known that the kernel density estimator performs badly in the high dimensional setting, [Stone, 1980] proved that the kernel density estimator of a p times continuously differentiable density in dimension d converges at most at the rate $n^{-p/(2p+d)}$. Therefore, for a given target error, the size of the sample must increase exponentially as the dimension increases. For a study of kernel density estimators in the high dimensional setting, see Chapter 7 of [Scott, 2015].

4.4 Experimental Evaluation

In order to carry out an experimental evaluation, we constructed a set of target densities with different shapes and recorded the performances of the estimators. We considered different density dictionaries. Finally we assessed the performance through the Kullback-Leibler divergence and the L_2 distance. All the experiments reported in this section were conducted in the univariate case.

4.4.1 Dictionaries considered

We did experiments with the following two dictionaries containing various types of densities.

1. The first dictionary, denoted by D_{GL}, is composed of Gaussian and Laplace densities. The Gaussian densities have their means in the set $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and their variances in $\{0.001, 0.01, 0.1, 1\}$. The Laplace densities have their means in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and their scales in $\{0.05, 0.1, 0.2, 0.5, 1\}$. Therefore, the dictionary D_{GL} has 54 elements. The plots of these functions are depicted in Figure 4.5.

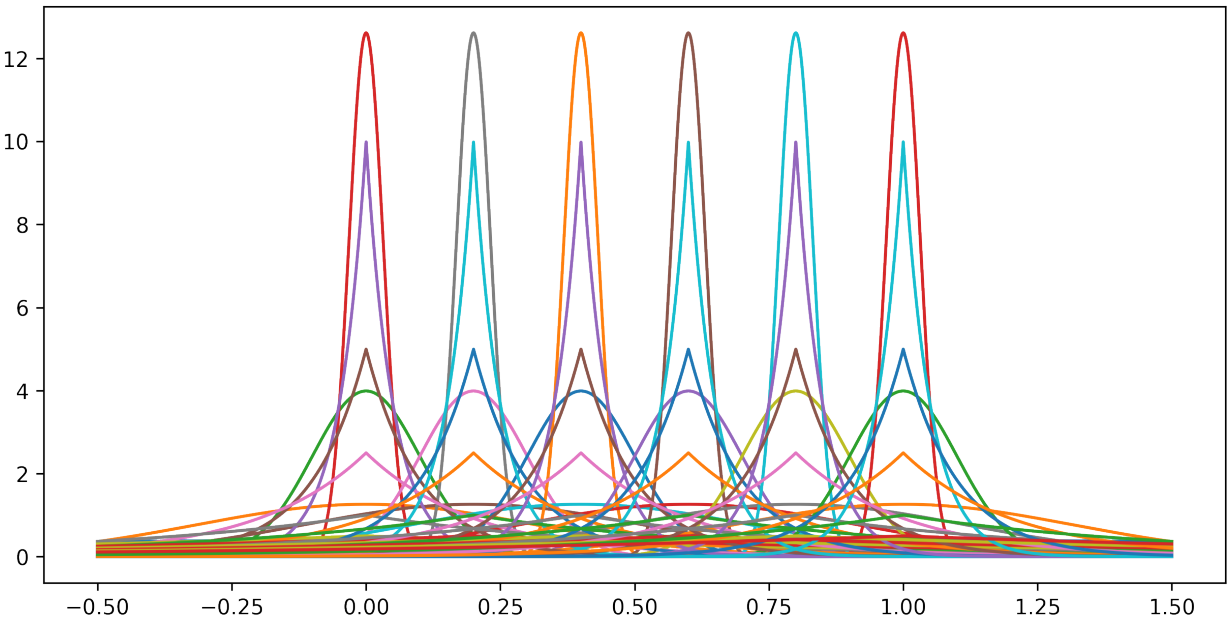

Figure 4.5: D_{GL}, set of Gaussian and Laplace densities.

2. The second dictionary, denoted by D_{GLU}, is obtained by enriching the first dictionary D_{GL} by the set of 10 uniform densities on the intervals $(i, i+0.1)$, $i \in \{0, 0.1, \ldots, 0.9\}$. This dictionary D_{GLU} has 64 elements.

A table of the dictionary D_{GL} (and D_{GLU} with the uniform densities) can be found in Figure 4.17.

4.4.2 Densities considered

We considered 5 target densities corresponding to 5 different scenarios. The 1^{st} and 2^{nd} will asses the performance of our method on uniform based densities, the 3^{rd} and 4^{th} on

dictionary based density. The last one is a complex density made from elements which are not in the dictionary that we will consider.

1. f_{unif}: A uniform density on $[0,1]$.

2. f_{rect}: A mixture of uniform densities on subintervals. This density is called "Rectangular":
$$f_{\text{rect}} = \frac{10}{7}\mathbf{1}_{[0,1/5]} + \frac{5}{7}\mathbf{1}_{[1/5,2/5]} + \frac{10}{7}\mathbf{1}_{[2/5,3/5]} + \frac{10}{7}\mathbf{1}_{[4/5,1]}. \tag{4.34}$$

3. f_{gauss}: A mixture of 5 Gaussian densities taken from the dictionary D_{GL} equally centered in $[0,1]$ with same variance:
$$f_{\text{gauss}} = \sum_{k=1}^{5} 0.2 f_k \quad \text{with} \quad f_k = \varphi_{(k/5,0.001)}. \tag{4.35}$$

4. $f_{\text{gauss-lapl}}$: A mixture of 5 Gaussian and Laplace densities taken from the dictionary D_{GL} with different variances and scales:
$$f_{\text{gauss-lapl}} = 0.2\big(\begin{array}{l} \varphi_{(0,10^{-2})} + \varphi_{(0.2,10^{-3})} + \varphi_{(0.6,10^{-3})} \\ \quad + \text{Lapl}_{(0.4,0.2)} + \text{Lapl}_{(0.8,0.1)}\big). \end{array} \tag{4.36}$$

5. f_{ext}: A mixture of Gaussian and Laplace densities taken from another dictionary D_{out}:
$$f_{\text{ext}} = \sum_{k=1}^{7} \frac{1}{7} f_k \quad \text{with} \quad f_k \in D_{out}. \tag{4.37}$$

These target densities are plotted in Figure 4.6.

4.4.3 Discussion of the results

In the numerical experiments reported in this section, the dictionaries used for the Adaptive Dantzig and the Maximum likelihood density estimators are D_{GL} and D_{GLU}. Note that the AD is the direct competitor of the MLE as both methods rely on a dictionary. However, in order to get a broader insight of what is going on, we also compared these dictionary based methods with other commonly used density estimators such as the EM algorithm on Gaussian mixtures with a model selection performed by the BIC criterion and Kernel Density Estimators (KDE). In the plots, KDE refers to the kernel density estimate with Scott's rule as chosen by default in the Python library Scipy, KDE-SJ refers to the

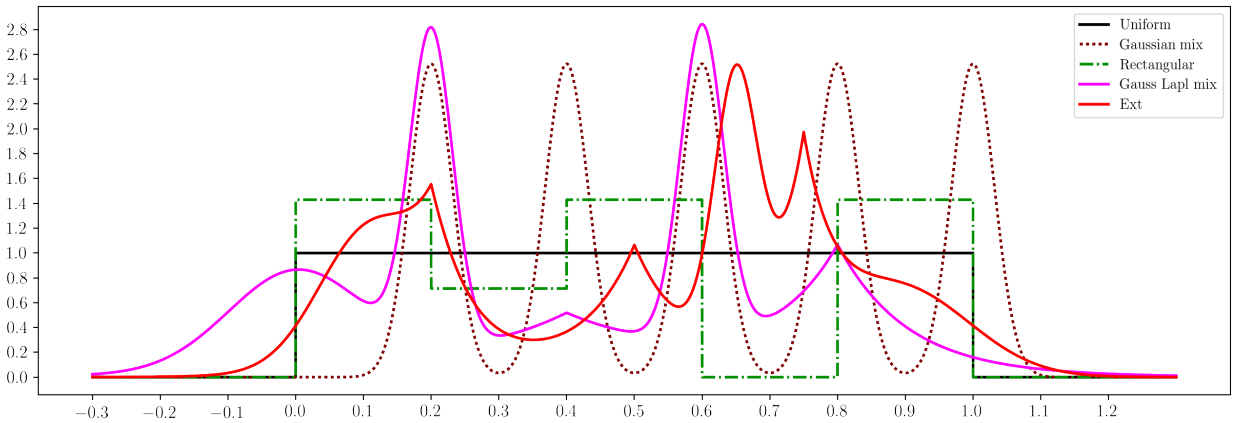

Figure 4.6: Five target densities considered in the experiments.

KDE with the Sheather-Jones bandwidth selector and KDE CV refers to the KDE with bandwidth selected via cross-validation. The two latter were implemented by ourselves.

For each scenario of the target density, f_{unif}, f_{rect}, f_{gauss}, $f_{\text{gauss-lapl}}$, f_{ext} and for each sample size N with $N \in \{100, 500, 1000\}$, we ran 200 simulations. The boxplots of the errors are plotted in Figure 4.7-Figure 4.19. The running times of different arguments are depicted in Figure 4.20. A rapid observation is that the performance of the MLE is good both in Kullback-Leibler and L_2 losses, and it outperforms in all considered scenarios the AD estimator. This is true both in terms of statistical accuracy and computational complexity. The comparison with the other estimation methods is more subtle, and requires a closer look to the results.

Mis-specification bias

Obviously, the densities f_{unif}, f_{rect} and f_{ext} were not built with elements in the dictionary D_{GL}. In other terms, they do not lie in the convex hull of the dictionary D_{GL}. Furthermore, they can be hardly approximated by convex combinations of functions from D_{GL}. Therefore, it is clear that whatever the dictionary based approach we use, it will have a significant bias due to the "model mis-specification".

Since the cardinality of the dictionary is chosen independently of the sample size n, this bias term is constant across different values of n. This is exactly what we observe in Figure 4.7. Such methods as the EM-BIC or various versions of KDE have an L_2 error that decreases significantly when the sample size increases, whereas the AD and, especially, the MLE show only a slight improvement of the error. This is a strong indication of the fact that the bias of the methods AD and MLE substantially dominates the bias, when the true

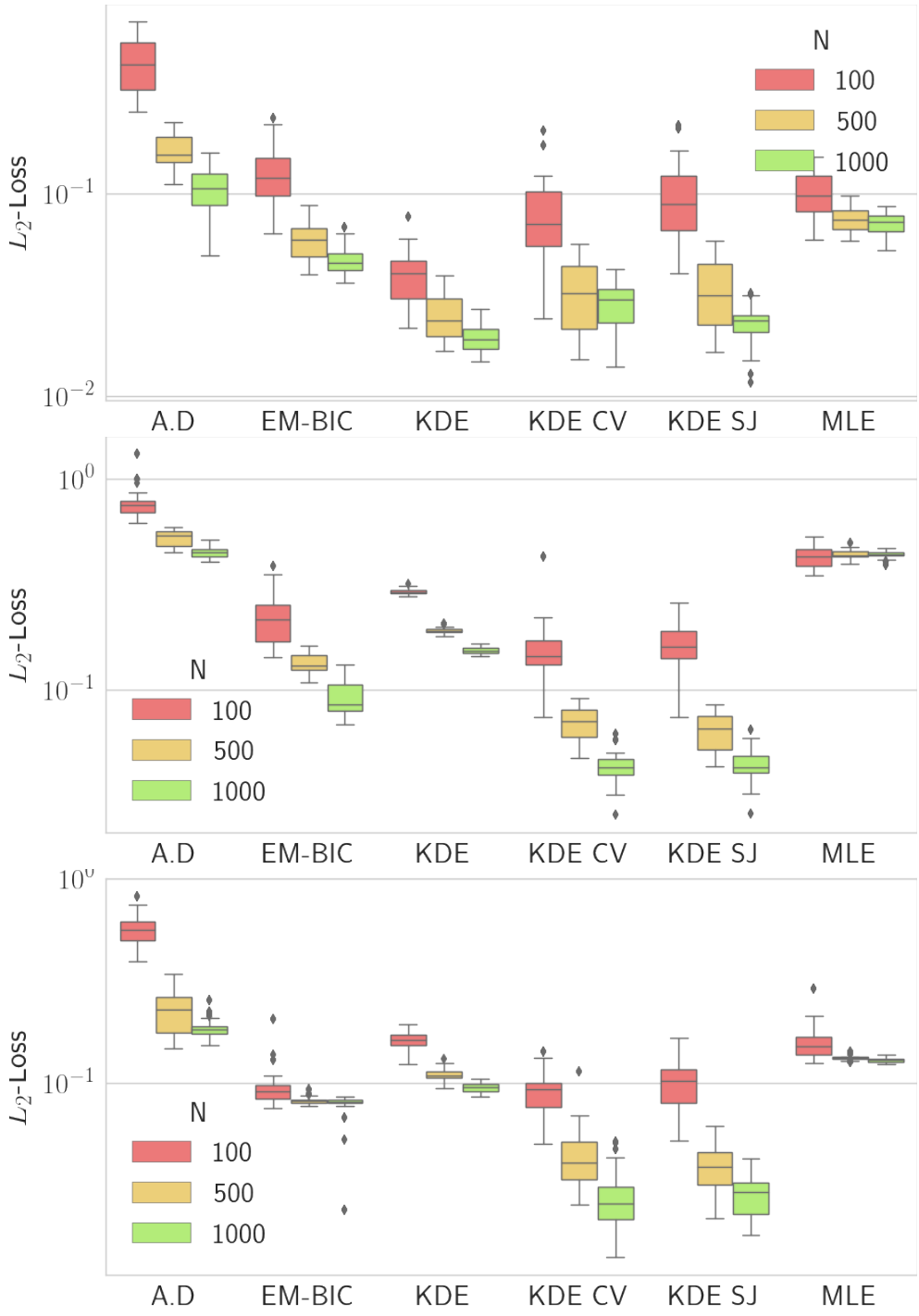

Figure 4.7: Results with f_{unif} (upper panel), f_{rect} (middle panel) and f_{ext} (lower panel) in $L2$ loss with D_{GL}.

density is chosen from the set $\{f_{\text{unif}}, f_{\text{rect}}, f_{\text{ext}}\}$. Thus, the apparently poor behavior of the MLE as compared to the EM-BIC and the KDE is not a surprise and, more importantly, it is not caused by the method of estimation itself but rather by the inappropriate choice of the dictionary.

Note that in all the experiments, the conclusions drawn from the error bars corresponding to the L_2-error can be drawn from the error bars corresponding to the KL-error.

Assessing estimation error

While for the three densities discussed in the foregoing paragraph the bias was largely dominating the variance, the situation is reversed for the densities f_{gauss} and $f_{\text{gauss-lapl}}$. Both of them belong to the convex hull of the dictionary D_{GL}, which implies that the mis-specification bias vanishes. Therefore, the error is mostly dominated by the estimation variance. This explains why for these two densities the MLE has the smallest error, both in L_2 and KL loss (see Figure 4.9 and Figure 4.10). Interestingly, the second best is EM-BIC, which performs better than the AD. Note that the default KDE in Scipy [Jones et al., 2001–] with Scott's rule presents poor results in these scenarios. This observation should come to mind of the practitioner when applying kernel density estimators with default package setting.

One can also remark that the error of the MLE when estimating f_{gauss} is smaller than the one of estimating $f_{\text{gauss-lapl}}$. This is perfectly in line with the theory developed in previous chapter, telling that the variance term is proportional to the sparsity index. In these examples, the sparsity index of $f_{\text{gauss-lapl}}$ is larger than that of f_{gauss}.

Impact of the choice of the dictionary

The discussion of the foregoing paragraphs demonstrates the importance of the choice of the dictionary. The purpose of the additional experiments conducted with the same target densities but with a larger dictionary, D_{GLU}, is to further illustrate this importance and to show that the size of the dictionary does not significantly impact the quality of estimation[1].

The inclusion of 10 uniform densities on $(0, 0.1), \ldots, (0.9, 1)$ to the dictionary D_{GL} removes the mis-specification bias in the case of a uniform and rectangular densities, and reduces it in the case of f_{ext}. The results are plotted in Figure 4.11 and Figure 4.18. We can see that the MLE becomes generally the best estimator when the density is uniform or rectangular. It is still slightly worse than the KDE with data-driven bandwidths for estimating f_{ext}. Finally, the results for the densities f_{gauss} and $f_{\text{gauss-lapl}}$ plotted on Fig-

[1] It certainly does impact the running time

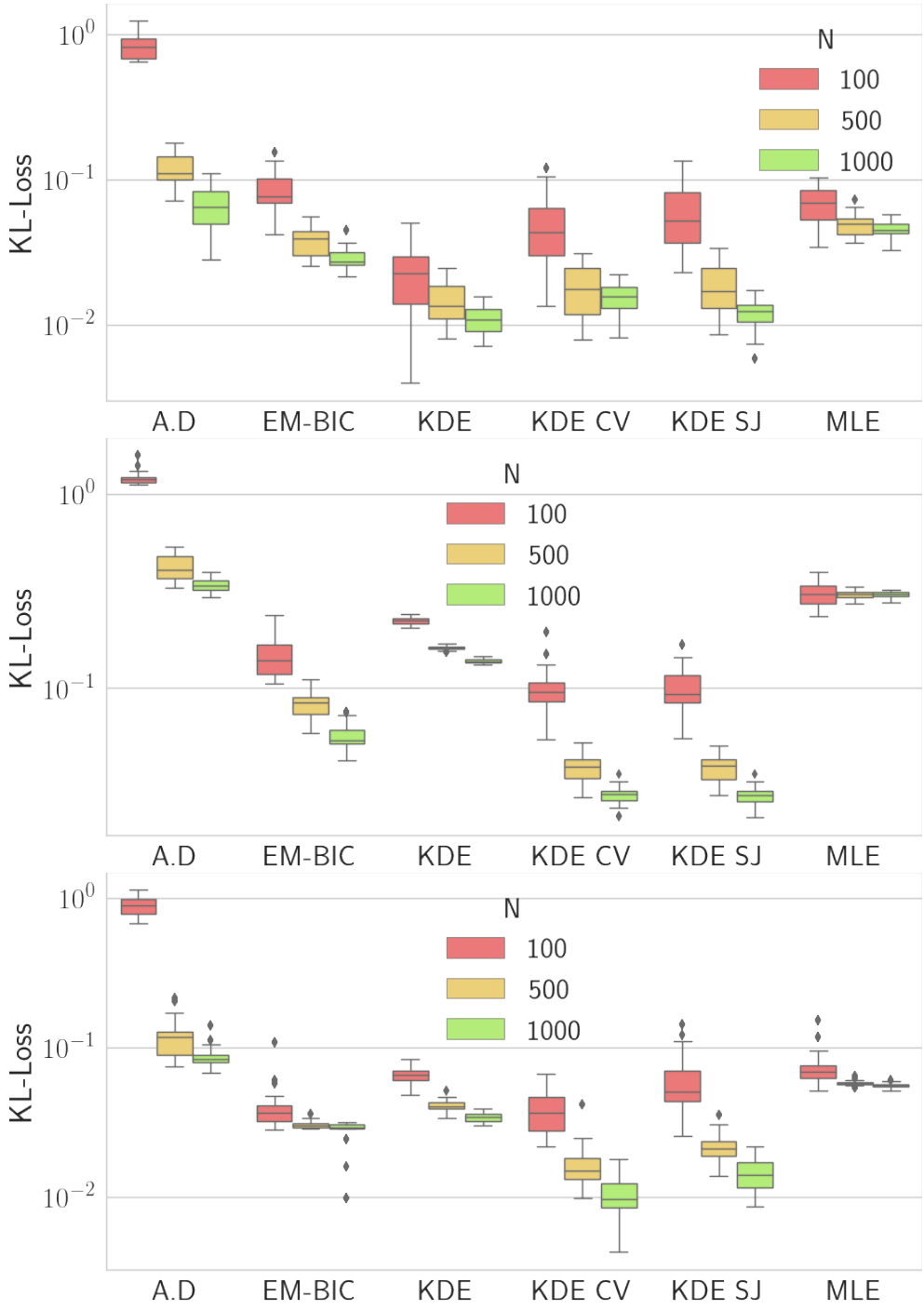

Figure 4.8: Results with f_{unif} (upper panel), f_{rect} (middle panel) and f_{ext} (lower panel) in KL loss with D_{GL}.

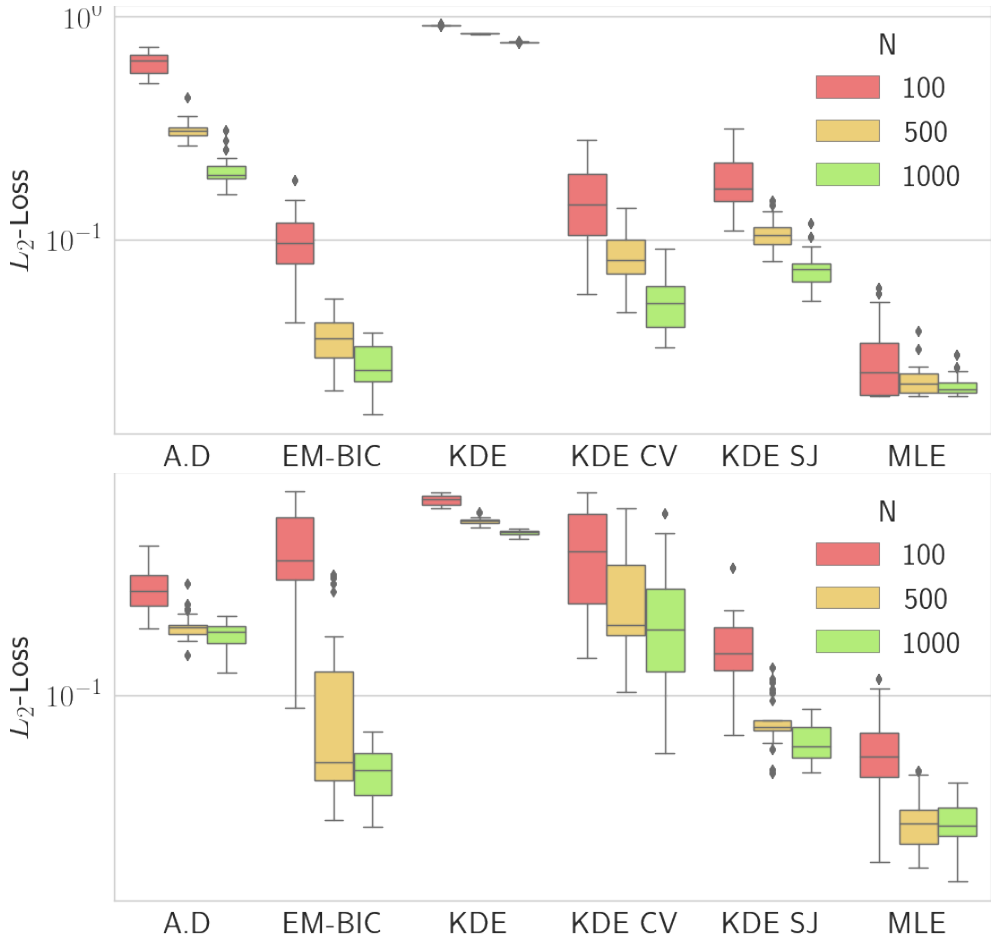

Figure 4.9: Results with f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel) in L_2 loss with D_{GL}.

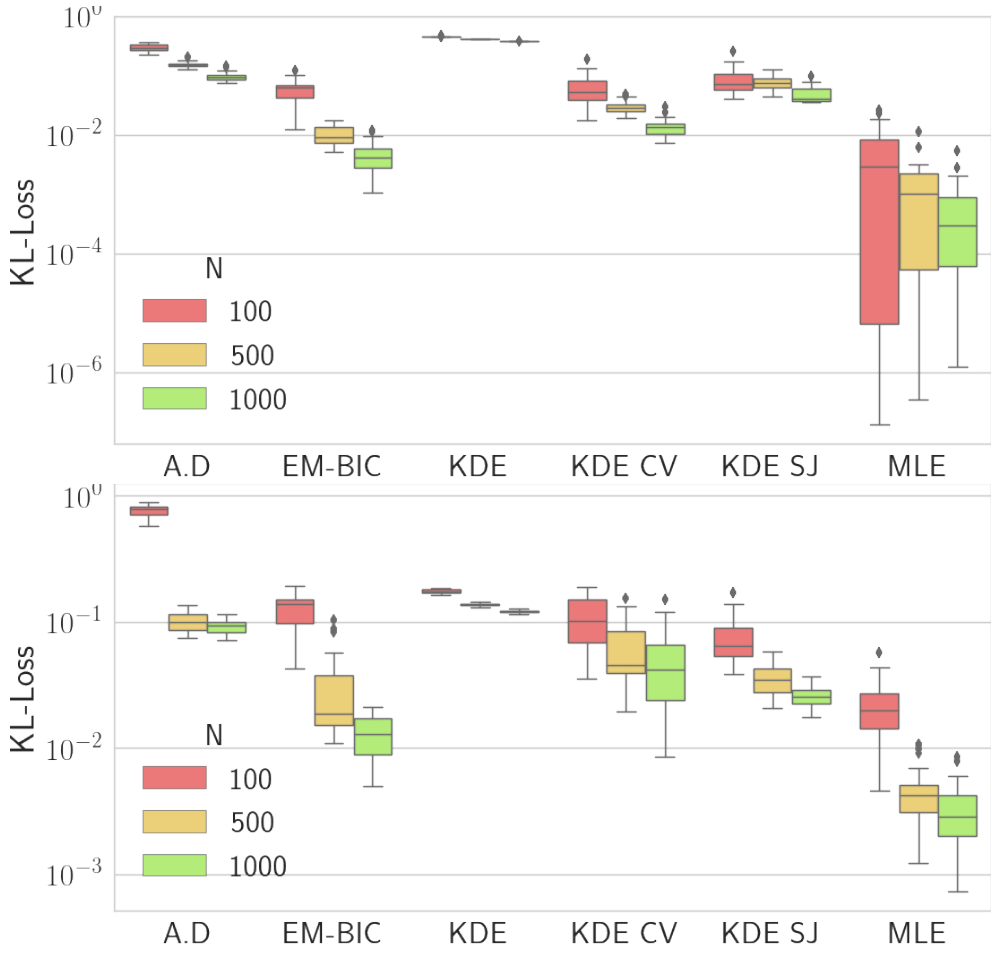

Figure 4.10: Results with f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel) in KL loss with D_{GL}.

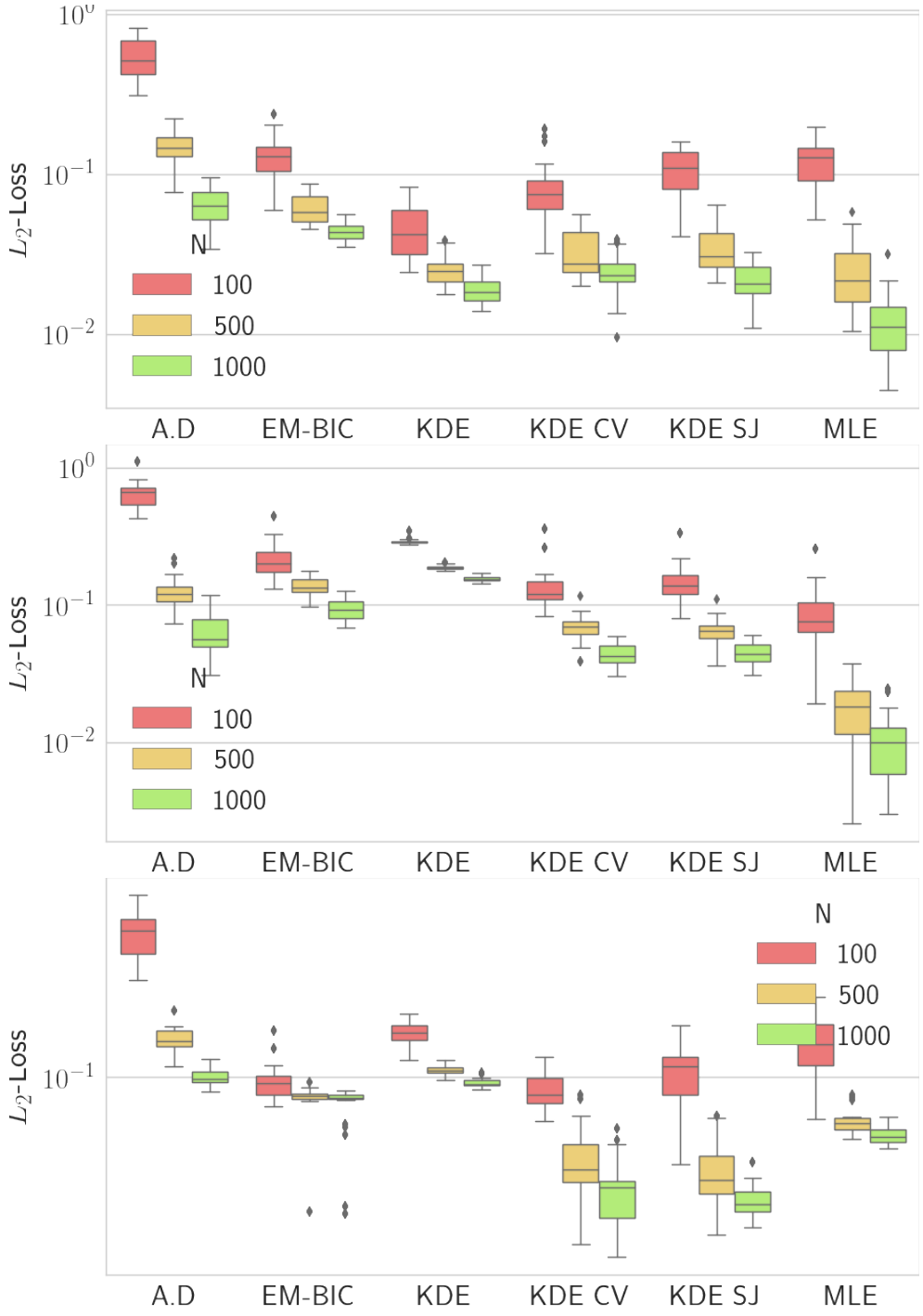

Figure 4.11: Results with f_{unif} (upper panel), f_{rect} (middle panel) and f_{ext} (lower panel) in L_2 loss with D_{GLU}.

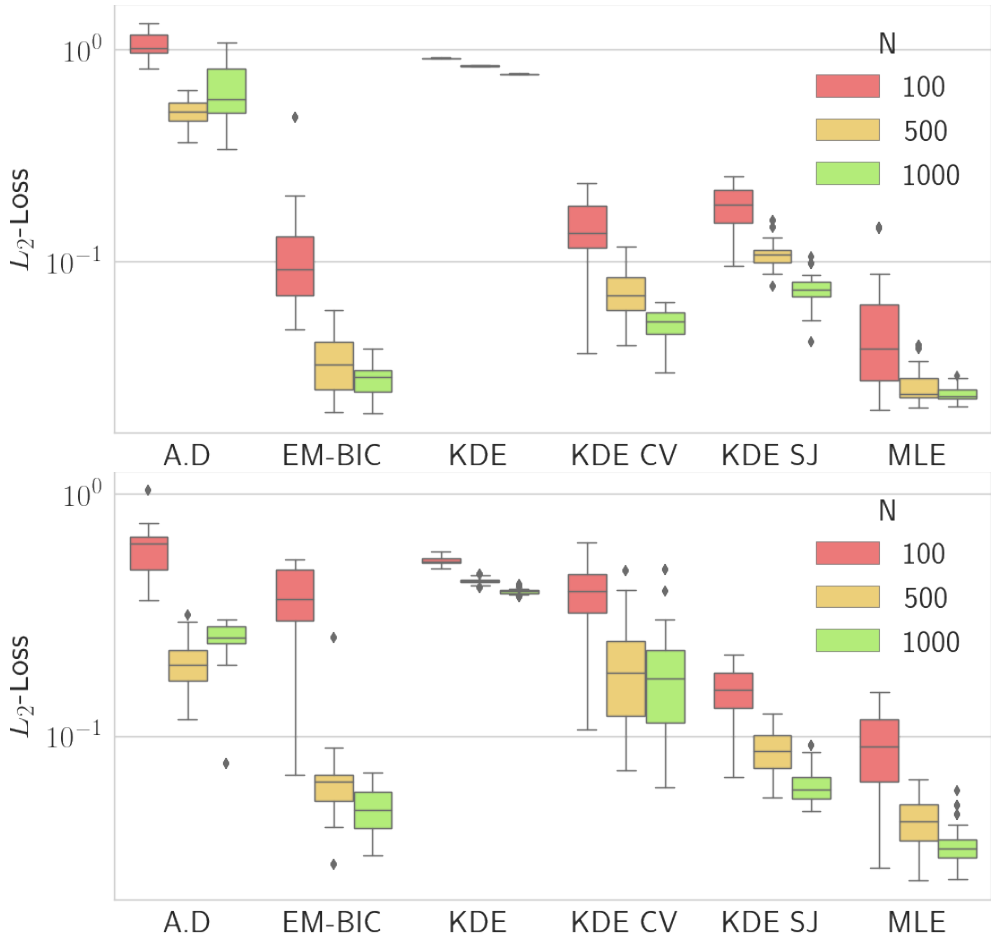

Figure 4.12: Results with f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel) in $L2$ loss with D_{GLU}.

ure 4.12 and Figure 4.19 confirm that adding new elements to the dictionary (even if they are "useless") do not deteriorate the quality of estimation. The ℓ_1-constraint allow us to avoid the overfitting.

Comparison of weights estimated by AD and MLE

A closer look on the estimated weights by AD and MLE gives us knowledge on the behavior of these estimators. We considered the full dictionary D_{GLU} and we provided a table of the indexes of components of this dictionary in Figure 4.17. We plotted the estimated weights of the true components of f_{gauss} and $f_{\text{gauss-lapl}}$ in Figure 4.13 and Figure 4.15. The MLE estimates correctly the real weights of f_{gauss} and most of the weights of $f_{\text{gauss-lapl}}$. We recall the reader that those weights were set to 0.2. However, AD did not succeed to

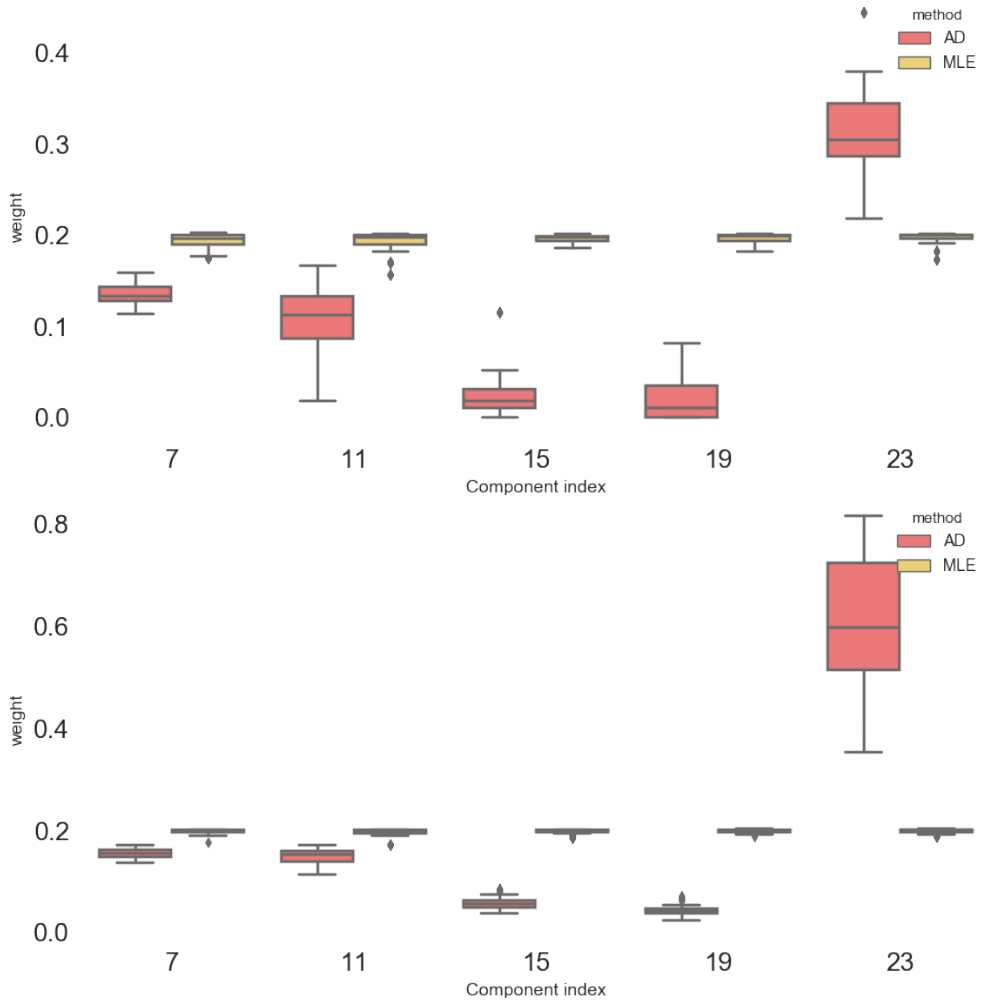

Figure 4.13: Estimated weights of the components of f_{gauss}, with $N = 500$ (upper panel) and $N = 1000$ (lower panel).

estimate correctly these weights. It turns out that AD gave importance on components that overlap the true densities of the mixture as shown in Figure 4.14 with the uniform components. Both AD and MLE provide sparse estimators, this can be seen by looking at components not used in the dictionary (see Figure 4.16). As a matter of fact, the estimated weight vector by AD is more sparse that MLE, but AD is more prone to be influenced by overlapping densities.

Concluding remarks

To conclude, the performance of the MLE method in these simulations is promising to achieve a good mixture density estimate. In addition, the computational efficiency of the

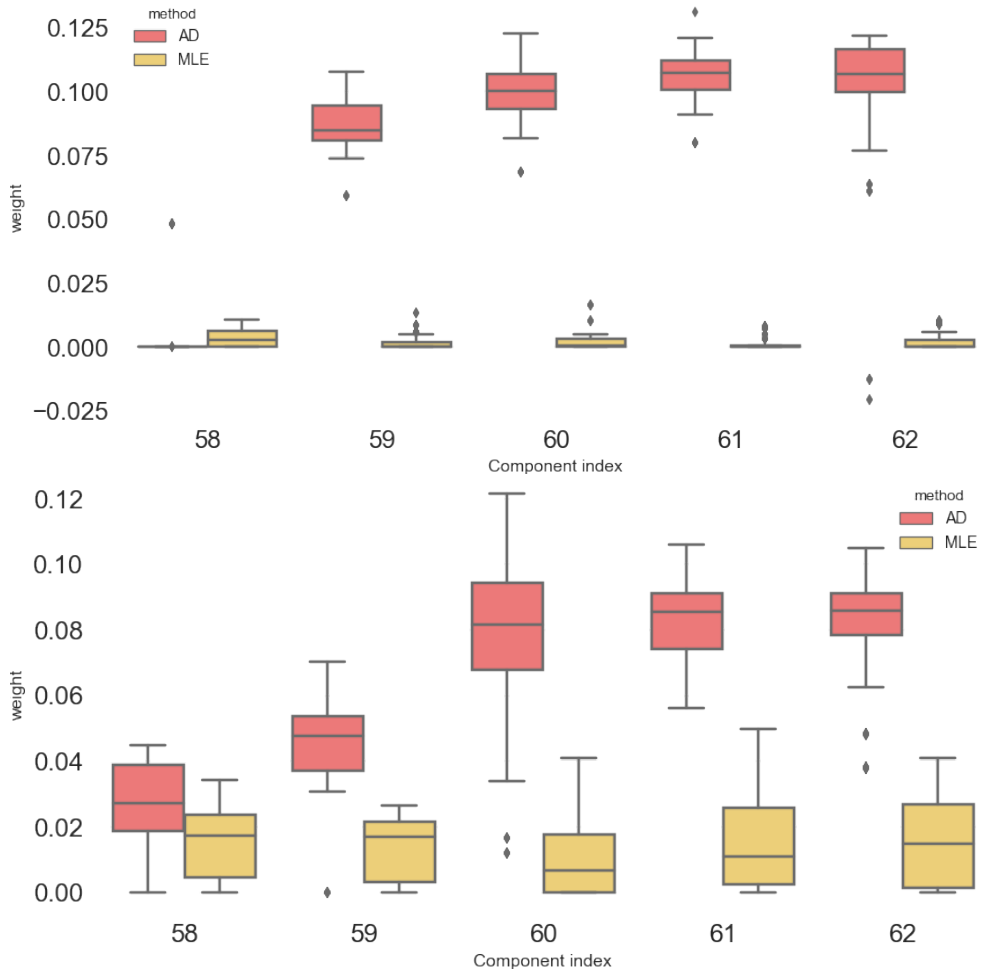

Figure 4.14: Estimated weights of uniform components of the dictionary for f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel) with $N = 1000$.

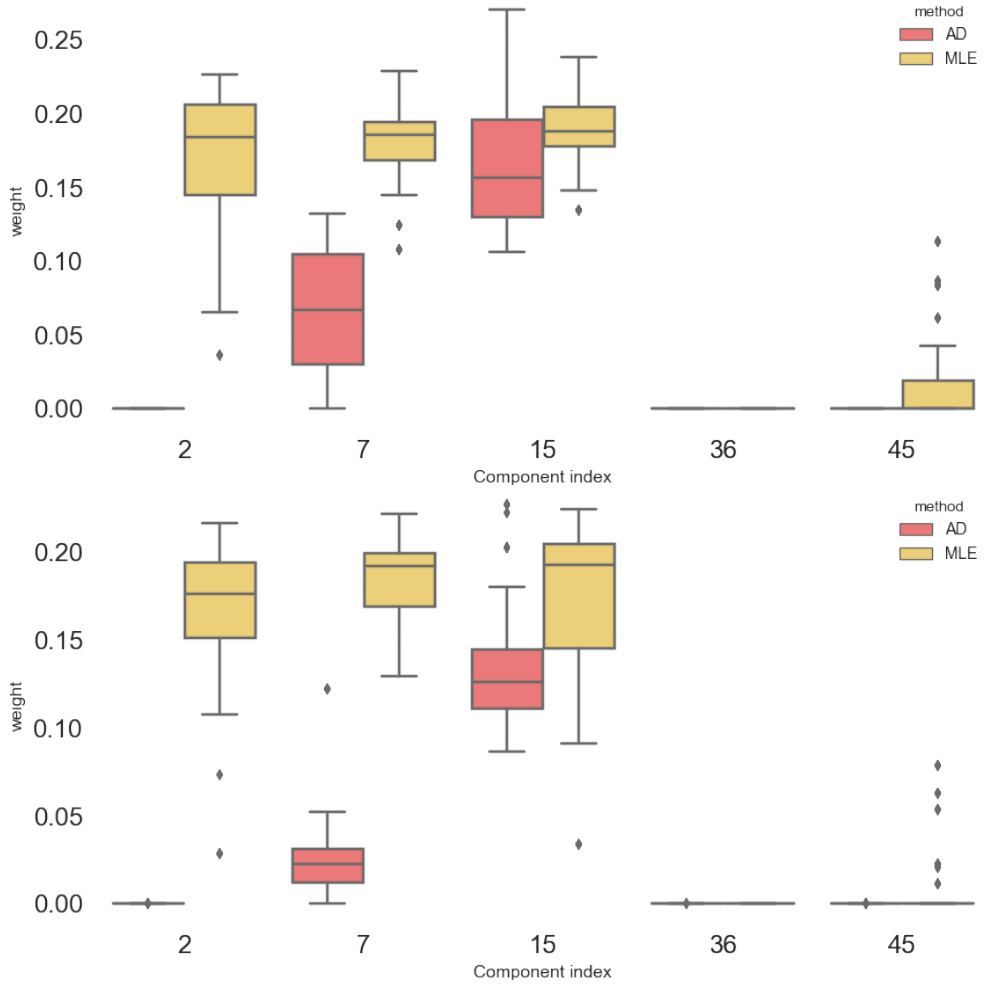

Figure 4.15: Estimated weights of the components of $f_{\text{gauss-lapl}}$, with $N = 500$ (upper panel) and $N = 1000$ (lower panel).

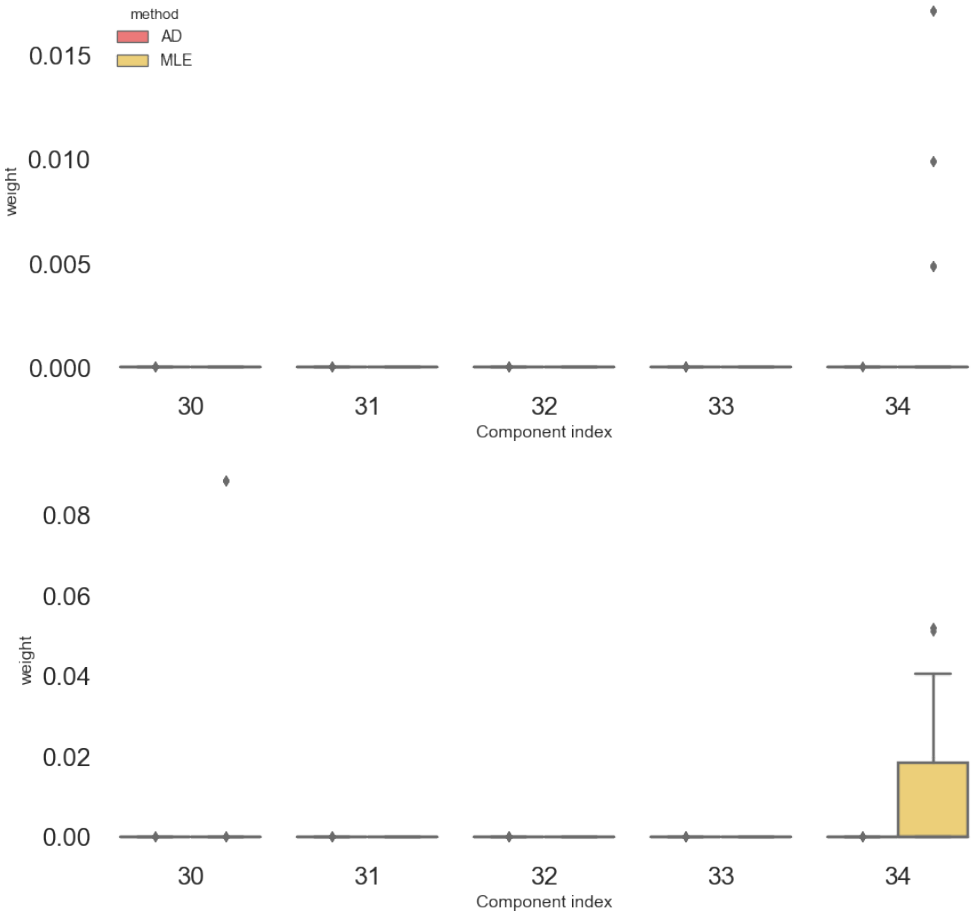

Figure 4.16: Estimated weights of non-used components of the dictionary for f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel), with $N = 1000$.

0	Normal(0 , 1)	22	Normal(1 , 0.01)	44	Laplace(0.8 , 0.05)
1	Normal(0 , 0.1)	23	Normal(1 , 0.001)	45	Laplace(0.8 , 0.1)
2	Normal(0 , 0.01)	24	Laplace(0 , 0.05)	46	Laplace(0.8 , 0.2)
3	Normal(0 , 0.001)	25	Laplace(0 , 0.1)	47	Laplace(0.8 , 0.5)
4	Normal(0.2 , 1)	26	Laplace(0 , 0.2)	48	Laplace(0.8 , 1)
5	Normal(0.2 , 0.1)	27	Laplace(0 , 0.5)	49	Laplace(1 , 0.05)
6	Normal(0.2 , 0.01)	28	Laplace(0 , 1)	50	Laplace(1 , 0.1)
7	Normal(0.2 , 0.001)	29	Laplace(0.2 , 0.05)	51	Laplace(1 , 0.2)
8	Normal(0.4 , 1)	30	Laplace(0.2 , 0.1)	52	Laplace(1 , 0.5)
9	Normal(0.4 , 0.1)	31	Laplace(0.2 , 0.2)	53	Laplace(1 , 1)
10	Normal(0.4 , 0.01)	32	Laplace(0.2 , 0.5)	54	Uniform(0.0 , 0.1)
11	Normal(0.4 , 0.001)	33	Laplace(0.2 , 1)	55	Uniform(0.1 , 0.2)
12	Normal(0.6 , 1)	34	Laplace(0.4 , 0.05)	56	Uniform(0.2 , 0.3)
13	Normal(0.6 , 0.1)	35	Laplace(0.4 , 0.1)	57	Uniform(0.3 , 0.4)
14	Normal(0.6 , 0.01)	36	Laplace(0.4 , 0.2)	58	Uniform(0.4 , 0.5)
15	Normal(0.6 , 0.001)	37	Laplace(0.4 , 0.5)	59	Uniform(0.5 , 0.6)
16	Normal(0.8 , 1)	38	Laplace(0.4 , 1)	60	Uniform(0.6 , 0.7)
17	Normal(0.8 , 0.1)	39	Laplace(0.6 , 0.05)	61	Uniform(0.7 , 0.8)
18	Normal(0.8 , 0.01)	40	Laplace(0.6 , 0.1)	62	Uniform(0.8 , 0.9)
19	Normal(0.8 , 0.001)	41	Laplace(0.6 , 0.2)	63	Uniform(0.9 , 1.0)
20	Normal(1 , 1)	42	Laplace(0.6 , 0.5)		
21	Normal(1 , 0.1)	43	Laplace(0.6 , 1)		

Figure 4.17: Indexes of components of the dictionary D_{GL} and D_{GLU}

MLE displayed in Figure 4.20 makes it highly attractive for performing density estimation. Our algorithm was coded in Python with some elements accelerated with the Just-In-Time (JIT) compiler Numba [Lam et al., 2015]. Compared to compiled optimized versions of KDE and EM from Scipy and Scikit-Learn[Pedregosa et al., 2011], we are confident that the computation time of our algorithm can be further decreased. Another important point is in the case of high dimensional data, KDE and EM+BIC methods are known to present poor performance. Our method needs the computation of the matrix $(f_j(X_i))_{(i,j)\in[N]\times[K]}$ which might consume a lot of memory. Some techniques such as a Mini-batch approach can help. Furthermore, at the light of the results in the uniform and rectangular case, the choice of the dictionary is a cornerstone in density estimation. The size of the dictionary should be chosen by considering both statistical arguments and computational limitations.

4.5 A method for constructing the dictionary of densities

In this section, we propose a data-driven method to construct a dictionary of densities for the KL-aggregation algorithm. We compare mixture densities estimated by this dictionary generation method and the KL-aggregation algorithm with the Kernel density estimator with the bandwidth selected via cross-validation and the Expectation-Maximization algorithm with the BIC criterion in different dimensional settings. We show experimentally that the KL-aggregation algorithm with a dictionary provided by this method offers good performance at an attractive computation cost.

4.5.1 Implementation of the dictionary generator

Given a sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{R}^p$, we construct the set of principal components C of the design matrix $\boldsymbol{X}$ by PCA. Then we build the set S of all subspaces spanned by two elements of C:

$$S = \{\text{span}(\boldsymbol{v}_i, \boldsymbol{v}_j),\ (\boldsymbol{v}_i, \boldsymbol{v}_j) \in C.\}. \tag{4.38}$$

On each subspace of S, we perform a clustering to find groups. For each group, we consider the points assigned to it in the original space and recover the empirical mean, the sample variance and construct a normal density with these parameters. A simple implementation would consider all principal components and thus $\frac{p(p-1)}{2}$ subspaces. On each of these subspace a clustering method such as K-means with an arbitrary large number of clusters K would be applied. The whole complexity would be $\mathcal{O}(p^2 n^{2K+1})$. To reduce

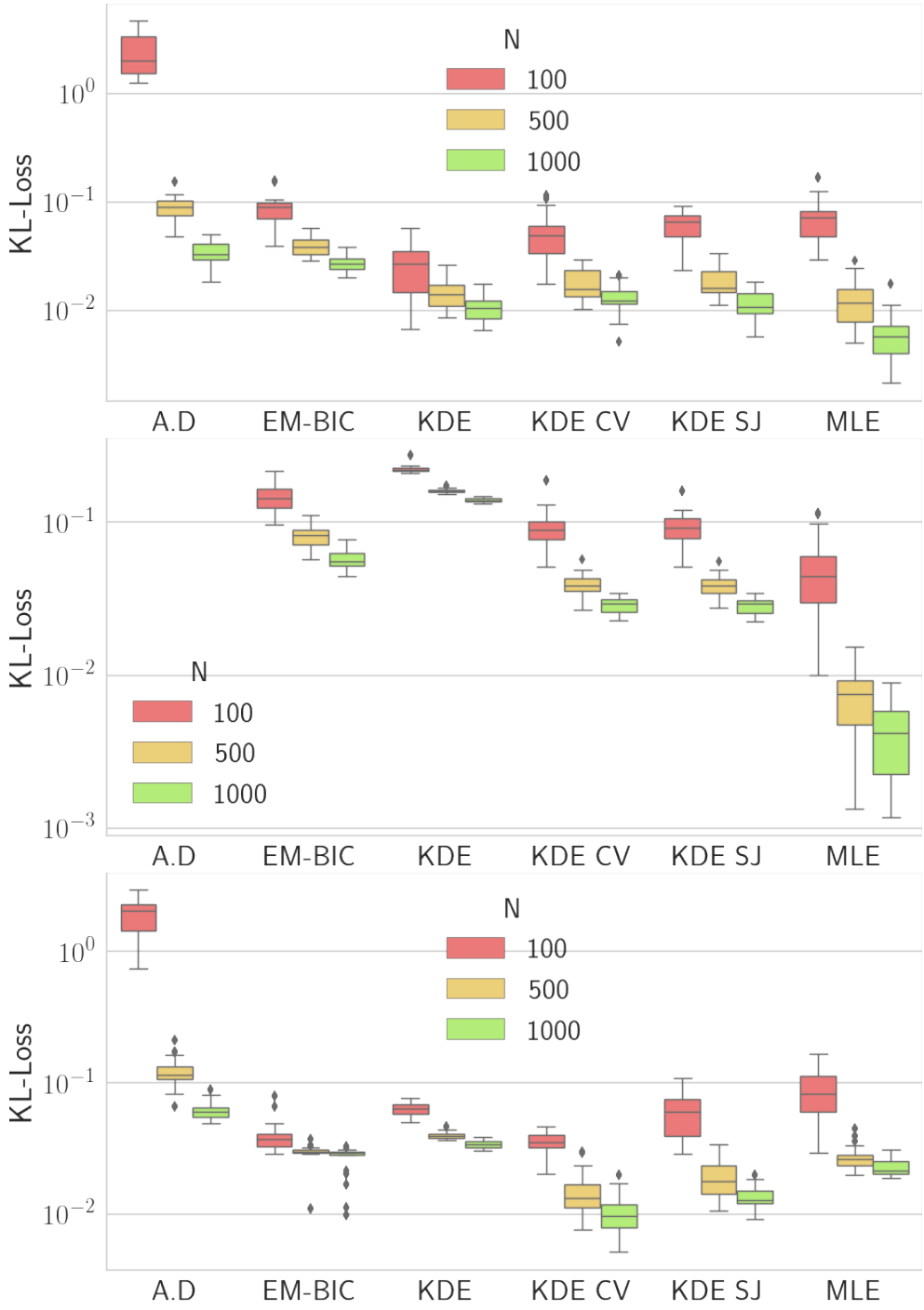

Figure 4.18: Results with f_{unif} (upper panel), f_{rect} (middle panel) and f_{ext} (lower panel) in KL loss with D_{GLU}.

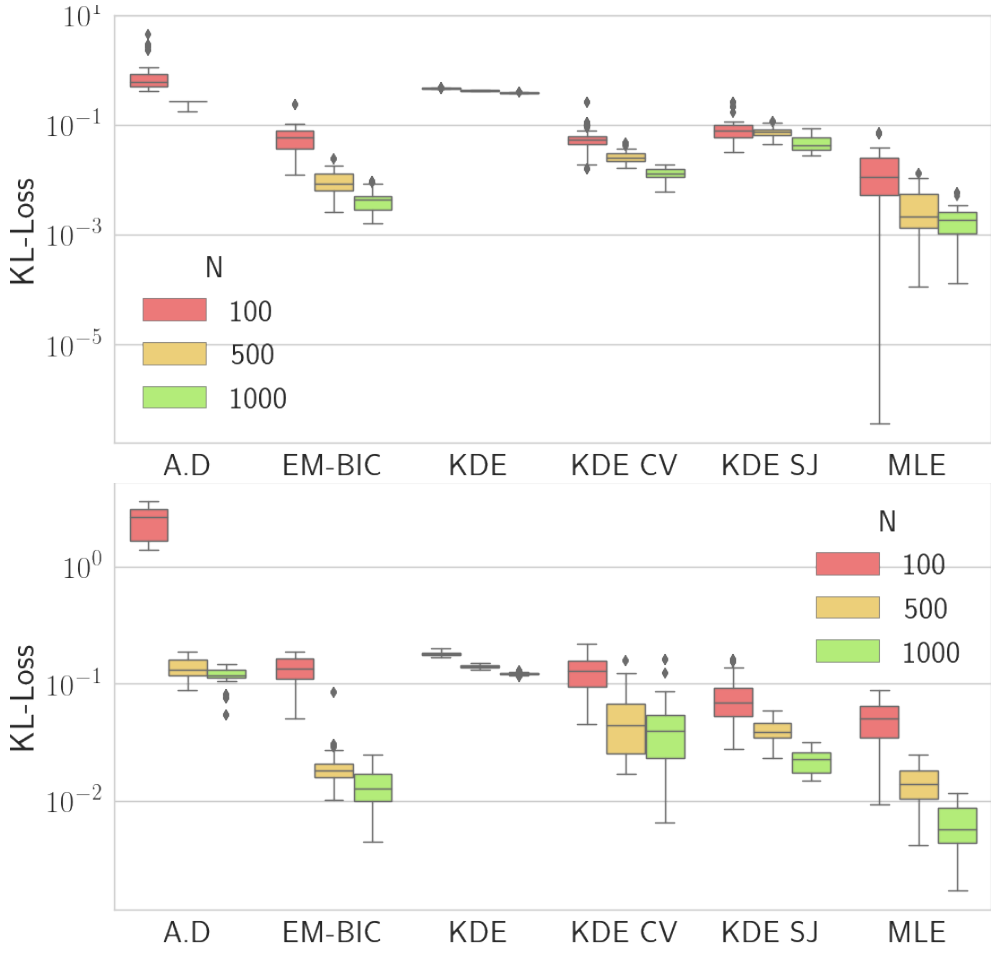

Figure 4.19: Results with f_{gauss} (upper panel) and $f_{\text{gauss-lapl}}$ (lower panel) in KL loss with D_{GLU}.

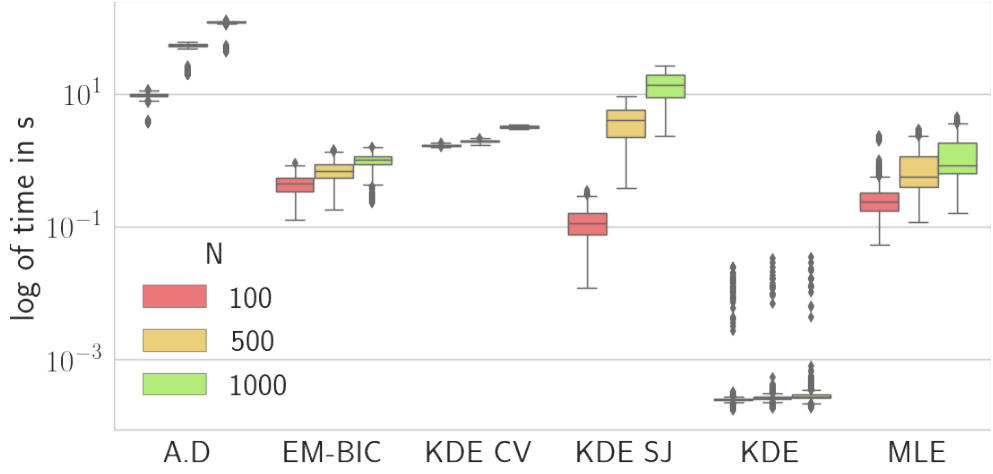

Figure 4.20: Computation times

the computational complexity of this procedure, especially in high dimension, we adopted three strategies:

1. Select the most informative components obtained via the PCA. One can use different techniques such as the Truncated SVD or the method proposed in [Gavish and Donoho, 2014] which circumvent the issue of not knowing rank($\boldsymbol{X}$). They considered the recovery of low-rank matrices from noisy data by hard thresholding of singular values by studying the asymptotic MSE. The AMSE-optimal choice of hard threshold would be for a n-by-p matrix with $n \neq p$, $\widehat{\tau}_* = \omega(\beta).y_{med}$, with $\beta = n/p$, y_{med} is the median singular value of $\boldsymbol{X}$ and $\omega(\beta)$ is described in [Gavish and Donoho, 2014]. An approximation of $\omega(\beta)$ is $\omega(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43$.

2. Perform a model selection for each clustering which reduces the number of densities added to the dictionary. The method chosen is EM with BIC.

3. We address the problem of density duplicates in the dictionary originating from the same subset of points. We saw in the previous section that overlapping densities can degrade the performance of our estimators. One would like to remove these similar densities by performing a two-sample test. The reduction of multivariate two-sample testing to a binary classification problem follows from Friedman in [Friedman, 2003]. To test whether two densities P and Q are equal, we draw two samples $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ and $\{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ from P and Q respectively and construct the dataset

$$\mathcal{D} = \{(\boldsymbol{u}_i, l_i)\}_{i=1}^{n+m} \coloneqq \{(\mathbf{y}_i, -1)\}_{i=1}^{n} \cup \{(\mathbf{z}_i, 0)\}_{i=1}^{m}. \tag{4.39}$$

   We shuffle $\mathcal{D}$ and keep a record of the original assignments for each sample in $\mathcal{D}$. Then, we split this dataset into two parts, $\mathcal{D}_{tr}$ for training a binary classifier and $\mathcal{D}_{te}$ for predicting the classification scores $\{s_i\}_{i=1}^{n+m}$. We consider the two sets S_+ and S_-, the first one contains the scores of the samples originating from $\{\mathbf{z}_i\}_{i=1}^{m}$ and S_- contains the scores of the samples originating from $\{\mathbf{y}_i\}_{i=1}^{n}$. We can view S_+ and S_- as two samples drawn from two probability distributions, $p_+(s)$ and $p_-(s)$, and apply a goodness-of-fit test such as the univariate Kolmogorov–Smirnov test, for testing the equality of these two densities. The resulting test statistic is the statistic for the multivariate two-sample test for the equality of the distributions P and Q.

The dictionary construction procedure is given in Figure 4.21

4.5.2 Experimental evaluation

We created a mixture of 6 components in dimension 5, see Figure 4.22, which mimics data that can be seen in real use cases. The simulation were run in dimension 3, 4 and 5 by selecting the corresponding first axis. We generated $N \in \{200, 500, 1000, 5000\}$ points and ran 200 simulations for each scenario (N, dimension). We used the dictionary generation procedure for the KL-aggregation algorithm with $K_{max} = 10$ on each subspaces and significance level $\alpha = 0.05$. The dataset has been split into two equal parts, one for the dictionary generation algorithm and the other for the KL-aggregation algorithm. We compared the L_2-loss and KL-loss of our method to EM-BIC ($K_{max} = 20$) and KDE-CV (The bandwidth h is selected via cross-validation in $[0.01, \dots, 1]$ in an equal partition of 20 elements). The computation times were also recorded.

Results without the selection of principal components and the goodness-of-fit test for the dictionary generator algorithm.

We compared, first, the KL-algorithm with the dictionary generated by our procedure to KDE-CV and EM-BIC without the two computation optimization techniques discussed before (selection of principal components and the deletion of similar densities). The time given for MLE is the total computational time of the generation of the dictionary and the aggregation algorithm. In the three scenarios (dimension 3,4 and 5), our algorithm presents same performance as EM-BIC in L_2 and KL loss with a better result when $N = 5000$ (see Figures 4.23 to 4.25). Both methods outperforms KDE-CV in all scenarios. This indicates that the set of bandwidths explored for KDE-CV does not fit the data correctly. Increasing the size of this set would increase dramatically the computation times of KDE-CV. Despite the quadratic increase of the size of the dictionary with the dimension, our algorithm takes less time to compute than KDE-CV and slightly more than EM-BIC.

Results with the selection of principal components and the goodness-of-fit test for the dictionary generator algorithm.

Adding the two computation optimization techniques, our algorithm still performs better than KDE-CV and has similar performance than EM-BIC in L_2-loss (see Figures 4.26 to 4.28). Unfortunately our method shows a bigger error variance for the KL-loss, especially when $N = 5000$. This behavior is not expected and may be due to incorrect settings and subtleties in the implementation. Despite adding more "intelligence" in the construction of the dictionary, this procedures counterbalance the cost of adding too much densities

to the KL-aggregation algorithm and therefore leads to smaller computational times independent of the size of the sample. Note that we implemented our methods in Python without Just-In-Time compilations and therefore suffers significant computation overhead compared to Numpy's implementation of EM-BIC and KDE-CV. We are confident that a proper optimized implementation would be significantly faster. This remark highlights the attractiveness of our methods when the size of the sample increases.

4.5.3 Concluding remarks

To conclude, the density dictionary generation method we developed is well suited for our KL-aggregation algorithm. Without the techniques that we implemented to lighten the density dictionary, our methods performs as well as EM-BIC in KL-loss and L_2-loss and slightly better with a large sample ($N = 5000$). With the selection of principal components and the tests of similarity of densities in the dictionary, we tried to solve the problem of computational complexity of our method as the dimension and the size of the sample increase. On this setting, our method shows computation times independent of the size of the sample. Unfortunately, our algorithm shows a large error variance when $N = 5000$ in KL-loss. We are confident that a fine tuning of the parameters of the selection of principal components method and of the tests of similarity of densities would solve this problem. Moreover, we observed in our simulations that the use of the selection of principal components technique and the tests of density similarities to lighten the density dictionary gives us an estimation of the number of real clusters in the data and can be seen as a parameter-free clustering method. From this perspective, our method can be related to a subspace clustering method.

Input: $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ with $\boldsymbol{X}_i \in \mathbb{R}^p$. And K_{max}, maximum number of clusters for EM-BIC, significance level α.

Output: A dictionary of densities $D = \{f_1, \ldots, f_M\}$.

1: Construct the set Ω of singular values of the design matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ which are greater than $\omega(\beta).y_{med}$ with y_{med} median of singular values, $\beta = p/n$ and $\omega(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43$.

2: Construct the set of principal components $\bar{C}$ corresponding to the singular values in Ω.

for $\boldsymbol{v}_i, \boldsymbol{v}_j \in \bar{C}$ **do**

3: Run EM-BIC with maximum K_{max} clusters on the data projected to $\text{span}(\boldsymbol{v}_i, \boldsymbol{v}_j)$, $\boldsymbol{X}_1^{(i,j)}, \ldots, \boldsymbol{X}_n^{(i,j)}$, and construct clusters of points $G_1, \ldots, G_K$.

4: For each cluster G_m, $m \in [K]$, compute the mean $\widehat{\boldsymbol{\mu}}_m$ and variance $\widehat{\boldsymbol{\Sigma}}_m$ of the points assigned to G_m in the original space $\mathbb{R}^P$.

5: Add to the dictionary D the Gaussian densities $\{\varphi_{(\widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\sigma}}_m)}\}_{m \in [K]}$.

end for.

for $\widehat{f}_i, \widehat{f}_j \in D$ **do**

6: Draw two samples $\{\mathbf{y}_1, \ldots, \mathbf{y}_l\}, \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ from $\boldsymbol{Y} \sim \widehat{f}_i$ and $\boldsymbol{Z} \sim \widehat{f}_j$ and construct the dataset $\mathcal{D} = \{(\boldsymbol{u}_i, l_i)\}_{i=1}^{n+m} \coloneqq \{(\mathbf{y}_i, -1)\}_{i=1}^{n} \cup \{(\mathbf{z}_i, 0)\}_{i=1}^{m}$.

7: Shuffle and split $\mathcal{D}$ into $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$.

8: Train a binary classifier on $\mathcal{D}_{tr}$ and get the classification scores $\{s_i\}$ on $\mathcal{D}_{te}$.

9: Separate $\{s_i\}$ into $\{s_i\}^+$, scores of points drawn from $\boldsymbol{Z}$ and $\{s_i\}^-$ for $\boldsymbol{Y}$.

10: Perform a two-samples Kolmogorov-Smirnov test on $\{s_i\}^+$ and $\{s_i\}^-$ and reject H_0 (The two multivariate samples are drawn from the same distribution) with significance level α.

11: **If** H_0 rejected, remove $\widehat{f}_j$ of D, **else**, keep $\widehat{f}_i$ and $\widehat{f}_j$.

end for.

Figure 4.21: Procedure for generating a dictionary of densities

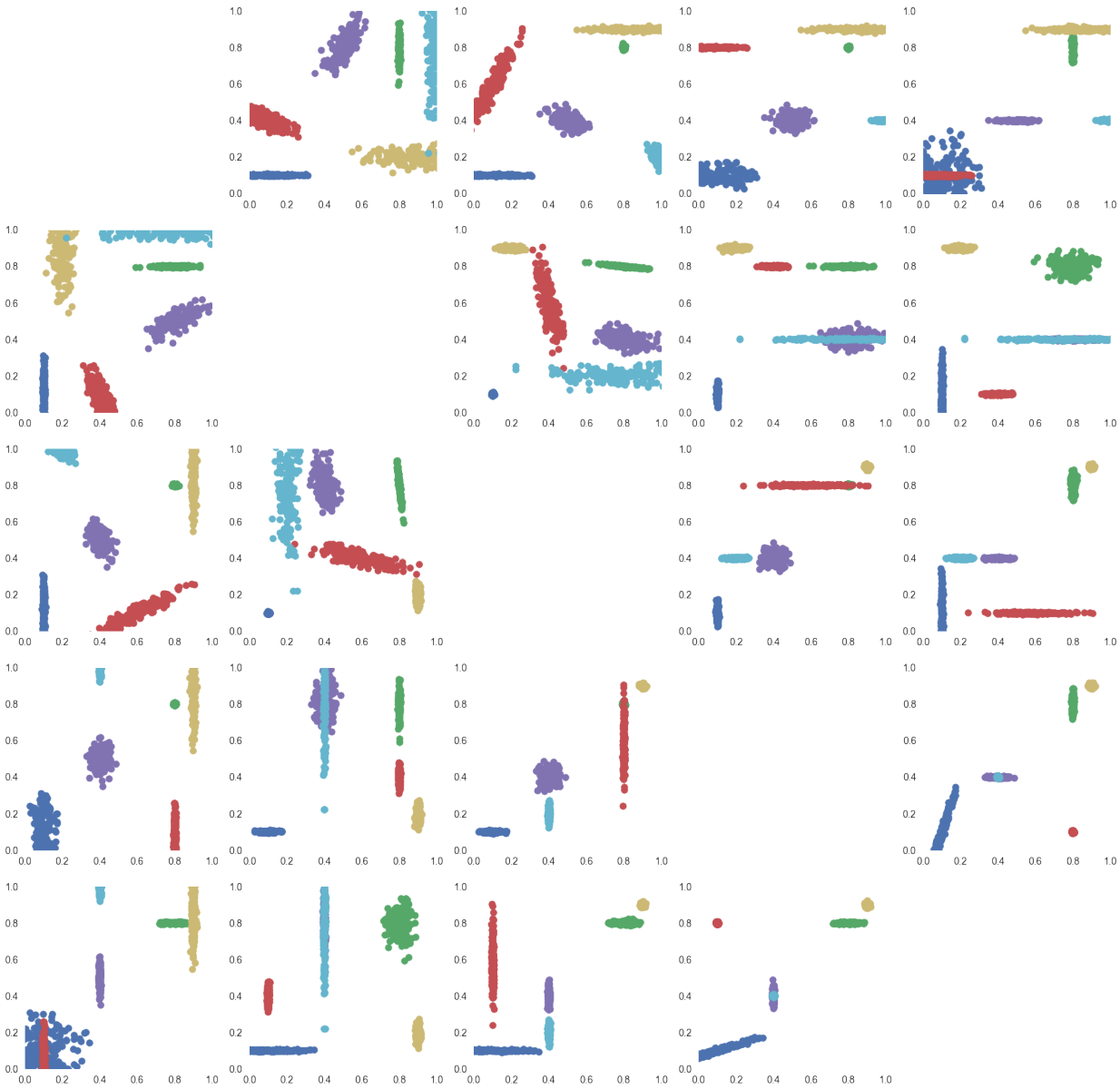

Figure 4.22: Simulated data for the dictionary generator algorithm and KL-aggregation

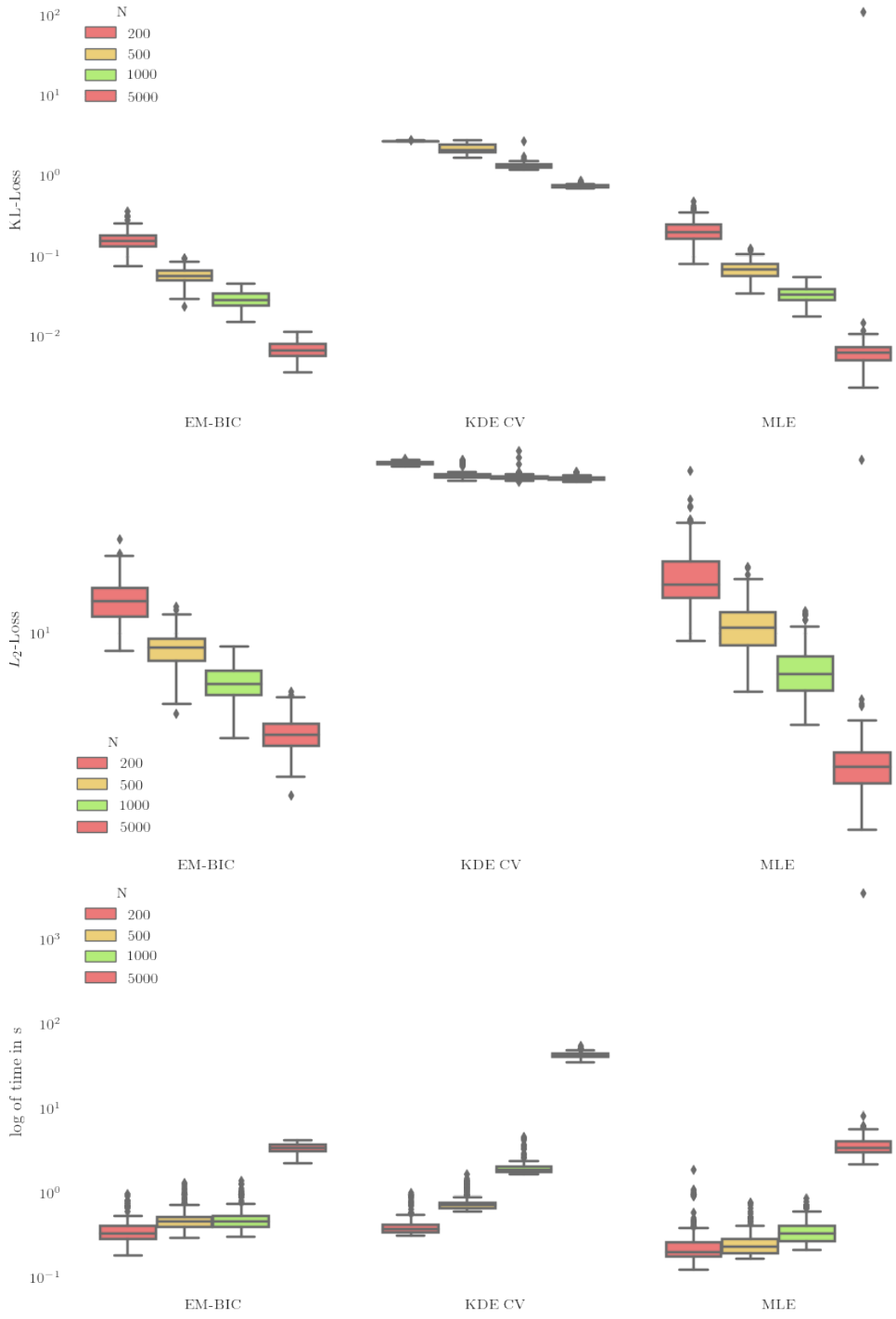

Figure 4.23: Results for dimension 3. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). Without dictionary generation optimizations.

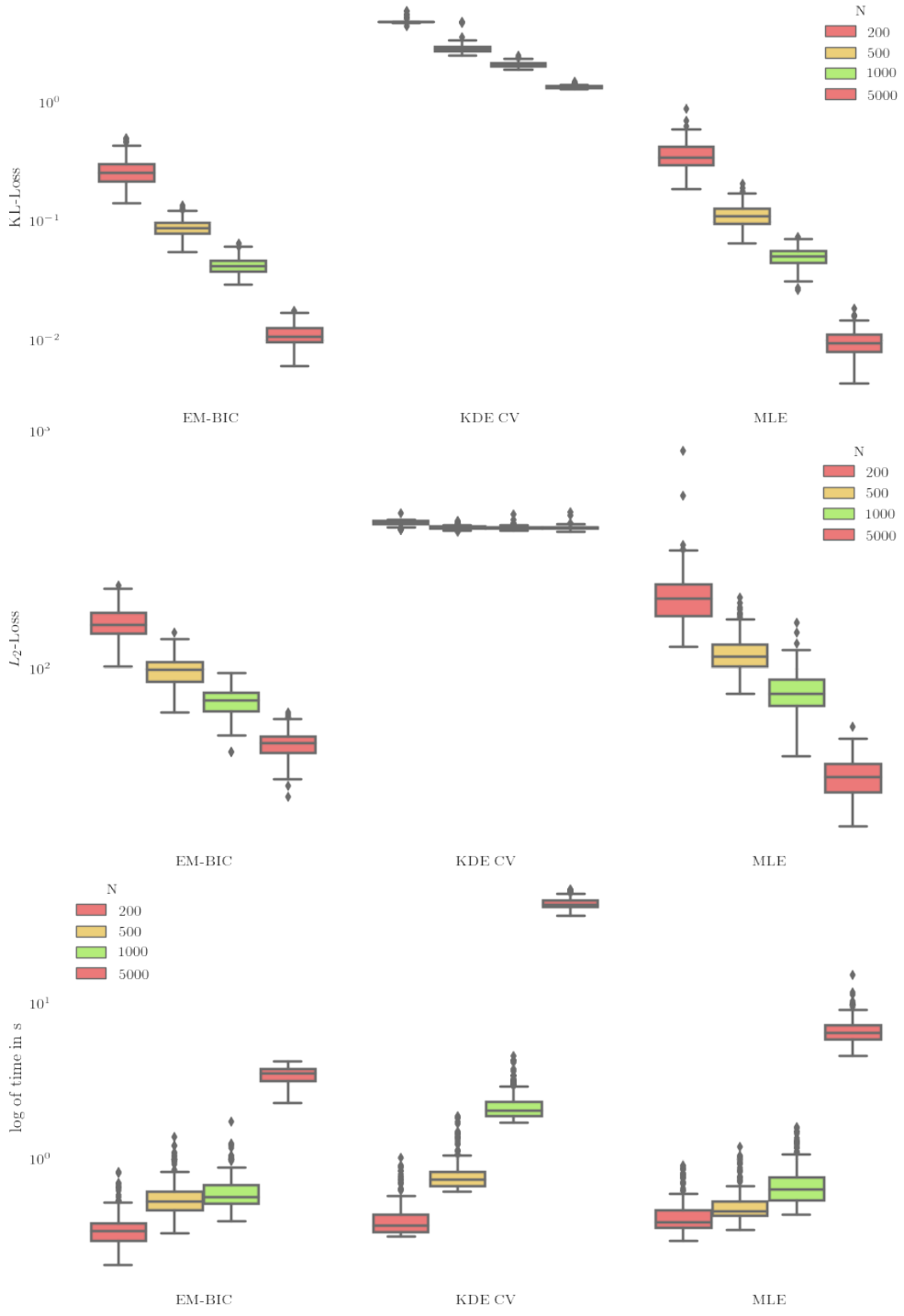

Figure 4.24: Results for dimension 4. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). Without dictionary generation optimizations.

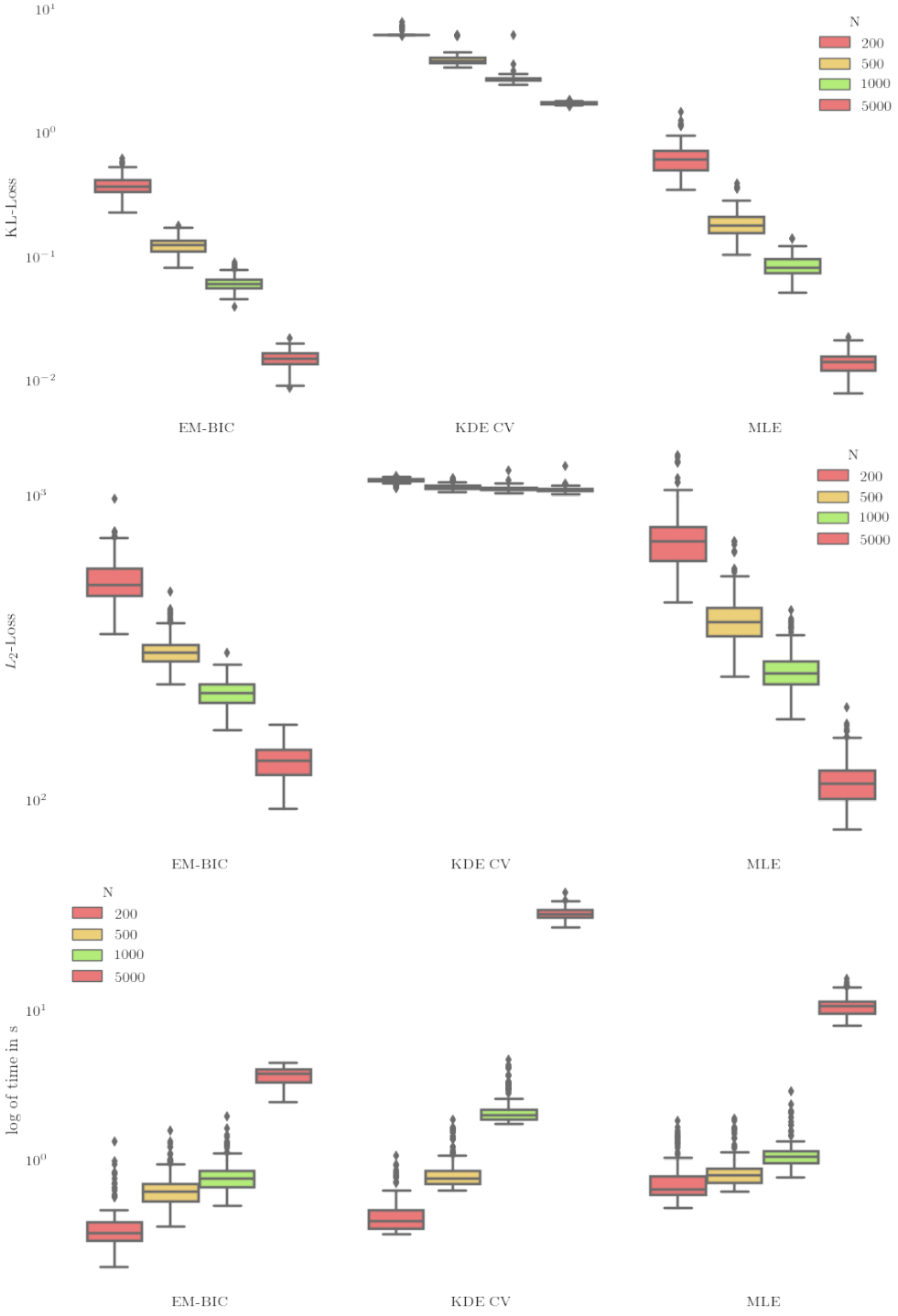

Figure 4.25: Results for dimension 5. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). Without dictionary generation optimizations.

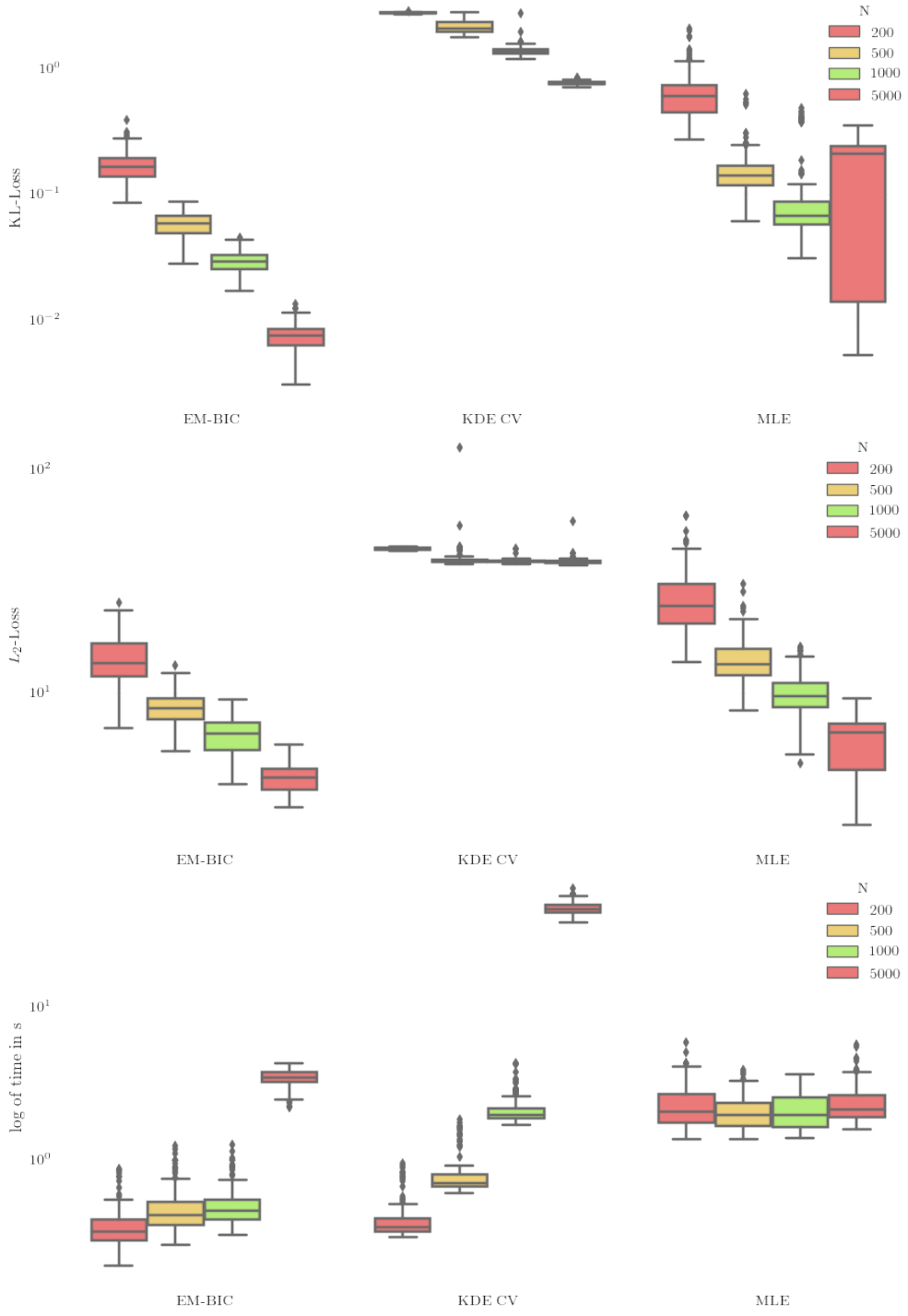

Figure 4.26: Results for dimension 3. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). With selection of principal components and deletion of similar densities.

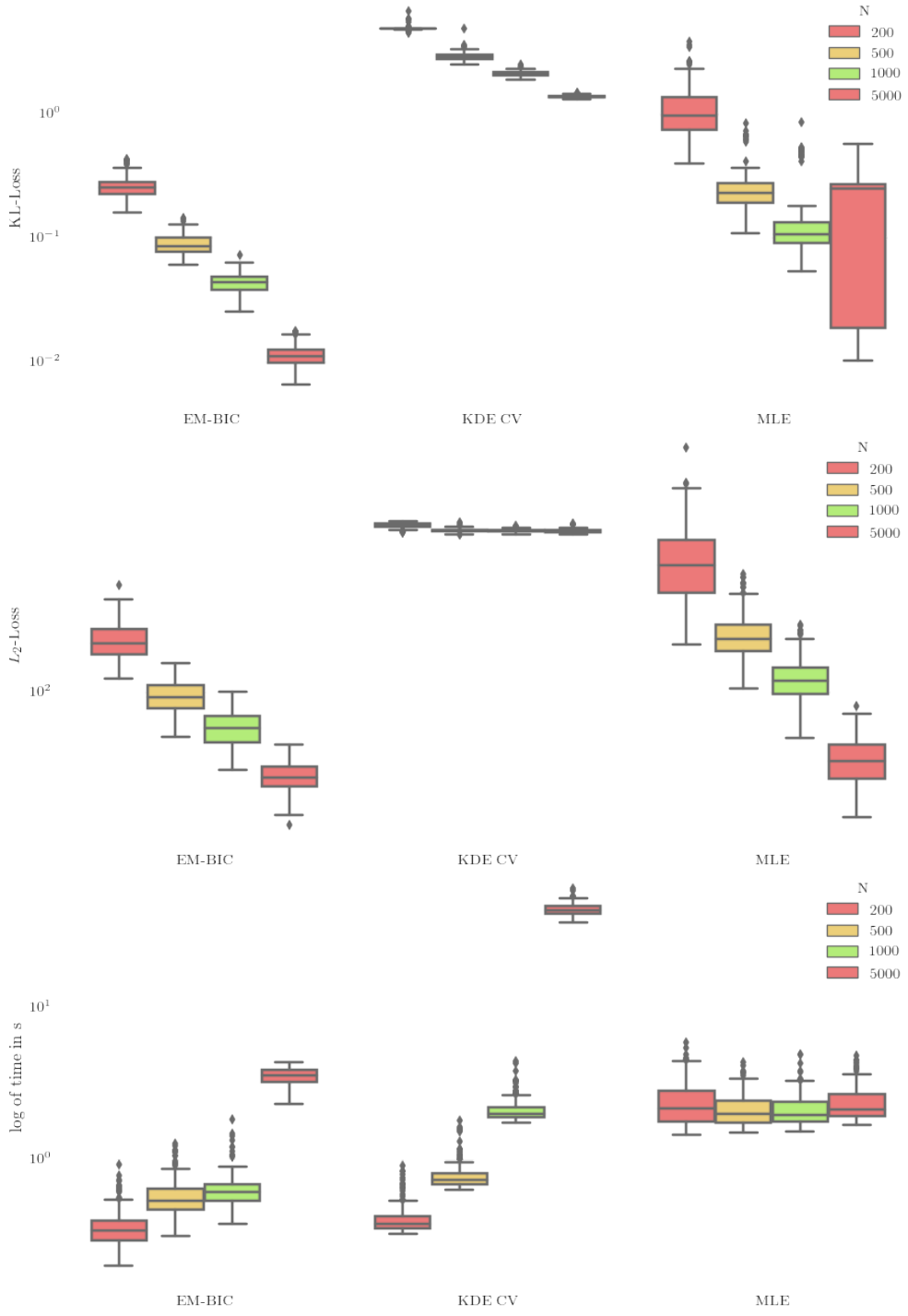

Figure 4.27: Results for dimension 4. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). With selection of principal components and deletion of similar densities.

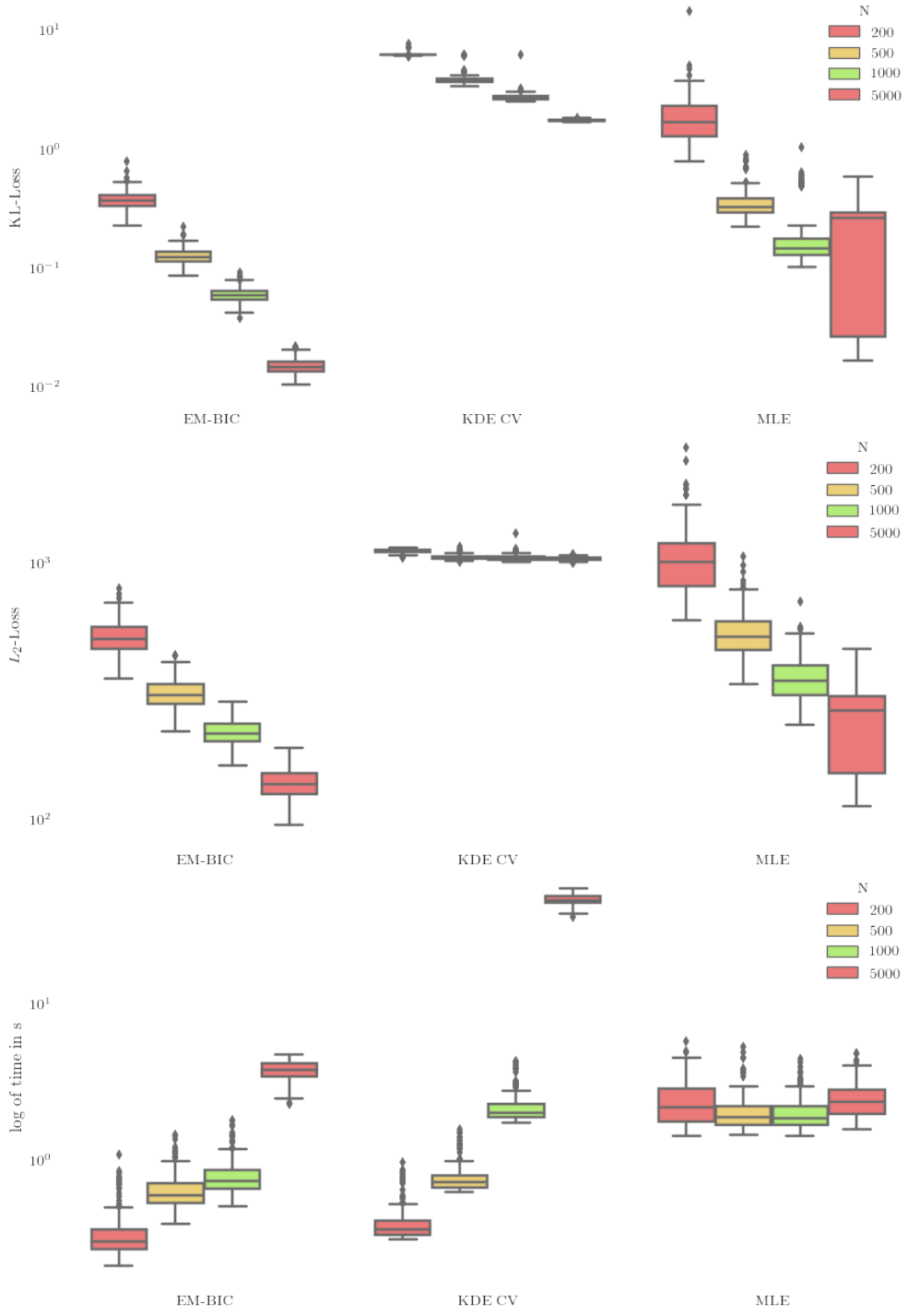

Figure 4.28: Results for dimension 5. KL-Loss (upper panel), L_2-Loss (middle panel) and computation time (lower panel). With selection of principal components and deletion of similar densities.

Bibliography

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 2007.

C. Hennig, M. Meila, F. Murtagh, and R. Rocci. Handbook of Cluster Analysis. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, 2015. ISBN 9781466551886.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pages 281–297, Berkeley, Calif., 1967. University of California Press.

S. Dasgupta. The Hardness of K-means Clustering. Technical report (University of California, San Diego. Department of Computer Science and Engineering). Department of Computer Science and Engineering, University of California, San Diego, 2008.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. Mach. Learn., 75(2):245–248, May 2009. ISSN 0885-6125. doi: 10.1007/s10994-009-5103-0. URL http://dx.doi.org/10.1007/s10994-009-5103-0.

Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In Proceedings of the Tenth Annual Symposium on Computational Geometry, SCG '94, pages 332–339, New York, NY, USA, 1994. ACM. ISBN 0-89791-648-4. doi: 10.1145/177424.178042. URL http://doi.acm.org/10.1145/177424.178042.

S. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2):129–137, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In Proceedings of the Twenty-second Annual Symposium on Computational Geometry, SCG '06, pages 144–153, New York, NY, USA, 2006. ACM. ISBN 1-59593-340-9. doi: 10.1145/1137856.1137880. URL http://doi.acm.org/10.1145/1137856.1137880.

David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. URL http://dl.acm.org/citation.cfm?id=1283383.1283494.

Vladimir Makarenkov and Pierre Legendre. Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. Journal of Classification, 18(2):245–271, 2001.

Joshua Zhexue Huang, Jun Xu, Michael Ng, and Yunming Ye. Weighting Method for Feature Selection in K-Means. Chapman and Hall/CRC, 2007. ISBN 978-1-58488-878-9. doi: doi:10.1201/9781584888796.ch10.

Renato Cordeiro de Amorim and Boris Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. Pattern Recogn., 45(3):1061–1075, March 2012. ISSN 0031-3203. doi: 10.1016/j.patcog.2011.08.012. URL http://dx.doi.org/10.1016/j.patcog.2011.08.012.

L. Kaufman and Peter J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley, 1990.

D. Sculley. Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 1177–1178, 2010. ISBN 978-1-60558-799-8.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), pages 165–176, Oct 2006. doi: 10.1109/FOCS.2006.75.

A. Guénoche, P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. Journal of Classification, 8(1):5–30, Jan 1991. ISSN 1432-1343. doi: 10.1007/BF02616245. URL https://doi.org/10.1007/BF02616245.

R. L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. IEEE Ann. Hist. Comput., 7(1):43–57, January 1985. ISSN 1058-6180. doi: 10.1109/MAHC.1985.10011. URL https://doi.org/10.1109/MAHC.1985.10011.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1):86–97, 2012. ISSN 1942-4795. doi: 10.1002/widm.53. URL http://dx.doi.org/10.1002/widm.53.

Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, 58(301):236–244, 1963. doi: 10.1080/01621459.1963.10500845.

G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies1. hierarchical systems. The Computer Journal, 9(4):373–380, 1967. doi: 10.1093/comjnl/9.4.373. URL +http://dx.doi.org/10.1093/comjnl/9.4.373.

Vladimir Batagelj. Generalized ward and related clustering problems. In In H.H. Bock (Ed.), Classification and Related Methods of Data Analysis, pages 67–74. North-Holland, 1988.

F. Murtagh. Multidimensional clustering algorithms. 1985.

M. Jambu. Exploration informatique et statistique des données. Collection technique et scientifique des télécommunications. Dunod, 1989. ISBN 9782040188405.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. IBM J. Res. Dev., 17(5):420–425, September 1973. ISSN 0018-8646. doi: 10.1147/rd.175.0420. URL http://dx.doi.org/10.1147/rd.175.0420.

Miroslav Fiedler. Algebraic connectivity of graphs. Czechoslovak Mathematical Journal, 23(2):298–305, 1973.

Ulrike von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17 (4):395–416, December 2007. ISSN 0960-3174. doi: 10.1007/s11222-007-9033-z. URL http://dx.doi.org/10.1007/s11222-007-9033-z.

Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. Linear Algebra and its Applications, 421(2):284 – 305, 2007. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2006.07.020. Special Issue in honor of Miroslav Fiedler.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888–905, August 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL http://dx.doi.org/10.1109/34.868688.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, pages 849–856. MIT Press, 2001.

L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 11 (9):1074–1085, Sep 1992. ISSN 0278-0070. doi: 10.1109/43.159993.

Dorothea Wagner and Frank Wagner. Between Min Cut and Graph Bisection, pages 744–750. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993. ISBN 978-3-540-47927-7. doi: 10.1007/3-540-57182-5_65. URL https://doi.org/10.1007/3-540-57182-5_65.

Stephen Guattery and Gary L. Miller. On the quality of spectral separators. SIAM Journal on Matrix Analysis and Applications, 19(3):701–719, 1998. doi: 10.1137/S0895479896312262. URL https://doi.org/10.1137/S0895479896312262.

Boaz Nadler and Meirav Galun. Fundamental limitations of spectral clustering. In in Advanced in Neural Information Processing Systems 19, B. Schölkopf and, pages 1017–1024, 2007.

André Hardy. On the number of clusters. Comput. Stat. Data Anal., 23(1):83–96, November 1996. ISSN 0167-9473. doi: 10.1016/S0167-9473(96)00022-9. URL http://dx.doi.org/10.1016/S0167-9473(96)00022-9.

Glenn W. Milligan and Martha C. Cooper. An examination of procedures for determining the number of clusters in a data set. Psychometrika, 50(2):159–179, Jun 1985. ISSN 1860-0980. doi: 10.1007/BF02294245. URL https://doi.org/10.1007/BF02294245.

Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math., 20(1):53–65, November 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL http://dx.doi.org/10.1016/0377-0427(87)90125-7.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society:

Series B (Statistical Methodology), 63(2):411–423, 2001. ISSN 1467-9868. doi: 10.1111/1467-9868.00293. URL http://dx.doi.org/10.1111/1467-9868.00293.

Gideon Schwarz. Estimating the dimension of a model. Ann. Statist., 6(2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL http://dx.doi.org/10.1214/aos/1176344136.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, Second International Symposium on Information Theory, pages 267–281, Budapest, 1973. Akadémiai Kiado.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39, No. 1:1–38, 1977.

Richard Bellman. Dynamic Programming. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957. URL http://books.google.com/books?id=fyVtp3EMxasC&pg=PR5&dq=dynamic+programming+richard+e+bellman&client=firefox-a#v=onepage&q=dynamic%20programming%20richard%20e%20bellman&f=false.

Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining, 5 (5):363–387, 2012. ISSN 1932-1872. doi: 10.1002/sam.11161. URL http://dx.doi.org/10.1002/sam.11161.

Charles Bouveyron and Camille Brunet. Model-Based Clustering of High-Dimensional Data: A review. Computational Statistics and Data Analysis, 71:52–78, 2013. doi: 10.1016/j.csda.2012.12.008. URL https://hal.archives-ouvertes.fr/hal-00750909.

Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: A review. SIGKDD Explor. Newsl., 6(1):90–105, June 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007731. URL http://doi.acm.org/10.1145/1007730.1007731.

C. Giraud. Introduction to High-Dimensional Statistics. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2014. ISBN 9781482237948.

P. Bühlmann and S. van de Geer. Statistics for High-Dimensional Data: Methods, Theory and Applications. Springer Series in Statistics. Springer Berlin Heidelberg, 2011. ISBN 9783642201929. URL https://books.google.fr/books?id=S6jYXmh988UC.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. Journal of Machine Learning Research, 2008.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. Biometrika, 94(1):19, 2007. doi: 10.1093/biomet/asm018. URL +http://dx.doi.org/10.1093/biomet/asm018.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. Ann. Statist., 34(3):1436–1462, 06 2006. doi: 10.1214/009053606000000281. URL http://dx.doi.org/10.1214/009053606000000281.

D. Edwards. Introduction to Graphical Modelling. Springer Texts in Statistics. Springer New York, 2000. ISBN 9780387950549.

A. P. Dempster. Covariance selection. Biometrics, 28(1):157–175, 1972. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2528966.

Leo Breiman. Heuristics of instability and stabilization in model selection. Ann. Statist., 24(6):2350–2383, 12 1996. doi: 10.1214/aos/1032181158. URL http://dx.doi.org/10.1214/aos/1032181158.

R. Mazumder. Topics in sparse multivariate statistics (thesis). 2012.

Neal Parikh and Stephen Boyd. Proximal algorithms. Found. Trends Optim., 1(3):127–239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. URL http://dx.doi.org/10.1561/2400000003.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci., 2(1):183–202, March 2009. ISSN 1936-4954. doi: 10.1137/080716542. URL http://dx.doi.org/10.1137/080716542.

Alexandre B. Tsybakov. Aggregation and minimax optimality in high-dimensional estimation. In Proceedings of the International Congress of Mathematicians (Seoul, August 2014), volume 3, pages 225–246, 2014.

O Catoni. The mixture approach to universal model selection. Technical report, 1997. URL http://cds.cern.ch/record/461892.

Yuhong Yang. Mixing strategies for density estimation. Ann. Statist., 28(1):75–87, 2000.

A. Juditsky, P. Rigollet, and A. B. Tsybakov. Learning by mirror averaging. Ann. Statist., 36(5):2183–2206, 2008.

A. B. Yuditskiĭ, A. V. Nazin, A. B. Tsybakov, and N. Vayatis. Recursive aggregation of estimators by the mirror descent method with averaging. Problemy Peredachi Informatsii, 41(4):78–96, 2005.

Arnak S. Dalalyan and Alexandre B. Tsybakov. Mirror averaging with sparsity priors. Bernoulli, 18(3):914–944, 2012.

P. C. Bellec. Optimal exponential bounds for aggregation of density estimators. Technical report, arXiv:1405.3907, May 2014.

C. Butucea, J.-F. Delmas, A. Dutfoy, and R. Fischer. Optimal exponential bounds for aggregation of estimators for the Kullback-Leibler loss. Technical report, arXiv:1601.05686, January 2016.

Dong Dai, Philippe Rigollet, and Tong Zhang. Deviation optimal learning using greedy Q-aggregation. Ann. Statist., 40(3):1878–1905, 2012.

Philippe Rigollet. Kullback-Leibler aggregation and misspecified generalized linear models. Ann. Statist., 40(2):639–665, 2012.

Ph. Rigollet and A. B. Tsybakov. Linear and convex aggregation of density estimators. Math. Methods Statist., 16(3):260–280, 2007.

K. Lounici. Generalized mirror averaging and D-convex aggregation. Math. Methods Statist., 16(3):246–259, 2007.

Florentina Bunea, Alexandre B. Tsybakov, and Marten H. Wegkamp. Aggregation for gaussian regression. Ann. Statist., 35(4):1674–1697, 08 2007.

Florentina Bunea, Alexandre B. Tsybakov, Marten H. Wegkamp, and Adrian Barbu. Spades and mixture models. Ann. Statist., 38(4):2525–2558, 2010.

K. Bertin, E. Le Pennec, and V. Rivoirard. Adaptive Dantzig density estimation. Ann. Inst. Henri Poincaré Probab. Stat., 47(1):43–74, 2011.

Jonathan Q. Li and Andrew R. Barron. Mixture density estimation. In Advances in Neural Information Processing Systems 12, pages 279–285, 1999.

Jonathan Q. Li. Estimation of Mixture Models. Phd thesis, Yale University, 1999.

Alexander Rakhlin, Dmitry Panchenko, and Sayan Mukherjee. Risk bounds for mixture density estimation. ESAIM Probab. Stat., 9:220–229, 2005.

Anatoli Juditsky and Arkadii Nemirovski. Functional aggregation for nonparametric regression. The Annals of Statistics, 28(3):681–712, 2000.

Alexandre B. Tsybakov. Optimal rates of aggregation. In Computational Learning Theory and Kernel Machines, COLT/Kernel, Proceedings, pages 303–313, 2003.

Guillaume Lecué. Lower bounds and aggregation in density estimation. J. Mach. Learn. Res., 7:971–981, 2006.

Dong Xia and Vladimir Koltchinskii. Estimation of low rank density matrices: Bounds in Schatten norms and other distances. Electron. J. Stat., 10(2):2717–2745, 2016.

Sara van de Geer and Peter Bühlmann. On the conditions used to prove oracle results for the Lasso. Electron. J. Stat., 3:1360–1392, 2009.

Guillaume Lecué and Shahar Mendelson. On the optimality of the empirical risk minimization procedure for the convex aggregation problem. Ann. Inst. Henri Poincaré Probab. Stat., 49(1):288–306, 2013.

Guillaume Lecué. Empirical risk minimization is optimal for the convex aggregation problem. Bernoulli, 19(5B):2153–2166, 2013.

Philippe Rigollet and Alexandre Tsybakov. Exponential screening and optimal rates of sparse estimation. Ann. Statist., 39(2):731–771, 2011.

Philippe Rigollet. Oracle inequalities, aggregation and adaptation. Phd thesis, Université Pierre et Marie Curie - Paris VI, November 2006.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. IEEE Trans. Inform. Theory, 57(10): 6976–6994, 2011.

Zhan Wang, Sandra Paterlini, Fuchang Gao, and Yuhong Yang. Adaptive minimax regression estimation over sparse ℓ_q-hulls. J. Mach. Learn. Res., 15:1675–1711, 2014.

Pierre C. Bellec, Arnak S. Dalalyan, Edwin Grappin, and Quentin Paris. On the prediction loss of the lasso in the partially labeled setting. Technical report, arXiv:1606.06179, June 2016.

S. Boucheron, G. Lugosi, and P. Massart. Concentration Inequalities: A Nonasymptotic Theory of Independence. OUP Oxford, 2013. ISBN 9780199535255.

Vladimir Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems, volume 2033 of Lecture Notes in Mathematics. Springer, 2011. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008.

Michel Ledoux and Michel Talagrand. Probability in Banach Spaces: isoperimetry and processes. Springer, Berlin, 1991.

A.B. Tsybakov. Introduction to Nonparametric Estimation. Springer Series in Statistics. Springer, 2009. ISBN 9780387790510.

Yu. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007. URL http://EconPapers.repec.org/RePEc:cor:louvco:2007076.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate O(1/sqr(k)). Soviet Mathematics Doklady, 27:372–376, 1983. URL http://www.core.ucl.ac.be/~{}nesterov/Research/Papers/DAN83.pdf.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 272–279, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390191. URL http://doi.acm.org/10.1145/1390156.1390191.

Weiran Wang, Miguel Á. Carreira-perpiñán, and Weprovideanelementaryproofofasimple Efficientalgorithmforcomputingtheeuclideanprojection. Projection onto the probability simplex: An, 2013.

Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. Ann. Statist., 35(6):2313–2351, 12 2007. doi: 10.1214/009053606000001523. URL http://dx.doi.org/10.1214/009053606000001523.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. Ann. Statist., 37(4):1705–1732, 08 2009. doi: 10.1214/08-AOS620. URL http://dx.doi.org/10.1214/08-AOS620.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/.

S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. Journal of the Royal Statistical Society, Series B: Methodological, 53:683–690, 1991.

David W. Scott. Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons, Inc, second edition edition, 2015.

Peter Hall and J. S. Marron. Estimation of integrated squared density derivatives. Statistics & Probability Letters, 6(2):109–115, 1987. URL http://EconPapers.repec.org/RePEc:eee:stapro:v:6:y:1987:i:2:p:109-115.

M. C. Jones and S. J. Sheather. Using non-stochastic terms to advantage in kernel-based estimation of integrated squared density derivatives. Statistics & Probability Letters, 11 (6):511–514, 1991. URL http://EconPapers.repec.org/RePEc:eee:stapro:v:11:y:1991:i:6:p:511-514.

Charles J. Stone. Optimal rates of convergence for nonparametric estimators. Ann. Statist., 8(6):1348–1360, 11 1980. doi: 10.1214/aos/1176345206. URL http://dx.doi.org/10.1214/aos/1176345206.

Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15, pages 7:1–7:6, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-4005-2. doi: 10.1145/2833157.2833162. URL http://doi.acm.org/10.1145/2833157.2833162.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. IEEE Transactions on Information Theory, 60(8):5040–5053, Aug 2014. ISSN 0018-9448. doi: 10.1109/TIT.2014.2323359.

Jerome H. Friedman. On multivariate goodness of fit and two sample testing. eConf, C030908, 2003.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Sur l'apprentissage non supervisé en haute dimension

Mots Clefs : clustering, agrégation, haute dimension, estimation de densité, mélanges.

Résumé : Deux sujet sont traités dans cette thèse: le clustering en haute dimension et l'estimation de densités de mélange. L'estimation des paramètres d'une loi mélange est un problème difficile en haute dimension. Trois méthodes sont présentées pour résoudre ce problème: la première est une estimation des matrices de covariances avec hypothèse de parcimonie, les deux autres visent à estimer le nombre de composantes du mélange. La deuxième partie étudie l'estimateur du maximum de vraisemblance d'une densité sous l'hypothèse qu'elle est bien approximée par un mélange de plusieurs densités données. Nous réalisons une étude statistique des performances de l'estimateur par rapport à la perte de Kullback-Leibler et établissons des bornes de risque sous la forme d'inégalités d'oracle exacte. Nous introduisons la notion d'agrégation (presque)-D-parcimonieuse et des bornes inférieures sont établies. Enfin, nous proposons un algorithme qui réalise l'agrégation en Kullback-Leibler de composantes d'un dictionnaire. Nous comparons sa performance avec différentes méthodes. Nous proposons ensuite une méthode pour construire le dictionnaire de densités et l'étudions de manière numérique.

Title : On unsupervised learning in high dimension

Keys words : clustering, aggregation, high dimension, density estimation, mixtures.

Abstract : Two subjects are treated in this thesis: high-dimensional clustering and estimation of mixture densities. The estimation of the parameters of a mixture law is a difficult problem in high dimension. Three methods are presented to solve this problem: the first is an estimation of covariance matrices with sparsity hypothesis, the other two are aimed at estimating the number of components of the mixture. The second part studies the maximum likelihood estimator of a density under the assumption that it is well approximated by a mixture of several given densities. We perform a statistical study of the performance of the estimator with respect to the loss of Kullback-Leibler and establish risk bounds in the form of exact oracle inequalities. We introduce the concept of (nearly)-D-sparse aggregation and lower bounds are established. Finally, we propose an algorithm that performs Kullback-Leibler aggregation of components of a dictionary. We compare its performance with different methods. We then propose a method to build the dictionary of densities and study it experimentally.