Double sparsity in high-dimensional Gaussian mixture estimation and clustering

Subject Overview

Supervisor: A.S. Dalalyan
PHd Student: M. Sebbar

February 14, 2016

Contents

1 Notation

For any vector $\boldsymbol{\mu} \in \mathbb{R}^p$ and $p \times p$ matrix $\boldsymbol{\Sigma}$ we denote by $\varphi_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ the probability density function of the Gaussian distribution $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^\top$ while $|\mathbf{A}|$ stands for its determinant if $\mathbf{A}$ is a square matrix. The set of $p \times p$ positive semidefinite matrices is denoted $\mathcal{S}_+^p$, and the set of positive definite matrices is denoted by $\mathcal{S}_{++}^p$. We write $\mathbf{1}_p$ for the p-vector $(1, \dots, 1)^\top$. For any integer $K > 0$, we define $[K]$ as the set $\{1, \dots, K\}$.

2 Introduction

The broad goal of this thesis is to tackle a clustering problem in the scope of mixtures model framework. More precisely, we will study the clustering of points drawn from high-dimensional Gaussian mixtures distributions.
Thus, in the first part of this section we present the Gaussian mixture model and the second part we describe the well know Expectation-Maximization algorithm (EM). We will also present the limitations of this algorithm in high-dimensional setting.

2.1 The Gaussian mixture model

The Gaussian mixture model is an important framework for clustering problems. It assumes that the observations are drawn from a mixture distribution the components of which are Gaussian with parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$:

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

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 $\mathbf{1}_p^\top \boldsymbol{\pi} = 1$. The density of one observation $\boldsymbol{X}_1$ is then given by:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k}(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^p, \tag{2}$$

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 X:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathbf{P}(Z=k) p_\theta(\boldsymbol{x}|Z=k) = \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathbb{R}^p. \tag{3}$$

In the clustering problem, the goal is to assign 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

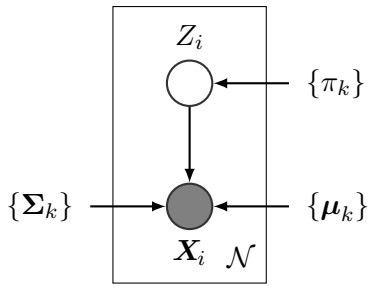
$$\tau_k(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|Z=k)\mathbf{P}(Z=k)}{p_{\boldsymbol{\theta}}(\boldsymbol{x})} = \frac{\pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x})}{\sum_{k'=1}^{K} \pi_{k'} \varphi_{\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}}(\boldsymbol{x})} \tag{4}$$

Since the true value of the parameter $\boldsymbol{\theta}$ is not available, formula (4) 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

$$\widehat{g}(\boldsymbol{x}) = g^*_{\widehat{\boldsymbol{\theta}}}(\boldsymbol{x}) = \arg\max_{k \in [K]} \tau_k(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}) = \arg\max_{k \in [K]} \widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k}(\boldsymbol{x}). \tag{5}$$

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 (2). The following graphical model depicts the scheme of the observations:



The log-likelihood of the Gaussian mixture model is

$$\ell_n(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}(\boldsymbol{x}_i) \right\}. \tag{6}$$

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 (6) 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:

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:`

$$\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ell(\boldsymbol{\theta} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \arg\max_{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}} \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) \right\}. \quad (7)$$

2: `Output the clustering rule:`

$$\widehat{g}(\cdot) = \arg\max_{k \in [K]} \widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k}(\cdot). \quad (8)$$

Figure 1: Clustering under Gaussian mixture modeling

2.2 EM Algorithm

The goal of the EM algorithm is to approximate a solution of the problem (7). 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 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:

$$\log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) \right\} = \max_{\substack{\boldsymbol{\tau} \in [0,1]^K \\ \boldsymbol{\tau}^\top \mathbf{1}_K = 1}} \sum_{k=1}^{K} \left\{ \tau_k \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_k \log(\pi_k / \tau_k) \right\}. \quad (9)$$

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 (7) and (9), we get

$$\widehat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \max_{\boldsymbol{\mathcal{T}}} \sum_{i=1}^{N} \sum_{k=1}^{K} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\}. \quad (10)$$

The great advantage of this new representation of the log-likelihood function is that the cost function in (10), 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 (10) is that the two optimization problems involved in the alternating maximization procedure admit explicit solutions.

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 2: EM algorithm for Gaussian mixtures

Lemma 1. *Let us introduce the cost function*

$$F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) = \sum_{i=1}^n \sum_{k=1}^K \Big\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \Big\}. \tag{11}$$

Then, the following two optimization problems

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) \in \arg\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}), \qquad \widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\mathcal{T}}} F(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) \tag{12}$$

has explicit solutions given by

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^n \tau_{i,k}, \qquad \widehat{\boldsymbol{\mu}}_k = \frac{1}{n\widehat{\pi}_k}\sum_{i=1}^n \tau_{i,k}\boldsymbol{x}_i, \qquad \forall k \in [K], \tag{13}$$

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n\widehat{\pi}_k}\sum_{i=1}^n \tau_{i,k}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top, \qquad \forall k \in [K], \tag{14}$$

$$\widehat{\tau}_{i,k} = \frac{\pi_k \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k}(\boldsymbol{x}_i)}{\sum_{k'\in[K]} \pi_{k'} \varphi_{\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}}(\boldsymbol{x}_i)}, \qquad \forall k \in [K],\ \forall i \in [n]. \tag{15}$$

Based on this result, the EM algorithm is defined as in Figure 2. 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.

3 Graphical Lasso for Gaussian mixtures

The EM algorithm experiences severe performance degradation in high-dimensional setting. A technique to avoid this degradation is by regularizing the parameters of the model. The following algorithm is inspired by the Graphical lasso (Friedman et al., 2007) (Banerjee et al., 2008) which penalizes the components of the precision matrix of a Gaussian graphical model.

We consider $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ a sample of n points drawn from a p-dimensional Gaussian mixture distribution. In this problem, we will consider the estimation of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ with $\theta_k = (\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)$ where $\boldsymbol{\Omega}_k$ is the precision matrix regarding the k 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

$$\ell_n^{pen}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - pen(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k \varphi_{(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)}(\boldsymbol{x}_i) \right\} - pen(\boldsymbol{\theta}). \quad (16)$$

In this problem, we suppose that within a cluster k, most pairs of features (X^i, X^j) are independent given the other features X^l with $i, j, l \in [p], l \notin \{i, j\}$. This property entails the sparsity of $\boldsymbol{\Omega}_k$. Therefore, we consider an l_1 regularization $pen(\theta_k) = \lambda_k ||\boldsymbol{\Omega}_k||_{1,1}$ 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

$$F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) = \sum_{k=1}^{K} \left(\sum_{i=1}^{n} \left\{ \tau_{i,k} \log \varphi_{\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k}(\boldsymbol{x}_i) + \tau_{i,k} \log(\pi_k / \tau_{i,k}) \right\} - \lambda_k ||\boldsymbol{\Omega}_k||_{1,1} \right) \quad (17)$$

Then, we have the two following optimization problems

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) \in \arg\max_{\boldsymbol{\theta}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}), \qquad \widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\mathcal{T}}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) \quad (18)$$

which has explicit solutions. For a $\widehat{\boldsymbol{\mathcal{T}}}$ given, estimates of $(\pi, \boldsymbol{\mu})$ obtained by the first optimization problem in 18 are

$$\widehat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \widehat{\tau}_{i,k}, \qquad \widehat{\boldsymbol{\mu}}_k = \frac{1}{n\widehat{\pi}_k} \sum_{i=1}^{n} \widehat{\tau}_{i,k} \boldsymbol{x}_i, \qquad \forall k \in [K] \quad (19)$$

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

$$\widehat{\tau}_{i,k} = \frac{\widehat{\pi}_k \varphi_{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k}(\boldsymbol{x}_i)}{\sum_{k' \in [K]} \widehat{\pi}_{k'} \varphi_{\widehat{\boldsymbol{\mu}}_{k'}, \widehat{\boldsymbol{\Omega}}_{k'}}(\boldsymbol{x}_i)}, \qquad \forall k \in [K], \ \forall i \in [n]. \quad (20)$$

However, due to the penality $\lambda_k||\mathbf{\Omega}_k||_{1,1}$, the estimation of $\mathbf{\Omega}_k$ is not straightforward. We introduce the weighted empirical covariance matrix

$$\mathbf{\Sigma}_{n,k} = \frac{1}{n}\frac{\sum_{i=1}^{n}\tau_{i,k}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top}{\sum_{i=1}^{n}\tau_{i,k}} \tag{21}$$

The Gaussian density in equation (17) can be expanded as follows

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

The opposite minimization problem regarding each $\mathbf{\Omega}_k$ is

$$\mathbf{\Omega}_k \in \underset{\mathbf{\Omega}\succeq 0}{\arg\min}\Big\{-\frac{n\pi_k}{2}\log|\mathbf{\Omega}| + \frac{1}{2}\sum_{i=1}^{n}\tau_{i,k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\mathbf{\Omega}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) + \lambda_k||\mathbf{\Omega}||_{1,1}\Big\} \tag{22}$$

Using the well-known commutativity property of the trace operator and dividing by $n\pi_k$

$$\mathbf{\Omega}_k \in \underset{\mathbf{\Omega}\succeq 0}{\arg\min}\Big\{-\frac{1}{2}\log|\mathbf{\Omega}| + \frac{1}{2}tr(\mathbf{\Sigma}_{n,k}\mathbf{\Omega}) + \frac{\lambda_k}{n\pi_k}||\mathbf{\Omega}||_{1,1}\Big\} \tag{23}$$

Our algorithm solves a graphical lasso problem within each cluster. We use a block coordinate ascent algorithm (Mazumder, 2012) to solve this convex problem as in the graphical lasso implementation in R `http://statweb.stanford.edu/~tibs/glasso/`

The alternating maximization procedure is summarized in the following algorithm.

Input: data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ and the number of clusters K
Output: parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k, \widehat{\pi}_k\}_{k\in[K]}$

1: `Initialize` $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
2: **Repeat**
3: `Update the parameter` $\boldsymbol{\mathcal{T}}$:

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

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

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

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

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

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

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

Figure 3: Graphical lasso algorithm for Gaussian mixtures

4 Estimating the number of clusters

The idea is to add a regularization term on the estimation of the $n \times K$ matrix $\boldsymbol{\mathcal{T}}$, the estimate of the number of clusters K will be the number of non-empty columns of $\boldsymbol{\mathcal{T}}$.

We consider a maximum number of clusters M, we note the convex set $A = \{\tau \in \mathbb{R}^M : \sum_{k=1}^{M} \tau_k = 1, \tau_k \geq 0 \quad \forall k \in [M]\}$ and the "indicator" function $\chi_A(.)$ defined by:

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \in A, \\ \infty & \text{if } x = 0 \end{cases}$$

We note $\boldsymbol{\mathcal{T}}_{.,k}$ the k^{th} column and $\boldsymbol{\mathcal{T}}_{i,.}$ the i^{th} line of $\boldsymbol{\mathcal{T}}$. We will estimate $\boldsymbol{\mathcal{T}}$ using the same equation 17, 18 with a regularization term:

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

Removing the penalization on $\boldsymbol{\Omega}$:

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

and the optimization problem:

$$\widehat{\boldsymbol{\mathcal{T}}}(\boldsymbol{\theta}) \in \arg\max_{\boldsymbol{\mathcal{T}}} F^{pen}(\boldsymbol{\theta}, \boldsymbol{\mathcal{T}}) \tag{24}$$

Unfortunately, the regularization term prevents to derive explicit solution as in previous chapters. Furthermore, we cant separate the objective function since we optimize along columns and lines 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:

$$F^{pen}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{T}}) = f(\boldsymbol{\mathcal{T}}) + g(\boldsymbol{\mathcal{T}}) \tag{25}$$

with:

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

$$g(\boldsymbol{\mathcal{T}}) = \sum_{i=1}^{n} \chi_A(\boldsymbol{\mathcal{T}}_{i,.})$$

f is convex and differentiable on its domain, g is also convex but not smooth. We will tackle this problem by using a proximal method:

$$\begin{aligned}
\boldsymbol{\mathcal{T}}^{k+1} =& \mathbf{prox}_{\lambda g}(\boldsymbol{\mathcal{T}}^k - \lambda \nabla f(\boldsymbol{\mathcal{T}}^k)) \\
=& P_A(\boldsymbol{\mathcal{T}}^k - \lambda \nabla f(\boldsymbol{\mathcal{T}}^k)) \\
=& \underset{\boldsymbol{\mathcal{T}}:\forall K, \boldsymbol{\mathcal{T}}^k \in A}{\arg\min} \left(||\boldsymbol{\mathcal{T}} - (\boldsymbol{\mathcal{T}}^k - \lambda \nabla f(\boldsymbol{\mathcal{T}}^k))||_2^2 \right)
\end{aligned}$$

The gradient of f on $\boldsymbol{\mathcal{T}}$ is given by:

$$\begin{aligned}
\left[\nabla_{\boldsymbol{\mathcal{T}}} f(\boldsymbol{\mathcal{T}}) \right]_{i,j} =& \left[\frac{\partial f}{\partial \boldsymbol{\mathcal{T}}_{ij}}(\boldsymbol{\mathcal{T}}) \right]_{i,j} \\
=& \log(\varphi_{\boldsymbol{\mu}_j, \boldsymbol{\Omega}_j}(x_i)) + \log(\frac{\pi_j}{\tau_{i,j}}) + \frac{\tau_{i,j}}{||\boldsymbol{\mathcal{T}}_{.,j}||_2} - 1
\end{aligned}$$

We will use FISTA to accelerate the convergence

Input:
Output: parameter estimate $\boldsymbol{\mathcal{T}}$
1: `Initialize` $t_1 = 1$ `and` $\boldsymbol{\xi}^0$ `with`

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

2: **Repeat**

$$\begin{aligned}
\boldsymbol{\mathcal{T}}^k =& \underset{\boldsymbol{\mathcal{T}}:\forall K, \boldsymbol{\mathcal{T}}^k \in A}{\arg\min} \left(||\boldsymbol{\mathcal{T}} - (\boldsymbol{\xi}^k - \lambda \nabla f(\boldsymbol{\xi}^k))||_2^2 \right) \\
t^{k+1} =& \frac{1 + \sqrt{1 + 4 * (t^k)^2}}{2} \\
\boldsymbol{\xi}^{k+1} =& \boldsymbol{\mathcal{T}}^k + \left(\frac{t^k - 1}{t^{k+1}} \right) (\boldsymbol{\mathcal{T}}^k - \boldsymbol{\mathcal{T}}^{k-1})
\end{aligned}$$

Figure 4: $\boldsymbol{\mathcal{T}}$ estimation with FISTA

We use the algorithm of last chapter with the new estimation procedure of $\boldsymbol{\mathcal{T}}$

Input: data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ and the number of clusters K
Output: parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k, \widehat{\pi}_k\}_{k\in[K]}$
`Initialize` $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.
Repeat
    `Update the parameter` $\boldsymbol{\mathcal{T}}$ `with previous algorithm`
    `Update the parameter` $\boldsymbol{\theta}$:

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

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

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

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

    `increment` t: $t = t + 1$.
Until `stopping rule`.
Return $\boldsymbol{\theta}^t$.

Figure 5: Graphical lasso algorithm for Gaussian mixtures with cluster number discovery

5 Algorithm 2

We consider the diagonal matrix $D_\lambda = diag(\lambda_1, \dots, \lambda_K)$.

Input: data vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$, the number of clusters K and D_λ
Output: parameter estimate $\widehat{\boldsymbol{\theta}} = \{\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Omega}}_k, \pi_k\}_{k\in[K]}$

`Initialize` $t = 0$, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$.

Repeat

`Update the parameter` $\boldsymbol{\mathcal{T}}$:

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

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

$$(\mu^k, B^k) \in \underset{(\mu,B)\in\mathbb{R}^p\times\mathbb{R}^{p\times p}, B_{jj}=1}{\arg\min} \left\{ \frac{1}{N} \sum_{n=1}^{N} \tau_n^k(t) ||(x_n - \mu)^T B||_2^2 + ||D_\lambda B||_{1,1} \right\}$$

$\pi_k^{t+1} =$

$\boldsymbol{\mu}_k^{t+1} =$

$\boldsymbol{\Omega}_k^{t+1}$

`increment` t: $t = t + 1$.

Until `stopping rule`.

Return $\boldsymbol{\theta}^t$.

Figure 6: Lasso for Gaussian mixtures

6 Draft-A structural analysis on $\boldsymbol{\Sigma}$ approach

We consider a multivariate Gaussian distribution with mean $\boldsymbol{\mu}^*$ and covariance $\boldsymbol{\Sigma}^*$ and $Y_1, \dots, Y_N \in \mathbb{R}^p$ iid drawn from this distribution. We would like to estimate $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$. We know that $\widehat{\boldsymbol{\mu}}_n = \bar{Y}_n$, then wlog we consider $\mu^* = 0$, the problem is to estimate $\boldsymbol{\Sigma}^*$. We will study the precision matrix and consider that Σ^{-1} is sparse. We note $\Sigma^{-1} = \Omega$, Y_n the n-th random variable and Y_n^i the i-th component of this vector.
If $\Sigma_{ij}^{-1} = 0 \Rightarrow Y^i \perp\!\!\!\perp Y^j$ conditionally to $Y^{l\neq\{i,j\}}$. Thus, it makes sense to impose a L_1 penalty on Σ^{-1} to increase its sparsity.

6.1 Graphical Lasso

Let consider a multivariate normal distribution with parameters $\mu^*,\ \Sigma^*$ with density;

$$\mathcal{N}(x|\mu^*, \Sigma^*) = \frac{1}{(2\pi)^{d/2}|\Sigma^*|^{1/2}} \exp^{-\frac{1}{2}(x-\mu^*)^T\Sigma^{-1*}(x-\mu^*)} \tag{26}$$

We consider $\mu = 0$. Given N datapoints $X_1, \ldots, X_N$ and $X_i \in \mathbb{R}^d$, the log-likelihood is given by:

$$\begin{aligned}\mathcal{L}(\Sigma) &= \log\left(\prod_{n=1}^{N} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp^{-\frac{1}{2}(x_n)^T\Sigma^{-1}(x_n)}\right) \\ &= -\frac{dN}{2}\log 2\pi - \frac{N}{2}\sum_{n=1}^{N}\log|\Sigma^*| - \frac{1}{2}\sum_{n=1}^{N} x_n^T\Sigma^{*,-1}x_n\end{aligned} \tag{27}$$

Note that $x_n^T\Sigma^{*,-1}x_n = tr(x_n^T\Sigma^{*,-1}x_n)$, and therefore:

$$\sum_{n=1}^{N} x_n^T\Sigma^{*,-1}x_n = tr\Big(\sum_{n=1}^{N} x_n^T\Sigma^{*,-1}x_n\Big) = tr\Big(\big[\sum_{n=1}^{N} x_n^T x_n\big]\Sigma^{*,-1}\Big) = tr(S_N\Sigma^*) \tag{28}$$

Where S_N is the empirical covariance matrix. We can replace that in the log-likelihood expression:

$$\mathcal{L}(\Sigma) = -\frac{dN}{2}\log 2\pi - \frac{N}{2}\sum_{n=1}^{N}\log|\Sigma^*| - \frac{1}{2}tr(S_N\Sigma^*) \tag{29}$$

Finally:

$$\mathcal{L}(\Sigma) = C + \frac{N}{2}\log|\Sigma^{-1}| - \frac{1}{2}tr(S_n\Sigma^{-1}) \tag{30}$$

Where C is a constant (dependent on N). Thus, considering the sparsity of the precision matrix $\Omega = \Sigma^{-1}$, we impose a penalization to the maximum likelihood estimator of Ω

$$\widehat{\Omega} \in argmin\big\{\log|\Omega| - tr(S_N\Omega) - \lambda||\Omega||_1\big\} \tag{31}$$

A reason to use the L_1 penalization instead of the ridge is that for an L_p penalization, the problem is convex for $p \geq 1$ and we have parsimonious property for $p \leq 1$.
This is a convex optimization problem, however the complexity is $O(p^3)$ (Source, high dim & var select Buhlmann 2006 ? Wassermann)

6.2 Column-Wise Lasso

We consider a gaussian vector $Y \in \mathbb{R}^d$, $Y \sim \mathcal{N}(0, \Sigma)$. We can write $Y = (Y^1, Y^{2:d})$. With this decomposition we can write the covariance matrix as following:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{pmatrix} \tag{32}$$

and according to theorem[?]: If Σ_{22} is inversible, then:

$$\begin{array}{rcl} \mathbb{E}[Y^1|Y^{2:d}] & = & \Sigma_{12}\Sigma_{22}^{-1}Y^{2:d} \\ Var[Y^1|Y^{2:d}] & = & \sigma_1^2 - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^T \end{array} \tag{33}$$

We have the following identity:

$$\begin{pmatrix} \omega_{11} & \Omega_{12} \\ \Omega_{12}^T & \Omega_{22} \end{pmatrix} \begin{pmatrix} \sigma_1^2 & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & I_{p-1} \end{pmatrix} \tag{34}$$

Which gives the following equations:

$$\begin{cases} \omega_{11}\sigma_1^2 + \Omega_{12}\Sigma_{12}^T & = 1 \qquad (*) \\ \omega_{11}\Sigma_{12} + \Omega_{12}\Sigma_{22} & = 0 \qquad (**) \\ \Omega_{12}^T\Sigma_{12} + \Omega_{22}\Sigma_{22} & = I_{p-1} \quad (***) \end{cases} \tag{35}$$

With (**) we have $-\omega_{11}\Sigma_{12}\Sigma_{22}^{-1} = \Omega_{12}$ and injected to (*) we have:

$$\begin{cases} \mathbb{E}[Y^1|Y^{2:d}] & = -\frac{1}{\omega_{11}}\Omega_{12}Y^{2:d} \\ Var[Y^1|Y^{2:d}] & = \frac{1}{\omega_{11}} \end{cases} \tag{36}$$

Finally, $Y^1 - \mathbb{E}[Y^1|Y^{2:d}]$ is a gaussian vector of $\mathbb{R}^{d-1}$, centered, independent of $Y^{2:d}$ and of covariance matrix $\sigma_1^2 - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^T$. If we denote $\xi^1 \sim \mathcal{N}(0,1)$ we have $Y^1 - \mathbb{E}[Y^1|Y^{2:d}] = \frac{1}{\sqrt{\omega_{11}}}\xi^1$.

Therefore, for $Y_1, \ldots, Y_n$ iid of law $\mathcal{N}(0, \Sigma^*)$ we have:

$$\begin{array}{rcl} Y_i^1 & = & -\frac{1}{\omega_{11}^*}\Omega_{12}Y_i^{2:d} + \frac{1}{\sqrt{\omega_{11}^*}}\xi_i^1 \\ & = & -\sum_{j=2}^d \frac{w_{ij}^*}{\omega_{11}^*}Y_i^j + \frac{1}{\sqrt{\omega_{11}^*}}\xi_i^1 \end{array} \tag{37}$$

and

$$\beta_1^{*T}Y_i = \frac{1}{\sqrt{\omega_{11}^*}}\xi_i^1 \Rightarrow \beta_1^{*T}\boldsymbol{Y} = \frac{1}{\sqrt{\omega_{11}^*}}\boldsymbol{\xi}^1 \tag{38}$$

with

$$\beta_1^* = \frac{1}{\sqrt{\omega_{11}^*}}\begin{bmatrix} w_{11}^* \\ w_{12} \\ \vdots \\ w_{1d} \end{bmatrix} \in \mathbb{R}^d \quad \text{and} \quad \boldsymbol{Y} = \big[verifier\big] \tag{39}$$

6.3 The Square-Root Lasso

7 Comments

References

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.

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.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2007.

R. Mazumder. Topics in sparse multivariate statistics (thesis). 2012.