Unmixing known Gaussian mixtures

Ami Wiesel

I. Problem formulation

Our model consists of K Gaussian components with known statistics

$$\mathbf{x}_i \sim N\left(\mathbf{m}_i, \mathbf{Q}_i\right) \quad i = 1, \cdots, K \tag{1}$$

where $\mathbf{x}_i$ are p dimensional vectors, $\mathbf{m}_i$ are known length p mean vectors, and $\mathbf{Q}_i \succeq \mathbf{0}$ are known positive semidefinite covariance matrices. We also use matrix notations in which we stack these vectors into

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, \cdots, \mathbf{x}_K] \\ \mathbf{M} &= [\mathbf{m}_1, \cdots, \mathbf{m}_K] \end{aligned} \tag{2}$$

and decompose $\mathbf{Q}_i = \mathbf{R}_i \mathbf{R}_i^T$ where $\mathbf{R}_i$ are full rank matrices of dimensions $p \times r$ and r is their rank. For simplicity, we assume that the ranks of all the components are identical. In some parts of this work, we will also consider the special spherical case in which $\mathbf{R}_i = \sigma_i \mathbf{I}$ and $\mathbf{Q}_i = \sigma_i^2 \mathbf{I}$.

We observe a convex combination of these vectors corrupted by noise

$$\mathbf{y} = \mathbf{X}\mathbf{a} + \mathbf{v} \tag{3}$$

where $\mathbf{v} \sim N\left(\mathbf{0}\, \sigma^2 \mathbf{I}\right)$, and $\mathbf{a}$ is a length K unknown abundance vector which satisfies

$$\mathcal{A} = \left\{ \mathbf{a} \,:\, \mathbf{a} \geq \mathbf{0},\ \mathbf{1}^T \mathbf{a} = 1 \right\} \tag{4}$$

Together, we have access to

$$\mathbf{y} \sim N\left(\mathbf{M}\mathbf{a}, \mathbf{Q}\left(\mathbf{a}\right)\right) \tag{5}$$

where $\mathbf{Q}\left(\mathbf{a}\right)$ is the total covariance which is also parameterized by $\mathbf{a}$

$$\mathbf{Q}\left(\mathbf{a}\right) = \sum_{i=1}^{K} a_i^2 \mathbf{Q}_i + \sigma^2 \mathbf{I} \tag{6}$$

and want to estimate $\mathbf{a}$.

In the sequel, we will resort to asymptotic analysis in which we use the following statistical model. These assumptions are only for analysis and are not necessary for the algorithms. We will consider the case where $p \to \infty$ but K and r remain fixed. We will assume that the rows of $\mathbf{M}$ are randomly generated as i.i.d. $N\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{M}}\right)$. Similarly, the rows of $\mathbf{R}_i$ are generated i.i.d. $N\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{R}}\right)$. In the special case where $\mathbf{\Sigma}_{\mathbf{R}} = \mathbf{0}$ this is exactly the classical asymptotic linear model, in which linear regression is asymptotically consistent as long as $\mathbf{M}^T\mathbf{M}/p \to \mathbf{\Sigma}_{\mathbf{M}} \succ \mathbf{0}$.

II. Maximum Likelihood

The straight forward approach to the estimation problem is to choose the estimate as the parameter that maximizes the marginal likelihood:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a} \in \mathcal{A}} \left(\mathbf{y} - \mathbf{M}\mathbf{a}\right) \mathbf{Q}^{-1}\left(\mathbf{a}\right) \left(\mathbf{y} - \mathbf{M}\mathbf{a}\right) + \log\left(\left|\mathbf{Q}\left(\mathbf{a}\right)\right|\right) \tag{7}$$

A. EM Algorithm

The problem has a natural interpretation where $\mathbf{X}$ are latent variables. This leads to a simple EM minimization approach. Here, we iteratively solve the minimization given the expected values of $\mathbf{X}$:

$$\hat{\mathbf{a}} \leftarrow \min_{\mathbf{a}\in\mathcal{A}} \overline{\|\mathbf{y}-\mathbf{X}\mathbf{a}\|^2} \tag{8}$$

where the overline denotes expectation with respect to the previous parameters. Let $\mathbf{x} = \left[\mathbf{x}_1^T, \cdots, \mathbf{x}_K^T\right]$ and $\mathbf{m} = \left[\mathbf{m}_1^T, \cdots, \mathbf{m}_K^T\right]$ be concatenated versions of $\mathbf{X}$ and $\mathbf{M}$, respectively. Then, the prior probability of $\mathbf{x}$ is

$$\mathbf{x} \sim N\left(\mathbf{f}, \operatorname{diag}\left\{\mathbf{Q}_j\right\}\right) \tag{9}$$

and its posterior after observing $\mathbf{y}$ parameterized by $\mathbf{a}$ is

$$\begin{aligned}
\mathrm{E}\left\{\mathbf{x}/y\right\} &= \mathbf{m} - \operatorname{diag}\left\{\mathbf{Q}_j\right\}(\mathbf{a}\otimes\mathbf{I})\left[\left(\mathbf{a}^T\otimes\mathbf{I}\right)\operatorname{diag}\left\{\mathbf{Q}_j\right\}(\mathbf{a}\otimes\mathbf{I}) + \sigma^2\mathbf{I}\right]^{-1}(\mathbf{y}-\mathbf{M}\mathbf{a}) && (10)\\
&= \mathbf{m} - \operatorname{diag}\left\{\mathbf{Q}_j\right\}(\mathbf{a}\otimes\mathbf{I})\,\mathbf{Q}^{-1}(\mathbf{a})\,(\mathbf{y}-\mathbf{M}\mathbf{a}) && (11)\\
&= \mathbf{m} - \operatorname{diag}\left\{\mathbf{Q}_j\right\}\left(\mathbf{a}\otimes\mathbf{Q}^{-1}(\mathbf{a})\,(\mathbf{y}-\mathbf{M}\mathbf{a})\right) && (12)\\
\operatorname{cov}\{\mathbf{x}/y\} &= \operatorname{diag}\left\{\mathbf{Q}_j\right\} - \operatorname{diag}\left\{\mathbf{Q}_j\right\}(\mathbf{a}\otimes\mathbf{I})\,\mathbf{Q}^{-1}(\mathbf{a})\left(\mathbf{a}^T\otimes\mathbf{I}\right)\operatorname{diag}\left\{\mathbf{Q}_j\right\} && (13)\\
&= \operatorname{diag}\left\{\mathbf{Q}_j\right\} - \operatorname{diag}\left\{\mathbf{Q}_j\right\}\left(\mathbf{a}\mathbf{a}^T\otimes\mathbf{Q}^{-1}(\mathbf{a})\right)\operatorname{diag}\left\{\mathbf{Q}_j\right\} && (14)\\
& && (15)
\end{aligned}$$

Thus, we get the following iterations

$$\hat{\mathbf{a}} \leftarrow \min_{\mathbf{a}\in\mathcal{A}} \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\overline{\mathbf{X}}\mathbf{a} + \mathbf{a}^T\overline{\mathbf{X}^T\mathbf{X}}\mathbf{a} \tag{16}$$

and

$$\overline{\mathbf{x}_i} = \mathbf{m}_i + a_i\mathbf{Q}_i\mathbf{Q}^{-1}(\mathbf{a})\left[\mathbf{y}-\mathbf{M}\mathbf{a}\right] \tag{17}$$

$$\left[\overline{\mathbf{X}^T\mathbf{X}}\right]_{ij} = \left[\overline{\mathbf{X}}^T\overline{\mathbf{X}}\right]_{ij} + \operatorname{Tr}\left\{\mathbf{Z}^{ij}\right\} \tag{18}$$

$$\begin{aligned}
\mathbf{Z}^{ii} &= \mathbf{Q}_i - a_i^2\mathbf{Q}_i\mathbf{Q}^{-1}(\mathbf{a})\,\mathbf{Q}_i\\
\mathbf{Z}^{ij} &= \mathbf{0} - a_ia_j\mathbf{Q}_i\mathbf{Q}^{-1}(\mathbf{a})\,\mathbf{Q}_j && (19)
\end{aligned}$$

Therefore, each iteration consists of a quadratic minimization over the simplex. This is of course a standard quadratic program, which can be solved instantly. We summarize the EM algorithm in figure 1, in the general case of several observation of $\mathbf{y}_i$.

The EM algorithm usually suffers from slow convergence. We suggest acceleration scheme as follows: given the last two EM iterates $\mathbf{x}^{(t-1)}$ and $\mathbf{x}^{(t)}$, we extrapolate as follows:

$$\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t)} + \gamma(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}).$$

The choice of γ depends much on the problem dimension. A proper γ can reduce the number of EM iterations.

For high dimensional problems, the EM algorithm might be a slow estimate even after acceleration. Given that $\mathbf{a}$ is sparse, we suggest a greedy search algorithm for the support. This may reduce the problem to a low dimensional one, and considerably accelerate computation. Figure 2 describes the search algorithm. We begin with an empty set of $\mathbf{a}$'s support indexes, denoted by $\mathcal{S}$. In each iteration, we choose the next index to be the one that yields the largest ML, under the regular constraints. We then add that index to the set $\mathcal{S}$. The maximum likelihood of a given support can be calculated by the EM algorithm. Thus,

Algorithm 1 EM algorithm: EM$(\mathbf{y}_i, \mathbf{M}, \mathbf{Q}_i, \sigma^2)$

Input: data points $\mathbf{y}_i \in \mathbb{R}^n, \quad i = 1, .., d,$.

Expectation matrix $\mathbf{M} = [\mathbf{m}_1, ..., \mathbf{m}_p] \in \mathbb{R}^{n \times p}$.

Covariance matrices $\mathbf{Q}_i \in \mathbb{R}^{n \times n}, \quad i = 1, ..., p$.

Noise variance σ^2.

Initialization: set $\mathbf{x}^{(t)} = \frac{1}{p}\mathbf{1}^{p \times 1}, \quad t = 0$;

Until convergence

- $\mathbf{Q}_{\mathbf{x}^{(\mathbf{t})}} \leftarrow \sum_{i=1}^{p} (\mathbf{x}_i^{(t)})^2 \mathbf{Q}_i + \sigma^2 \mathbf{I}$;
- Compute $\overline{\mathbf{X}} = [\overline{\mathbf{x}}_1, ..., \overline{\mathbf{x}}_p] \in \mathbb{R}^{nd \times p} : \overline{\mathbf{x}}_i \leftarrow \mathbf{1}^{d \times 1} \otimes \mathbf{m}_i + \text{Vec}\Big(\mathbf{x}_i^{(t)} \mathbf{Q} \mathbf{Q}_{\mathbf{x}^{(\mathbf{t})}}^{-\mathbf{1}} (\mathbf{y} - \mathbf{1}^{1 \times d} \otimes \mathbf{M}\mathbf{x}^{(t)})\Big)$;
- Compute $\overline{\mathbf{X}^T \mathbf{X}}$: $\Big[\overline{\mathbf{X}^T \mathbf{X}}\Big]_{i,j} \leftarrow \Big[\overline{\mathbf{X}}^T \overline{\mathbf{X}}\Big]_{i,j} - d \cdot \text{Tr}\Big(\mathbf{x}_i^{(t)} \mathbf{x}_j^{(t)} \mathbf{Q}_i \mathbf{Q}_{\mathbf{x}^{(\mathbf{t})}}^{-\mathbf{1}} \mathbf{Q}_j\Big); \quad \forall i, j \in \{1, .., p\}$

- $\Big[\overline{\mathbf{X}^T \mathbf{X}}\Big]_{i,i} \leftarrow \Big[\overline{\mathbf{X}^T \mathbf{X}}\Big]_{i,i} + d \cdot \text{Tr}\Big(\mathbf{Q}_i\Big), \quad \forall i = 1, .., p.$
- $\mathbf{x}^{(t+1)} = \min_{\mathbf{u}} \quad \mathbf{u}^T \Big[\overline{\mathbf{X}^T \mathbf{X}}\Big] \mathbf{u} - 2(\text{Vec}(\mathbf{y}))^T \overline{\mathbf{X}} \mathbf{u} \quad \text{s.t} \quad \mathbf{u} \geq 0, \quad \mathbf{1}^T \mathbf{u} = 1.$
- $t \leftarrow t + 1$.

Return $\mathbf{x}^{(t)}$.

Algorithm 2 Greedy search algorithm

Input: data points $\mathbf{y}_i \in \mathbb{R}^n, \quad i = 1, .., d.$

Sparsity constant s.

Likelihood function $\mathcal{L}(\mathbf{y}; \mathbf{a})$

Initialization: Set $k = 1$, $\mathcal{S} = \varnothing$.

While $|\mathcal{S}| < s$

- Find index $i \notin \mathcal{S}$ that leads to the largest $\mathcal{L}(\mathbf{y}; \mathbf{a})$, s.t: $\mathbf{1}^T \mathbf{a} = 1, \quad \mathbf{a} \geq \mathbf{0}$, Supp$(\mathbf{a}) = \mathcal{S} \cup \{i\}$.
- Update: $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$.

End while

Return $\mathcal{S}$

in our model, we only need to use $\mathbf{Q}_i$ and $\mathbf{m}_i$ corresponding to the current support. At the final stage, we fully perform the EM algorithm on the $\mathcal{S}$ we have calculated.

B. Aitken acceleration

After 3 consecutive iteration results $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{u}^{(3)}$, the Aitken update is given by:

$$\frac{u_i^{(1)} - (u_i^{(2)} - u_i^{(1)})^2}{u_i^{(3)} - 2u_i^{(2)} + u_i^{(1)}}, \quad , \forall i \in \{1, .., p\}, \tag{20}$$

as long as the denominator is non zero.

Algorithm 3 $\mathbf{M}, \mathbf{Q}_i, \mathbf{A}$ estimation

Input: data points $\mathbf{y}_i \in \mathbb{R}^n, \quad i = 1, .., d.$

Sparsity constant s.

dimension p.

Initialization: Compute p clusters of $\mathbf{y}_i$, via k-means. Set $\hat{\mathbf{M}} \in \mathbb{R}^{n \times p}$ to be the centroid matrix. $\hat{\mathbf{Q}}_i \leftarrow \mathbf{I}^{p \times p}$.

Until convergence

- **Set** $\mathbf{M} = \mathbf{0}^{n \times p}$, $\mathbf{Q}^{(i)} = \mathbf{0}^{n \times p}, \quad i = 1, .., p, \quad \mathbf{z} = \mathbf{0}^{p \times 1}$.
- **For each** $\mathbf{y}_i$, $i = 1, .., d$:
  - Compute $\hat{\mathcal{S}}$ using algorithm 2.
  - $(\hat{\mathbf{a}}_i)_{(\hat{\mathcal{S}})} \leftarrow \text{EM}(\mathbf{y}_i, \hat{\mathbf{M}}_{(\hat{\mathcal{S}})}, \hat{\mathbf{Q}}_{i \in \hat{\mathcal{S}}}, \sigma^2)$ (Algorithm 1)
  - Compute $E(\mathbf{X}_{\hat{\mathcal{S}}} | \mathbf{y}_i)$ and $E(\mathbf{X}_j \mathbf{X}_j^T | \mathbf{y}_i)$ for each $j \in \hat{\mathcal{S}}$.
  - $\mathbf{Q}^{(j)} \leftarrow \mathbf{Q}^{(j)} + E(\mathbf{X}_j \mathbf{X}_j^T | \mathbf{y}_i) + E(\mathbf{X}_j | \mathbf{y}_i) E^T(\mathbf{X}_j | \mathbf{y}_i), \quad z_j \leftarrow z_j + 1, \quad j \in \hat{\mathcal{S}}$
  - $\mathbf{M}_{\hat{\mathcal{S}}} \leftarrow \mathbf{M}_{\hat{\mathcal{S}}} + E(\mathbf{X}_{\hat{\mathcal{S}}} | \mathbf{y}_i)$
  - $\mathbf{M}_j \leftarrow \frac{\mathbf{M}_j}{z_j}, \quad j = 1, ..., p$
  - $\mathbf{Q}^{(j)} \leftarrow \frac{\mathbf{Q}^{(j)}}{z_j} - \mathbf{M}\mathbf{M}^T \quad j = 1, ..., p.$
- **end for**
- $\hat{\mathbf{M}} \leftarrow \mathbf{M}, \quad \hat{\mathbf{Q}}_j \leftarrow \mathbf{Q}^{(j)}$.

End while

Return $\mathcal{S}$

C. Estimating everything: mean and covariance matrices

We now consider the following settings: We have observation matrix $\mathbf{Y}^{n \times d}$, such that each column of $\mathbf{Y}$, denoted by $\mathbf{y}_i$, has been generated as follows:

$$\mathbf{Y}_i = \mathbf{X}^{(i)} \mathbf{a}^{(i)} + \epsilon, \quad , \mathbf{X}_j^{(i)} \sim \mathcal{N}(\mathbf{m}_j, \mathbf{Q}_j). \tag{21}$$

Our benchmark algorithm is an alternating minimization procedure. In the initialization step, we estimate $\mathbf{M}$ and $\mathbf{A}$ by mixture model, assuming no correlation between components. In particular, we assume that all $\mathbf{a}^{(i)}$ are 1-sparse vectors. This assumptions allows us to perform k-means estimation of $\mathbf{M}$. Given estimates of $\mathbf{M}$ and $\mathbf{Q}_j$, we can run the previous EM algorithm to estimate the $\mathbf{a}$'s. Since that procedure is based on the conditional expectation $E(\mathbf{X}|\mathbf{y})$, we can use that to further estimate the mean and covariance, simply by averaging all $E(X|\mathbf{y}_i$ for the relevant support of $\mathbf{a}^{(i)}$. We also calculate $E(\mathbf{X}\mathbf{X}^T|\mathbf{y}_i)$ for the sample covariance. We then plug in the new mean and covariance estimates for the next iteration. A detailed pseudocode is given below, Algorithm 3: