Probabilistic Graphical Models

Homework 2

Authors

Bastien Déchamps
bastien.dechamps@eleves.enpc.fr

and

Mathieu Orhan
mathieu.orhan@eleves.enpc.fr

Supervisors

Pierre Latouche

and

Nicolas Chopin

1 Classification: K-means, and the EM algorithm

1. Let us consider data points $(x_i)_{1\leq i\leq n} \in \mathbb{R}^d$. We introduce K latent variables $(z_i)_{1\leq i\leq n}$ such that $P(z_i = k) = p_k$ and $x_i \mid z_i = k \sim \mathcal{N}(\mu_k, D_k)$, where D_k is assumed to be a diagonal matrix. We denote $\theta = (p_k, D_k, \mu_k)_{1\leq k\leq K}$ the parameters to estimate. EM algorithm seeks to maximize the log-likelihood which is expressed as:

$$\log p\left(X|\theta\right) = \sum_{i=1}^{n} \log\left(\sum_{l=1}^{K} p_l \mathcal{N}\left(x_i \mid \mu_l, D_l\right)\right) \tag{1}$$

E-step Using Bayes formula, we compute the posterior probabilities $\gamma_{ki} = p\left(Z_i = k|X\right)$:

$$\boxed{\gamma_{ki} = \frac{p_k \mathcal{N}\left(x_i \mid \mu_k, D_k\right)}{\sum_{l=1}^{K} p_l \mathcal{N}\left(x_i \mid \mu_l, D_l\right)}} \tag{E}$$

M-step We estimate θ given the values of the posterior probabilities γ_{ki}.
We first maximize the log likelihood wrt $p = (p_k)_{1\leq k\leq K}$ under the constraint that $\sum_{l=1}^{K} p_l = 1$.
This can be achieved using the Lagrange multiplier λ and by maximizing this quantity:

$$\log p\left(X|\theta\right) + \lambda\left(\sum_{l=1}^{K} p_l - 1\right)$$

If we denote $n_k = \sum_{i=1}^{n} \gamma_{ki}$, it leads to:

$$\boxed{p_k = \frac{n_k}{n}} \tag{2}$$

Let's derivate $\log p\left(X|\theta\right)$ with respect to μ_k and set it to 0:

$$0 = \sum_{i=1}^{n} \gamma_{ki}(x_i - \mu_k)$$

This leads to the following equation:

$$\boxed{\mu_k = \frac{1}{n_k}\sum_{i=1}^{n} \gamma_{ki} x_i}$$

Let's derivate $\log p(X|\theta)$ with respect to D_{kj} the j-th element of the diagonal of D_k, and set it to 0:

$$0 = \sum_{i=1}^{n} \gamma_{ki} \left[-D_{kj} + (\mu_{kj} - x_{ij})^2 \right]$$

This leads to the following equation:

$$\boxed{D_{kj} = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\mu_{kj} - x_{ij})^2}$$

2. This model contains much less parameters than the standard Gaussian mixture model (only $(2d+1)K$ compared with $K\left(\frac{d(d+1)}{2} + d + 1\right)$ for the full covariance model). It allows faster computations while having a performance close to the full covariance model if the features are not too correlated.

3. In appendix A (figures 4, 5, 6) are represented the results of the diagonal EM as well as the k-means and the full covariance matrix EM on the Iris dataset, for $K \in \{2, 3, 4\}$ and for each pair of features. The confidence ellipses and the means of each clusters are also represented when possible.

4. The k-means algorithm optimizes for *compactness* and will totally fail at clustering crossing Gaussian distributions. An example of such a clustering is represented on figure 1. Both the EM algorithms perform well on retrieving the cross shaped clustering.

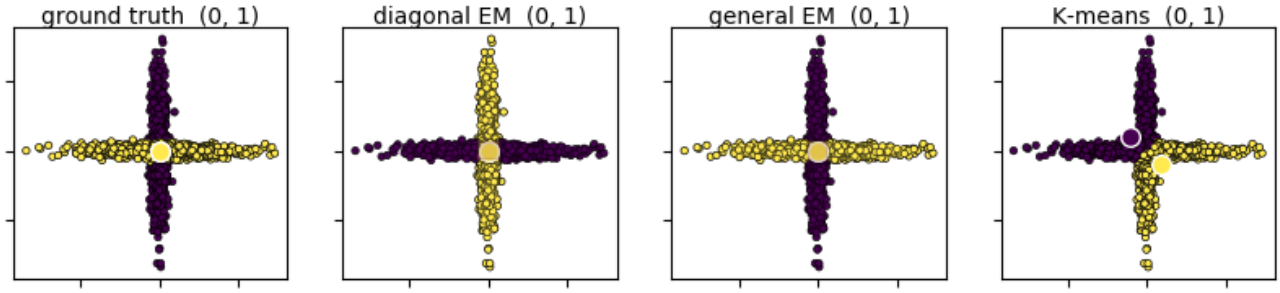

Figure 1: Limits of the k-means clustering

2 Graphs, algorithms and Ising

1. We start by considering the sum-product algorithm on an undirected chain represented by a graph G. The joint probability $p(x_1, ..., x_n)$ can be factorized on the cliques of G using

the potentials $\psi_i(x_i)$ and $\psi_{i,i+1}(x_i, x_{i+1})$:

$$p(x_1, ..., x_n) = \frac{1}{Z} \prod_{i=1}^{n} \psi_i(x_i) \prod_{i=1}^{n-1} \psi_{i,i+1}(x_i, x_{i+1})$$

where Z is the partition function. The marginalization of one node i can be obtained using the messages from its neighbors $i-1$ and $i+1$. Let's write these messages $\mu_{i-1\to i}$ and $\mu_{i+1\to i}$. Then:

$$p(x_i) = \frac{1}{Z} \mu_{i-1\to i} \mu_{i+1\to i} \psi_i(x_i)$$

Descending message $\mu_{i\to i-1}(x_{i-1})$:

$$\mu_{i\to i-1}(x_{i-1}) = \sum_{x_i} \psi_{i-1,i}(x_{i-1}, x_i) \psi_i(x_i) \mu_{i+1\to i}(x_i)$$

Ascending message $\mu_{i\to i+1}(x_{i+1})$:

$$\mu_{i\to i+1}(x_{i+1}) = \sum_{x_i} \psi_{i,i+1}(x_i, x_{i+1}) \psi_i(x_i) \mu_{i-1\to i}(x_i)$$

After propagating the messages, Z can be computed by normalizing any marginal.

If states are discretes, then it is straightforward to represent potentials as array of real numbers. If they are not discrete, it depends, it could be an approximation, a discretization or an analytical form. An example is given on figure 2 where the marginalization laws are represented for 3 different distributions.

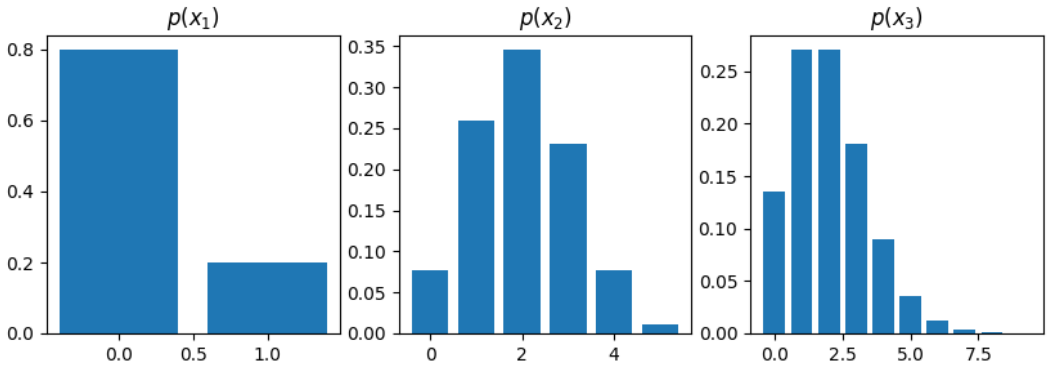

Figure 2: Marginalization laws for $\psi_1 = (4, 1)$, $\psi_2 \propto \mathcal{B}(5, 0.4)$, $\psi_3 \propto \mathcal{P}(2)$ (truncated on $k = 10$) obtained with the sum-product algorithm. Edge potential are all set to 1 (independant)

2. Let's consider the Ising model on binary variables $X_1, ..., X_n$ nodes of a grid of size $h \times w$.

$$p(x_1, ..., x_n) = \frac{1}{Z(\alpha, \beta)} \exp\left(\alpha \sum_i x_i + \beta \sum_{i \sim j} \mathbf{1}_{x_i = x_j}\right)$$

Where we note $i \sim j$ when i and j are neighbors. We will now use the idea of the junction tree algorithm to use the algorithm implemented question 1. We consider a grid of size $h = 100$ and $w = 10$. If we group the rows, the cardinal of the state space of one group is exactly $2^{10} = 1024$ and the resulting chain is of length 100.

The joint probability $p(x_1, ..., x_n)$ can be factorized on the rows $r_1, ..., r_h$ using h potentials ϕ_k and $\phi_{k,k+1}$:

$$p(x_1, ..., x_n) = \frac{1}{Z} \prod_{k=1}^{h} \phi_k(x_{r_k}) \prod_{k=1}^{h-1} \phi_{k,k+1}(x_{r_k}, x_{r_{k+1}})$$

Where $\phi_k(x_{r_k})$ is defined as:

$$\phi_k(x_{r_k}) = \prod_{j=1}^{w} \exp\left(\alpha x_{(k-1)w+j}\right) \prod_{j=1}^{w-1} \exp\left(\beta \mathbf{1}_{x_{(k-1)w+j} = x_{(k-1)w+j+1}}\right)$$

And $\phi_{k,k+1}$, as:

$$\phi_{k,k+1}(x_{r_k}, x_{r_{k+1}}) = \prod_{j=1}^{w} \exp\left(\beta \mathbf{1}_{x_{(k-1)w+j} = x_{kw+j}}\right)$$

The evolution of the partition function Z with respect to the interaction parameter β is represented on figure 3.

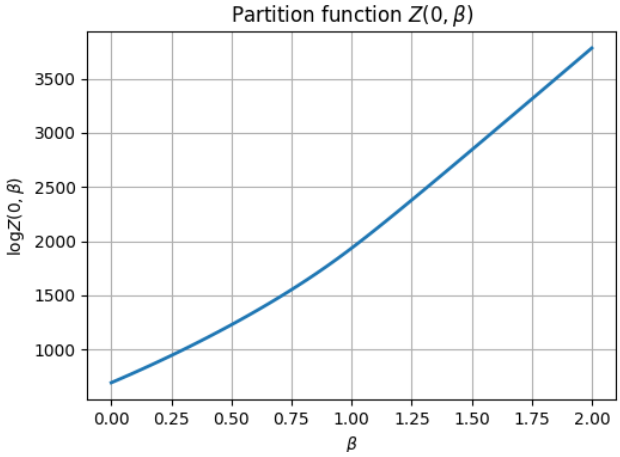

Figure 3: Evolution of $\log Z(0, \beta)$ wrt β

A Models comparison on Iris dataset

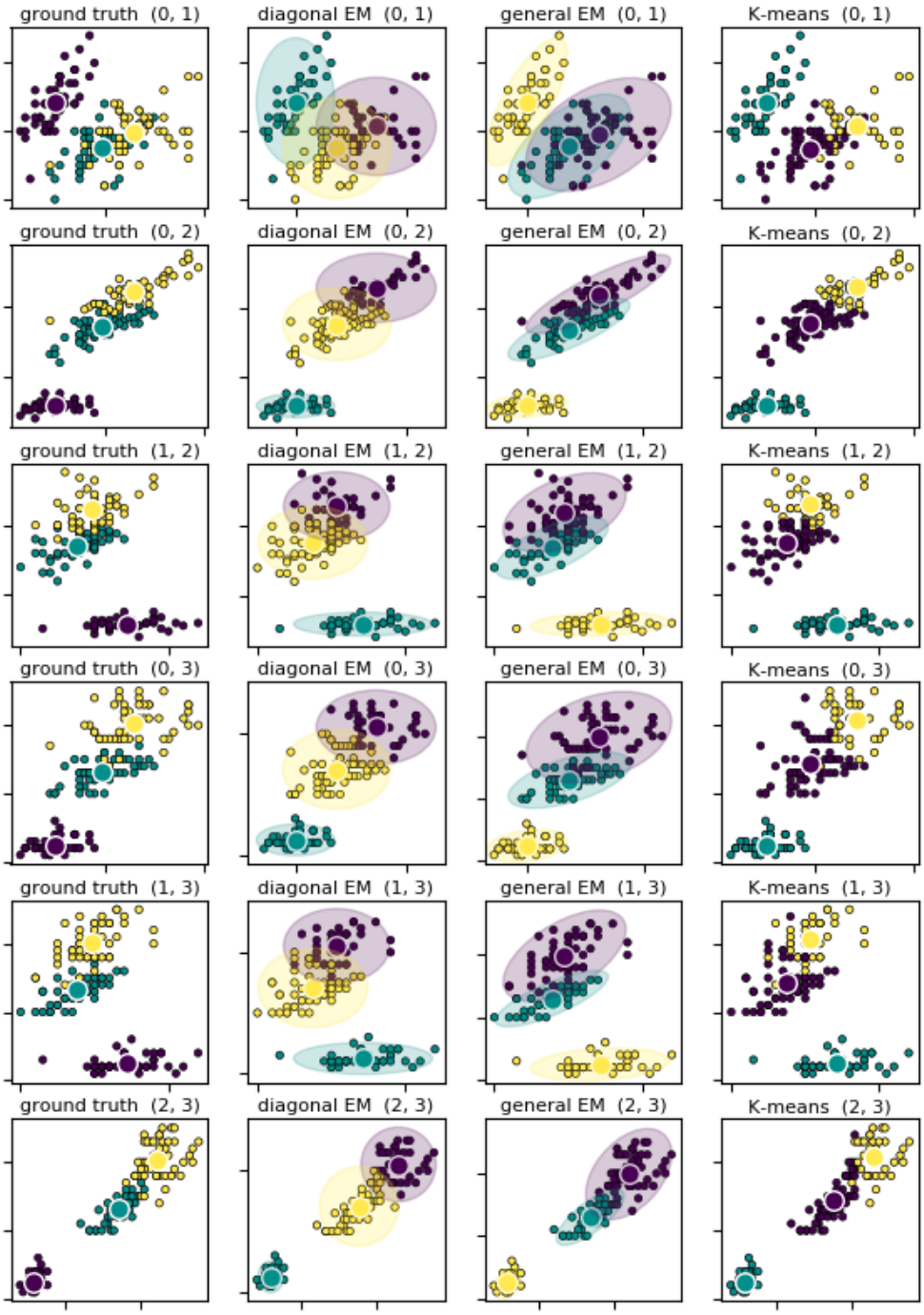

Figure 4: Results for $K = 3$

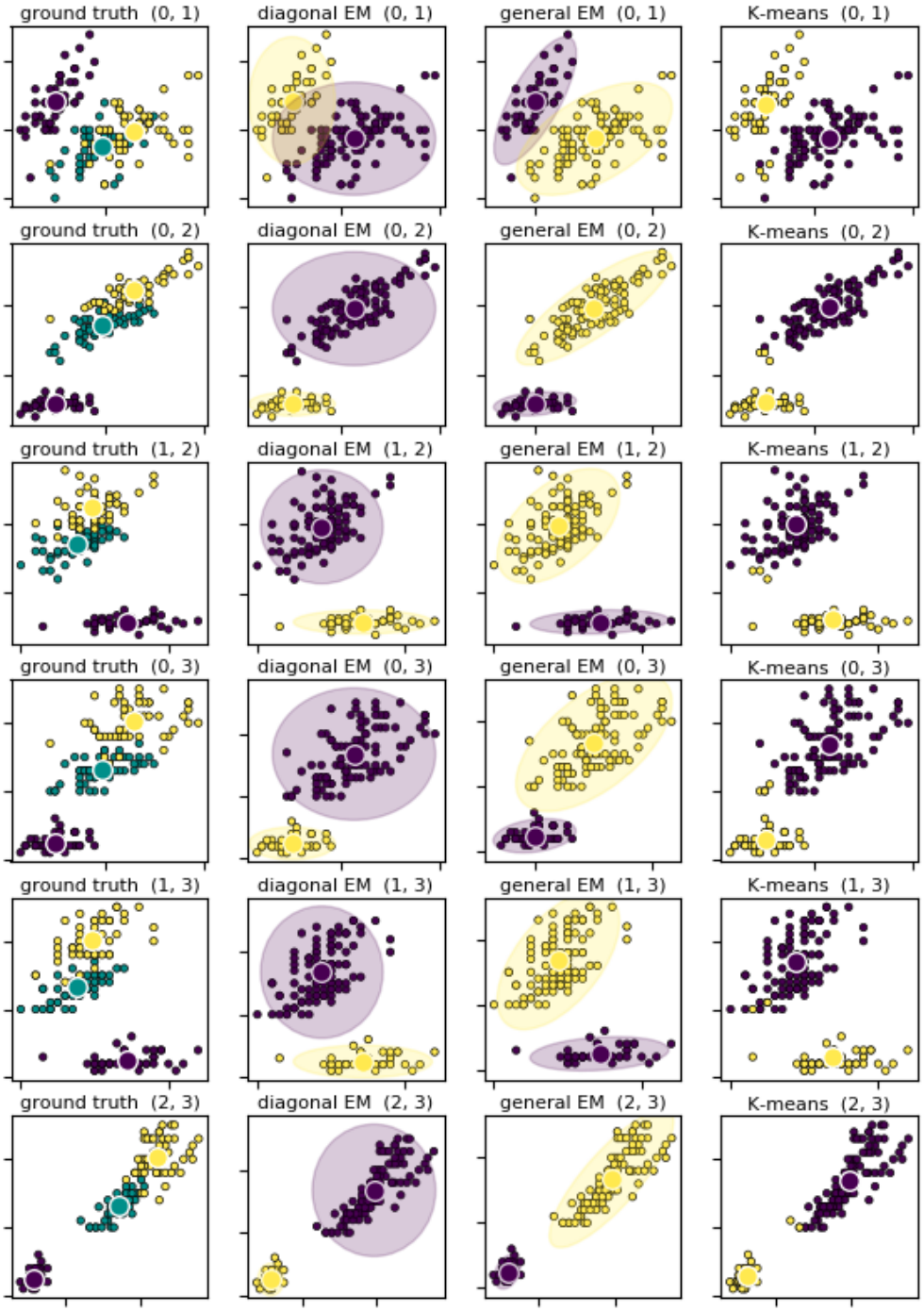

Figure 5: Results for $K = 2$

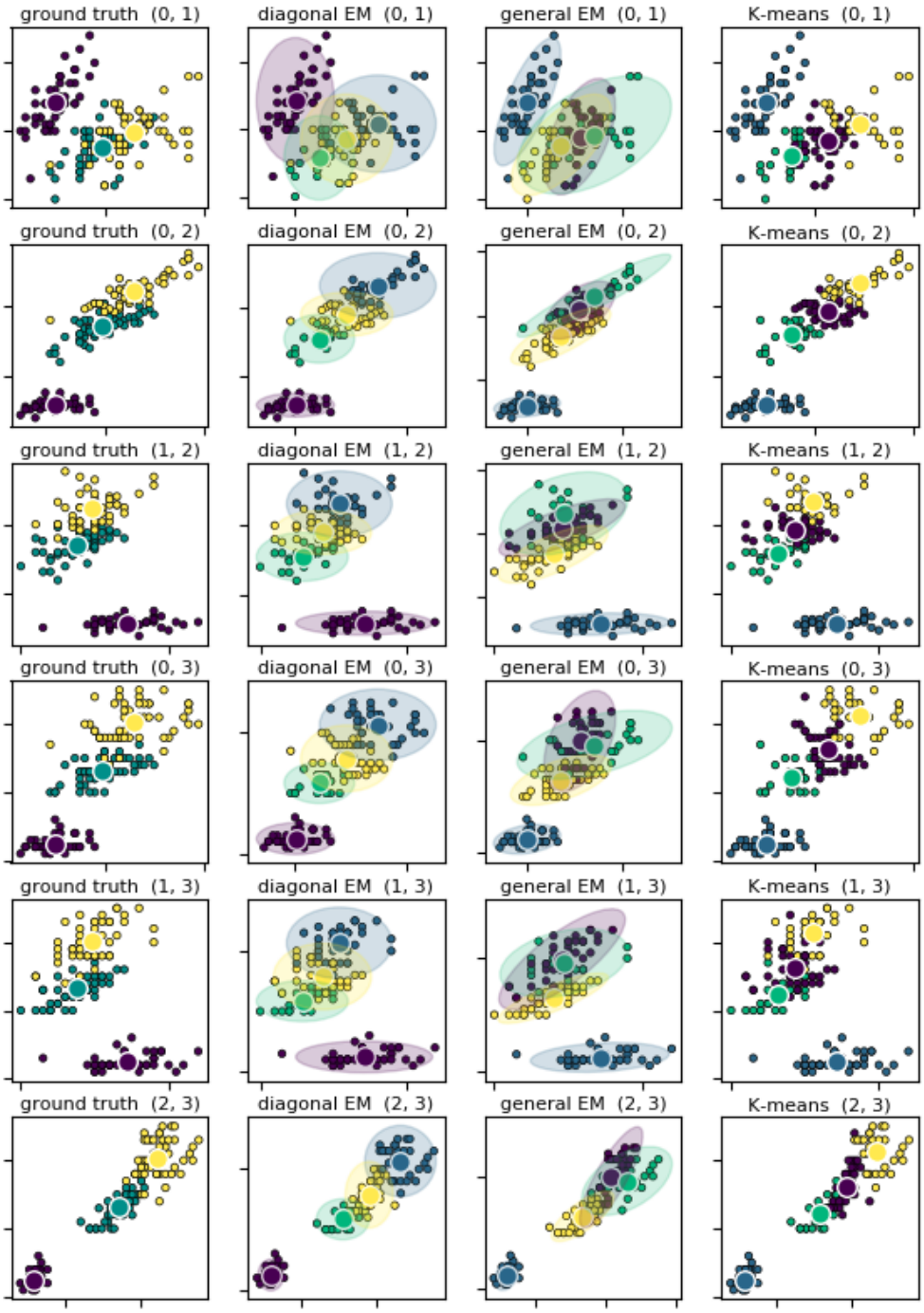

Figure 6: Results for $K = 4$