MVA - Probabilistic Graphical Models

DM2

Ariane ALIX, Hassen MIRI

December 31th, 2019

---

1 Classification: K-means, and the EM algorithm

Let's consider a mixture model, with K components, where datapoints $(X_i)_{i=1,...,n}$ have a probability p_k to be in component k: $P(Z_i = k) = p_k$, and, conditional on $Z_i = k$, $X_i \sim N(\mu_k, D_k)$, a multivariate Gaussian distribution with mean μ_k and diagonal (not full) covariance matrix D_k.

1. The Expectation-Maximization algorithm aims to compute *incomplete* the Maximum Likelihood Estimator of the datapoints $(X_i)_{i=1,...,n}$ under the parameters $(p_k, \mu_k, D_k)_{i=1,...,K}$.

The *incomplete* log-likelihood (the *complete* one would be $\log(p(X, Z); \theta)$ can be written as:

$$\begin{aligned}\mathcal{L}(X_1, ..., X_n; (p_1, \mu_1, D_1), ..., (p_K, \mu_K, D_K)) &= \sum_{i=1}^{n} \log(p(X_i; (p_1, \mu_1, D_1), ..., (p_K, \mu_K, D_K))) \\ &= \sum_{i=1}^{n} \log(\sum_{k=1}^{K} (p(X_i, Z_i = k; p_k, \mu_k, D_k)) \\ &= \sum_{i=1}^{n} \log(\sum_{k=1}^{K} p_k \, p(X_i | Z_i = k; \mu_k, D_k))\end{aligned}$$

The EM algorithm consists of two steps : the Expectation (E) where we evaluate the log-likelihood using the current estimates of the parameters $(p_k, \mu_k, D_k)^{(t)}_{i=1,...,K}$, and the Maximization (M) where we compute new estimates for the parameters that will maximize the log-likelihood found during the E step.

Let us use the notations:

- $\theta^{(t)} = (p_k, \mu_k, D_k)^{(t)}_{i=1,...,K}$
- $q^{(t)}_{ik} = p(Z_i = k | X_i; \theta^{(t)})$

Expectation step

We first compute :

$$\begin{aligned} q_{ik}^{(t)} &= \frac{p(Z_i = k)p(X_i|Z_i = k;\theta^{(t)})}{p(X_i;\theta^{(t)})} \\ &= \frac{p_k^{(t)}p(X_i|Z_i = k;\theta^{(t)})}{\sum_{l=1}^{K} p_l^{(t)}p(X_i|Z_i = l;\theta^{(t)})} \end{aligned}$$

Maximization step

We want to solve:

$$\begin{aligned} \theta^{(t+1)} =& \arg\max_{\theta} \mathbb{E}_{q^{(t)}}[\hat{\mathcal{L}}(\theta^{(t)}] \\ \text{s.t} \quad & \sum_{i=1}^{K} p_k = 1 \end{aligned}$$

With $\hat{\mathcal{L}}$ the *complete* log-likelihodd:

$$\begin{aligned} \mathbb{E}_{q^{(t)}}[\hat{\mathcal{L}}(\theta^{(t)}] &= \mathbb{E}_{q^{(t)}}[\sum_{i=1}^{n}(\log(p(X_i, Z_i;\theta^{(t)}))] \\ &= \mathbb{E}_{q^{(t)}}[\sum_{i=1}^{n}(\sum_{k=1}^{K}\log(p_k pmp(X_i|Z_i = k;\theta^{(t)}))] \\ &= \sum_{i=1}^{n}(\sum_{k=1}^{K} q_{ki}^{(t)}(\log(p_k) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|D_k| - \frac{1}{2}(X_i - \mu_k)^T D_k^{-1}(X_i - \mu_k))) \end{aligned}$$

With the constraint, we get the dual objective function :

$$\sum_{i=1}^{n}(\sum_{k=1}^{K} q_{ki}^{(t)}(\log(p_k) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|D_k| - \frac{1}{2}(X_i - \mu_k)^T D_k^{-1}(X_i - \mu_k))) + \lambda(1 - \sum_{k=1}^{K} p_k)$$

From there, using the Karush-Kuhn-Tucker conditions on the gradients:

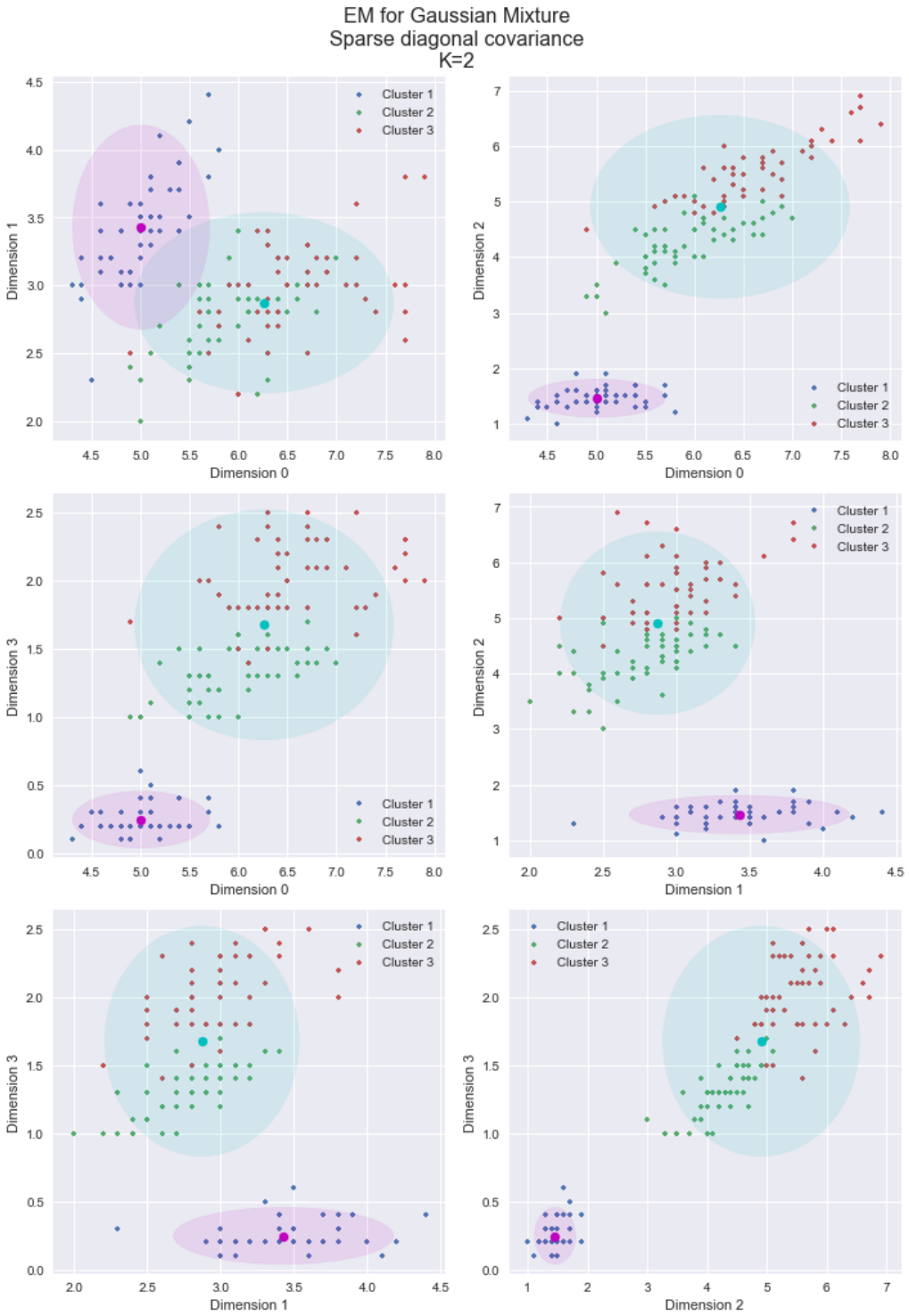
- $\forall k \quad \frac{\sum_{i=1}^{n} q_{ki}^{(t)}}{p_k} = \lambda$
- $\sum_{i=1}^{n} q_{k,i}^{(t)} D_k^{-1}(\mu_k - X_i) = 0$
- $\sum_{i=1}^{n} q_{k,i}^{(t)}(D_k^{-1} - D_k^{-2}\text{diag}(X_i - \mu_k)^2) = 0$

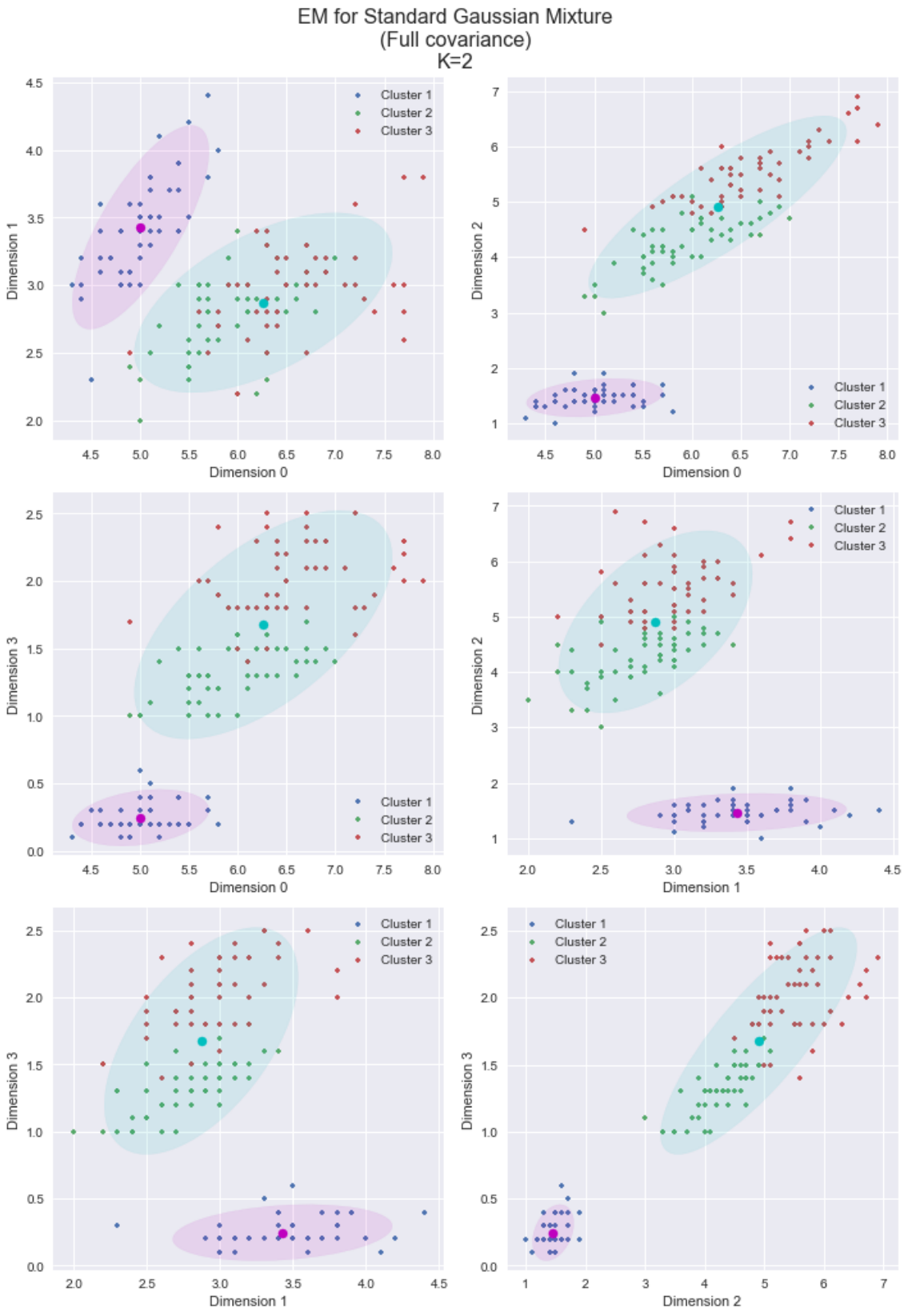
Hence:

- $p_k^{(t)} = \frac{\sum_{i=1}^{n} q_{ki}^{(t)}}{n}$
- $\mu_k^{(t)} = \frac{\sum_{i=1}^{n} q_{ki}^{(t)} X_i}{\sum_{i=1}^{n} q_{ki}^{(t)}}$
- $D_k^{(t)} = \frac{\sum_{i=1}^{n} q_{ki}^{(t)}\text{diag}(X_i - \mu_k)^2}{\sum_{i=1}^{n} q_{ki}^{(t)}}$

2. For datasets with relative independance between dimensions, not placing any constraint on the covariance terms could introduce unnecessary parameters. When some of the variables have weak correlations we could use sparse covariance matrices to characterize the components with a parsimonious representation. This implies a performance close to the one we would obtain with full covariance matrices but with a simpler model, generally more computationally efficient.

3.




EM for Standard Gaussian Mixture
(Full covariance)
K=2
Cluster 1
Cluster 2
Cluster 3
Dimension 0
Dimension 1
Dimension 2
Dimension 3

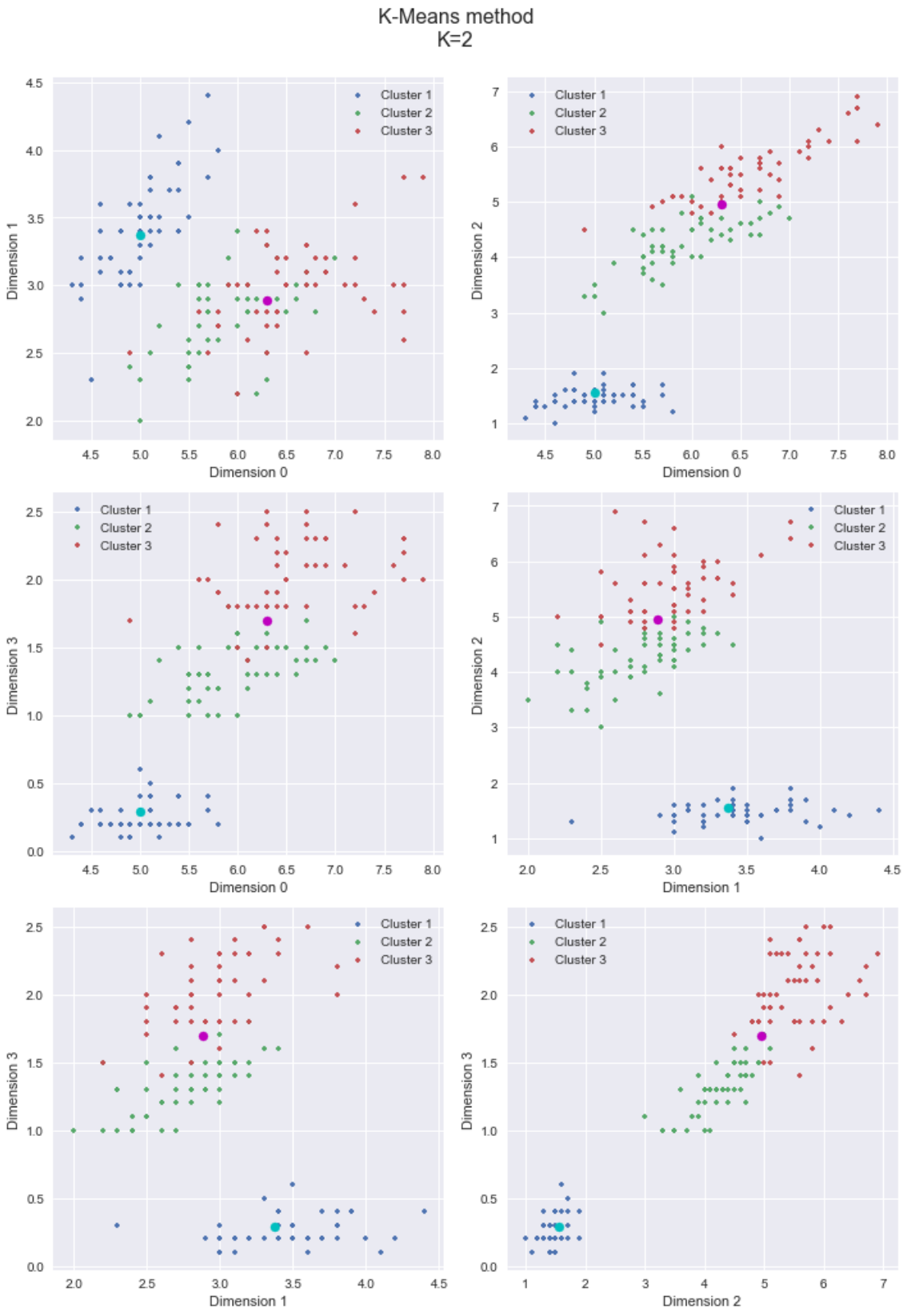
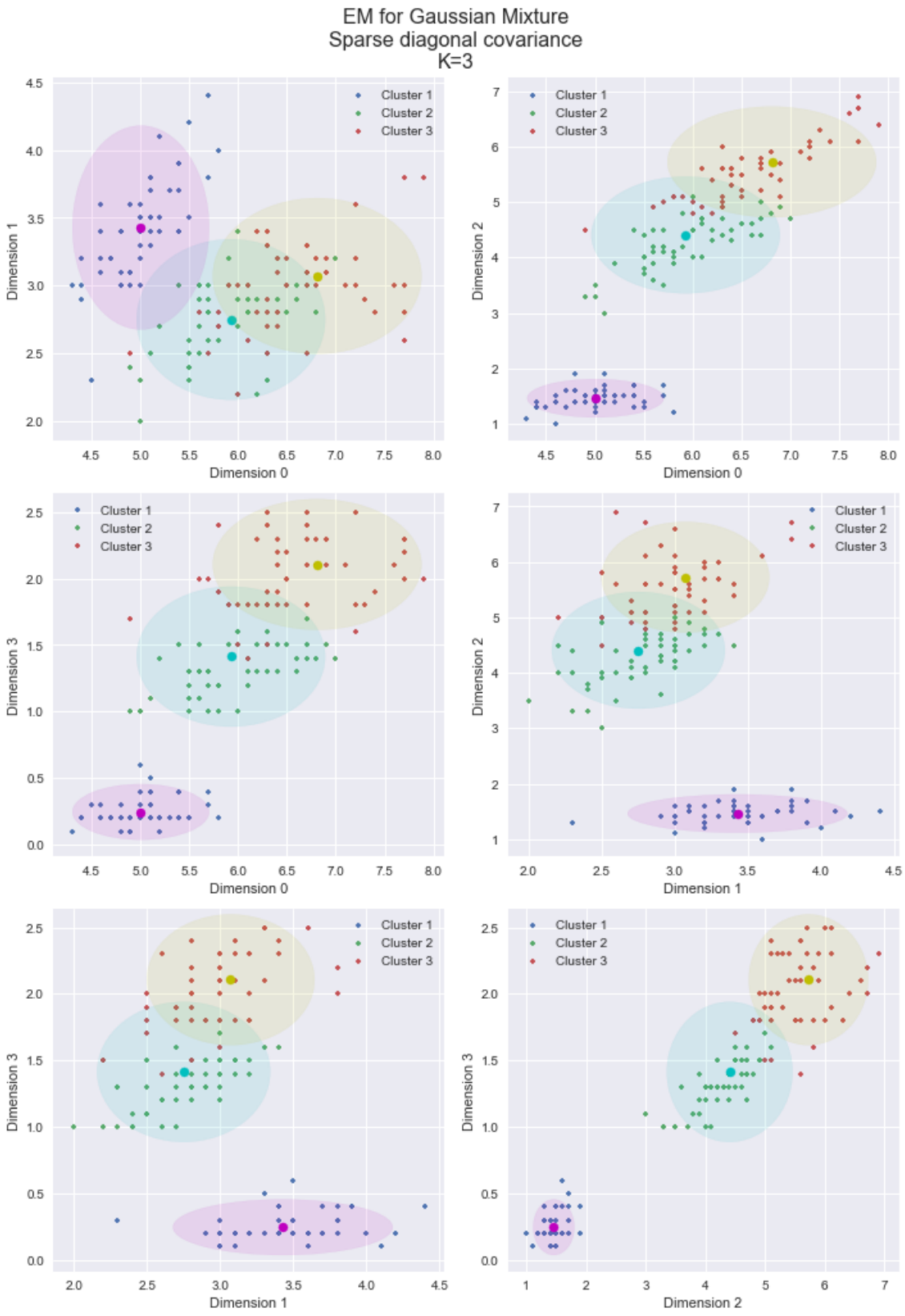
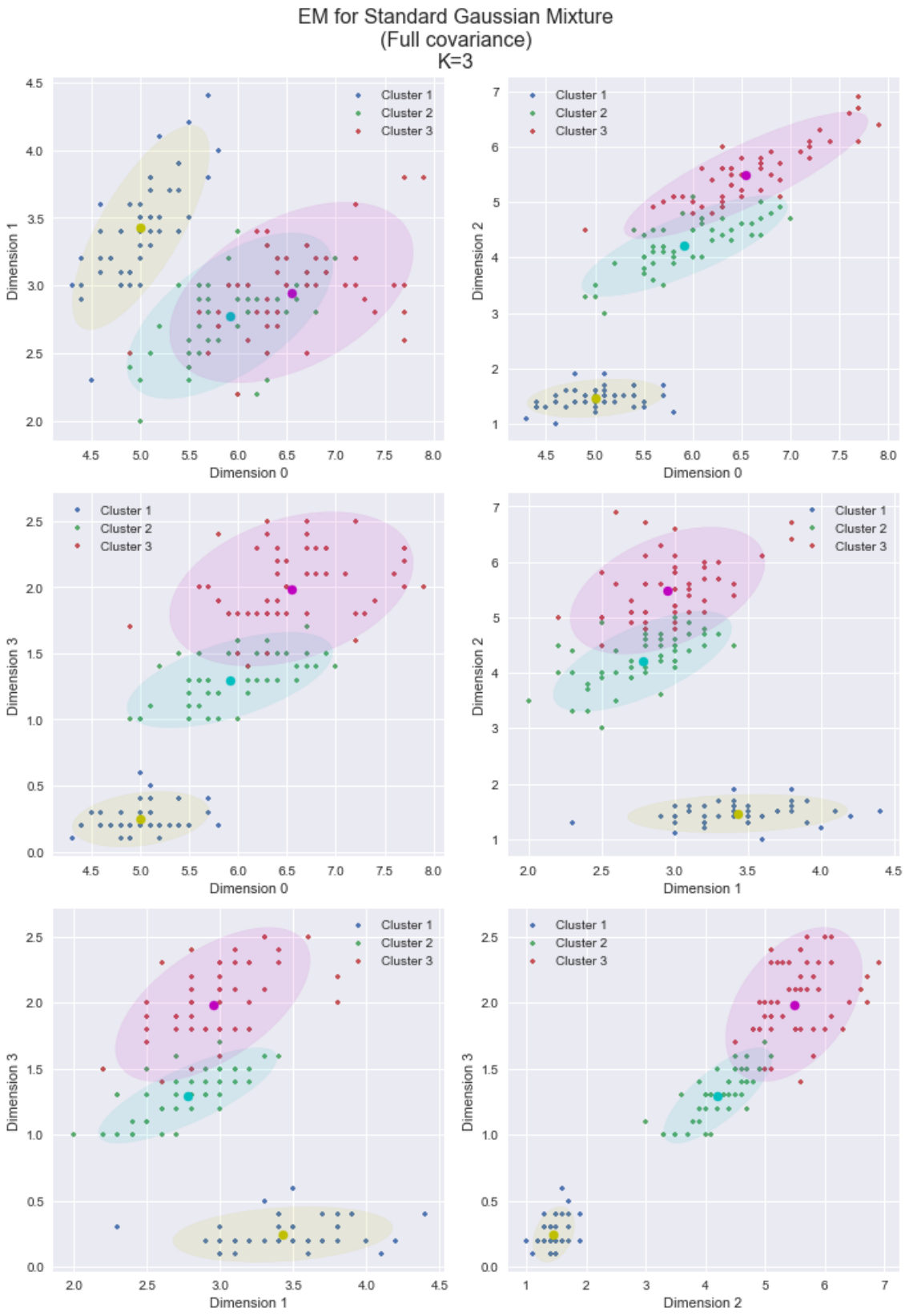


Figure 1: Centroïds computed with 3 different methods with the hypothesis of 2 clusters


EM for Gaussian Mixture
Sparse diagonal covariance
K=3
Cluster 1
Cluster 2
Cluster 3
Dimension 0
Dimension 1
Dimension 2
Dimension 3


EM for Standard Gaussian Mixture
(Full covariance)
K=3
Cluster 1
Cluster 2
Cluster 3
Dimension 0
Dimension 1
Dimension 2
Dimension 3

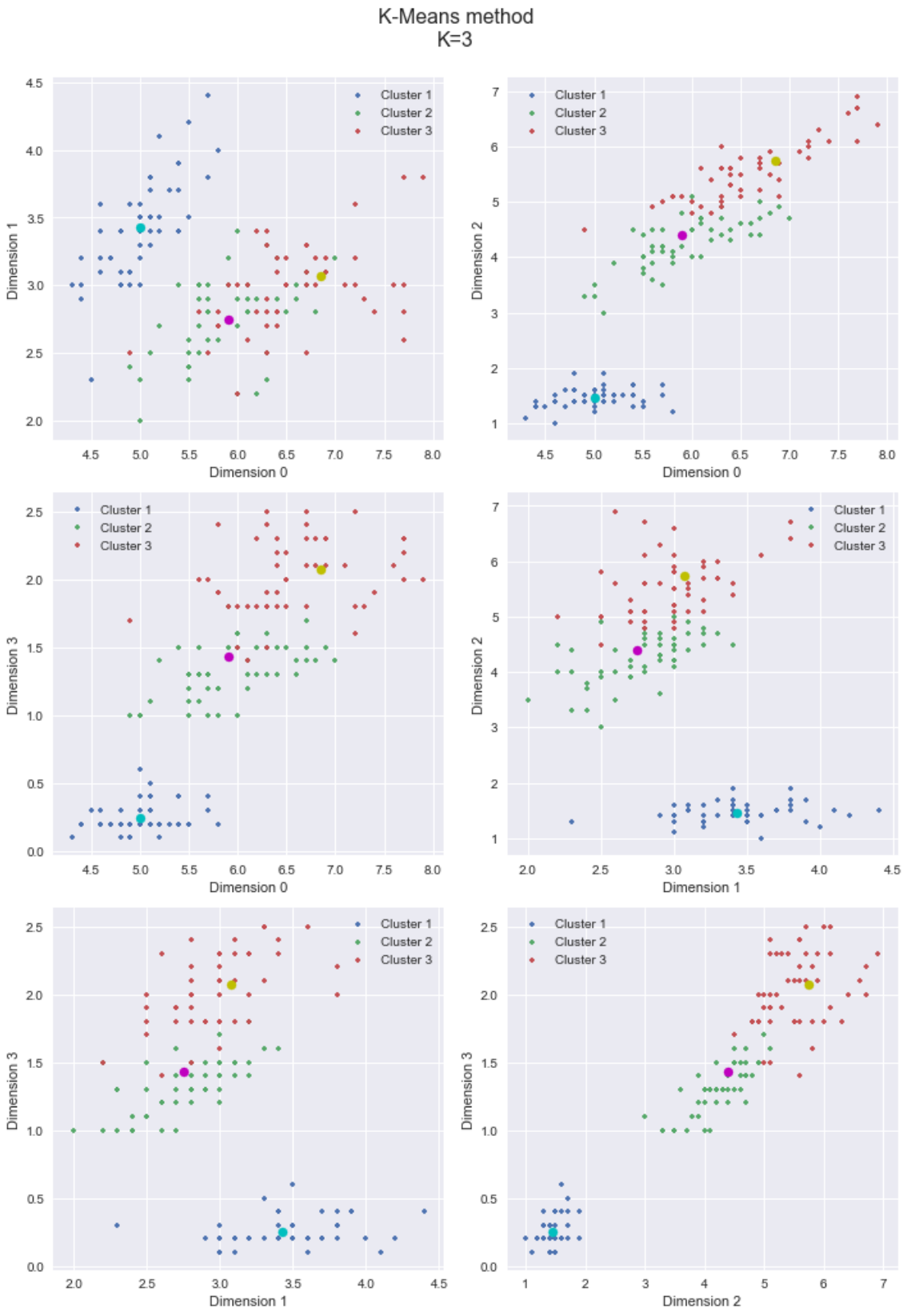
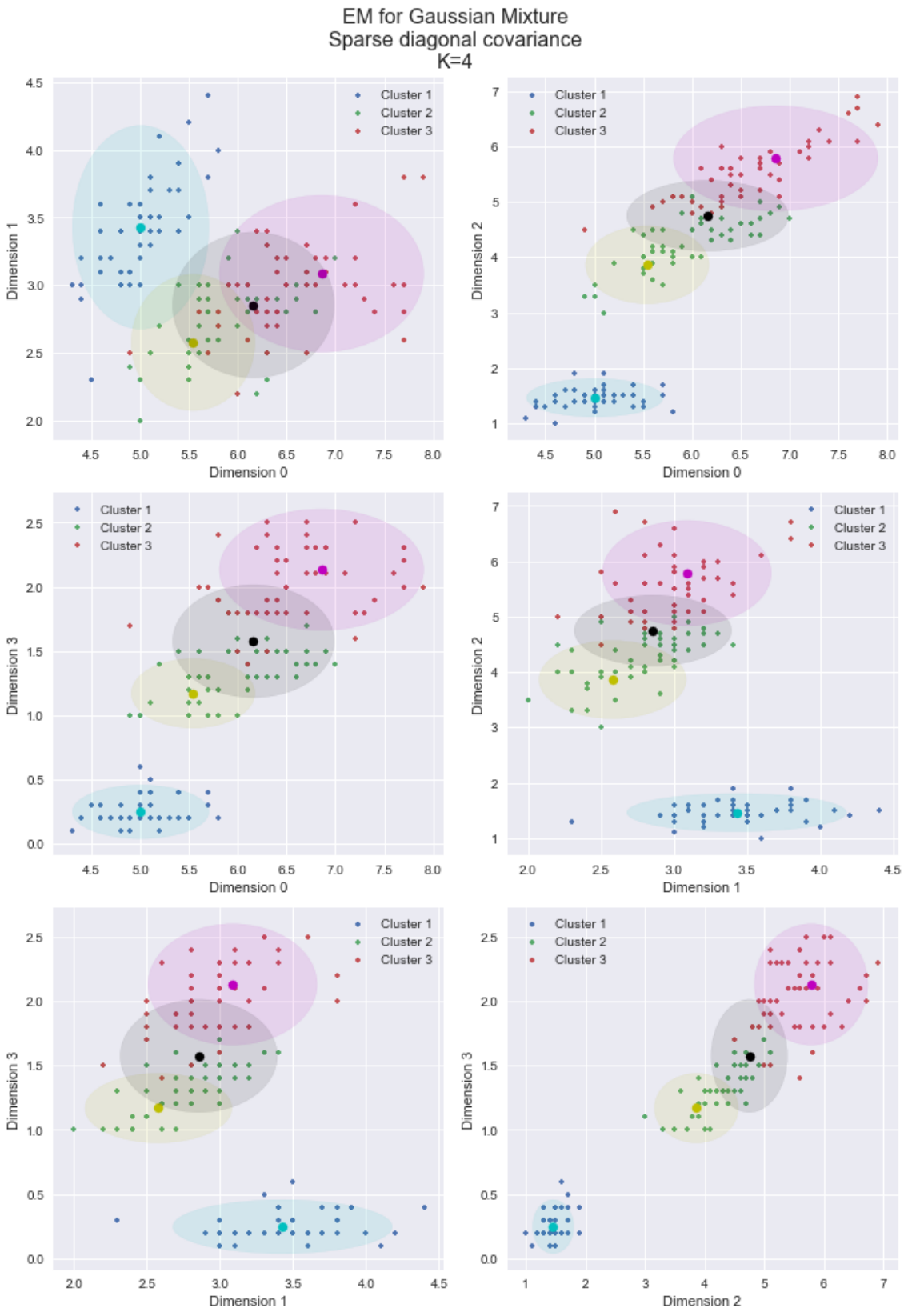
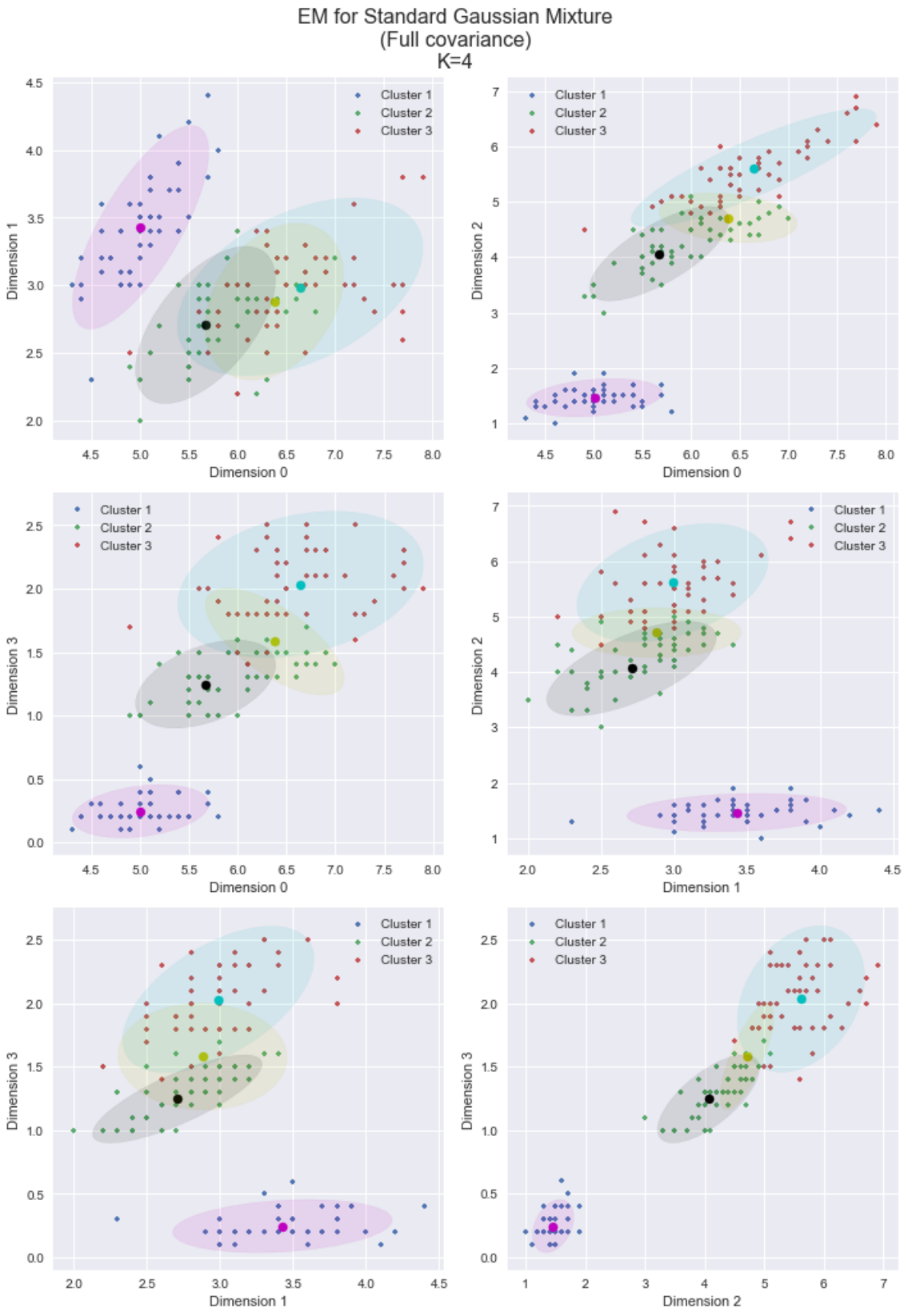


Figure 2: Centroïds computed with 3 different methods with the hypothesis of 3 clusters


EM for Gaussian Mixture
Sparse diagonal covariance
K=4
Cluster 1
Cluster 2
Cluster 3
Dimension 0
Dimension 1
Dimension 2
Dimension 3


EM for Standard Gaussian Mixture
(Full covariance)
K=4
Cluster 1
Cluster 2
Cluster 3
Dimension 0
Dimension 1
Dimension 2
Dimension 3

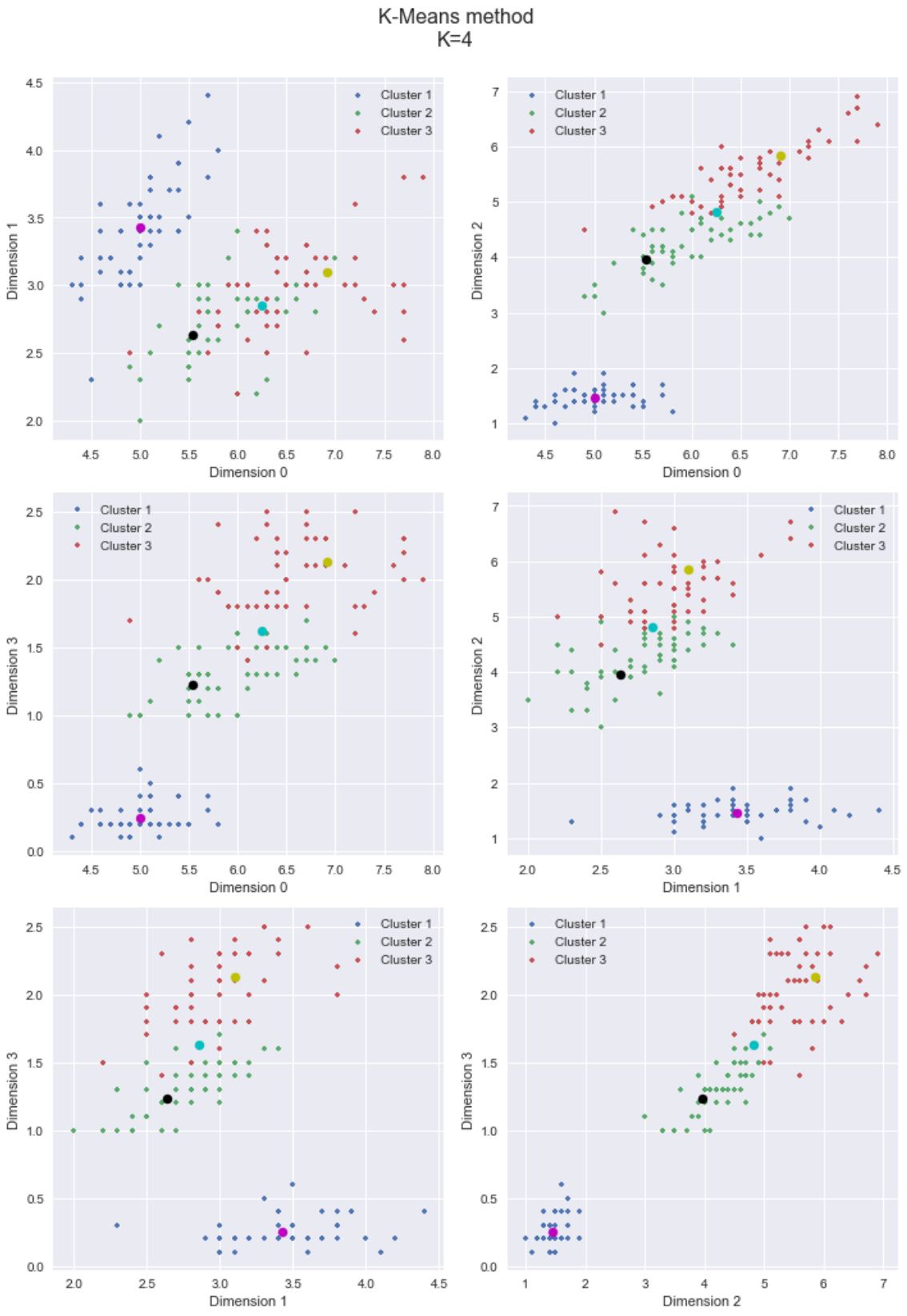


Figure 3: Centroïds computed with 3 different methods with the hypothesis of 4 clusters

We plotted the results of the clustering of the Iris dataset with 3 different methods : the Expectation-Minimization with not full diagonal covariance matrix like studied, another EM with standard covariance, and the K-Means method. Both the standard EM and K-Means were already implemented in *scikit-learn*.

In the plots, the colors of the points corresponds to their true labels, and the bigger points represent the centroïds computed by the methods. For the first two methods, ellipses representing the confidence interval (2 standard deviations) were also plotted. Since the K-Means does not return any estimation of the covariance, it could not have ellipses.

We notice that the accuracy looks better for K=3, which is in fact the true number of clusters of the Iris dataset.

4. K-Means is generally biased toward spherical clusters (isotropic) because it is based on the L2 norm and hard assignements, unlike the EM algorithm which returns the probabilities of belonging to clusters. As a consequence it would perform badly on a dataset with non-spherical clusters clos to each other.

Here below are two examples where the color of a point is associated to its predicted cluster:

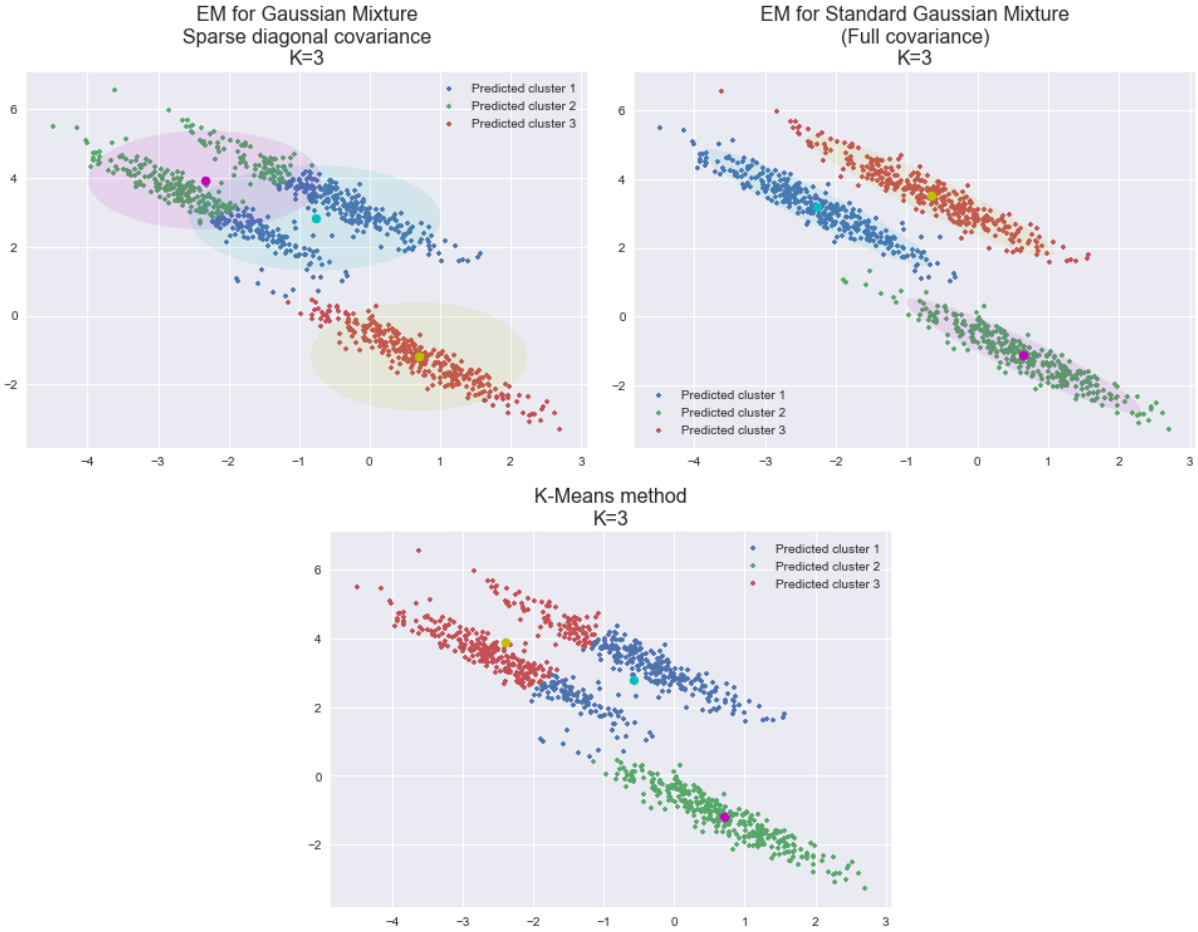


Figure 4: Example with anisotropic clusters

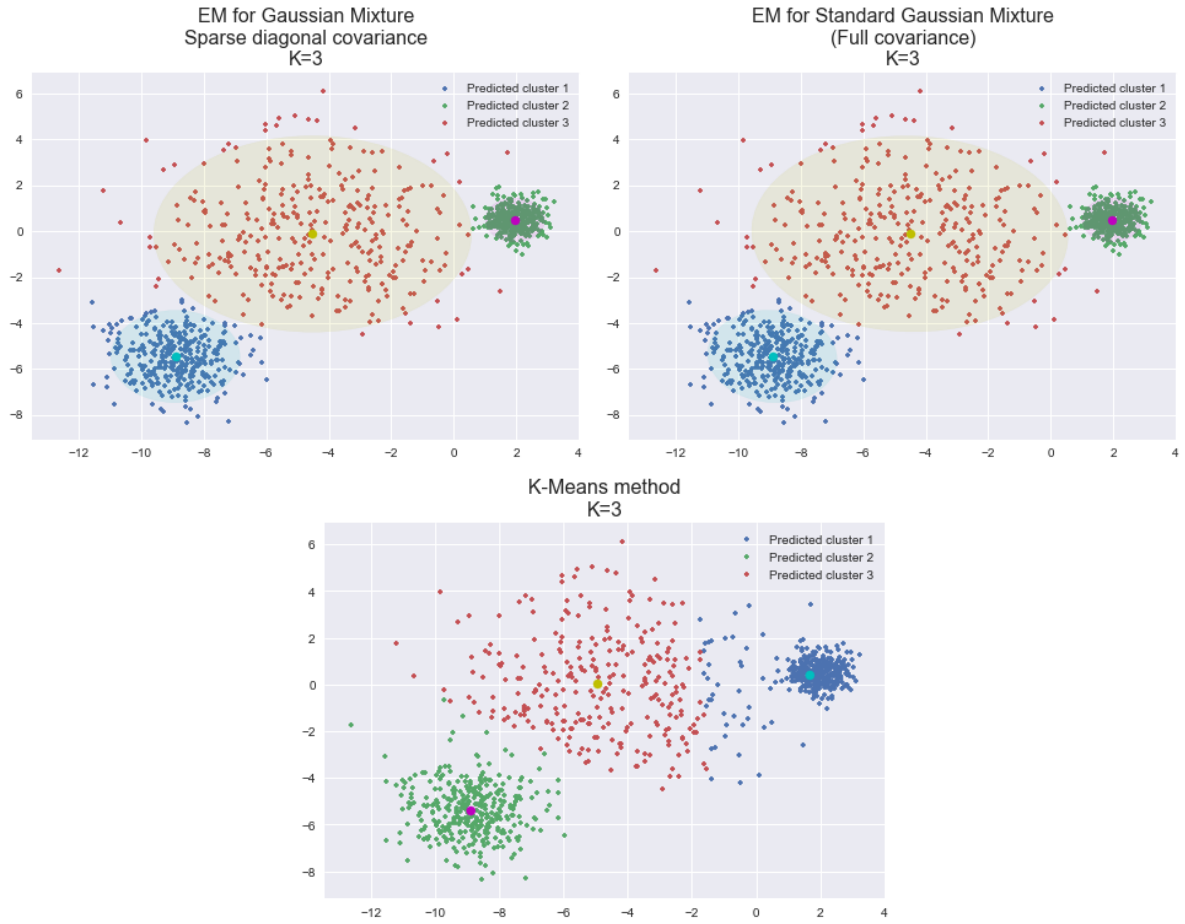


Figure 5: Example with blobs of varied size

The KMeans is globally less accurate in these cases.

We can also notice that the EM with sparse diagonal matrix does not work well in the first case : it is due to the fact that there is a strong dependance between the 2 dimensions relatively to the clusters that can not be expressed by a diagonal matrix.

2 Graphs, algorithms and Ising

1. For an undirected chain the distribution can be written as:

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

so the marginal distribuiton $p(x_i)$ for a certain i is written:

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

with the forward and backward messages calculated as:

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

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

The functions ψ_i could be represented as vectors. In the case where X_i takes discrete values in 1..K ψ_i could be represented as $[\psi_i(X_i = j, j = 1..K]$. We can generalize this to the continuous case where we sample discrete values from our continuous interval. Following the same logic, the edges functions $\psi_{i-1,i}$ would be represented as a matrix for all the possible discrete values of X_i and X_{i-1} $[[\psi_{i,i-1}(X_i = j, X_{i-1} = l, j = 1..K], l = 1..K']$

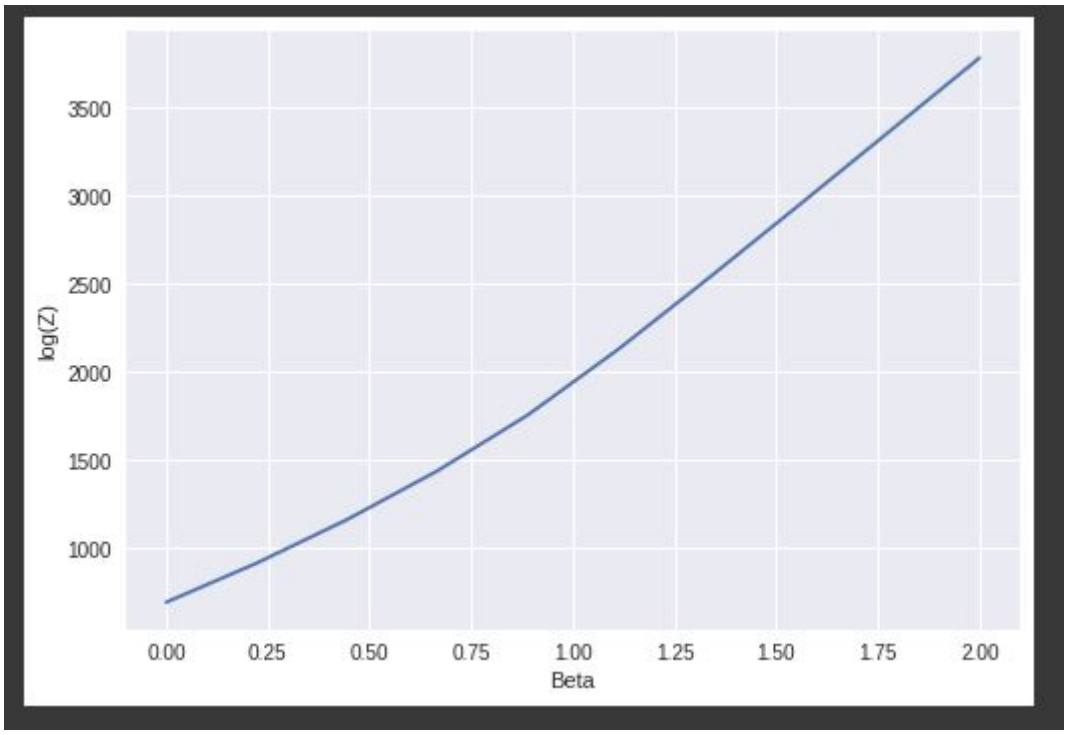
2. One way of converting the Ising model into an undirected chain model is to construct supernodes on one dimension of the model. Since in our case the width is smaller than the height, we will consider a junction tree that assembles all the nodes on one line of height into a super node. The supernode will contain 2^{10} possible states.
With this model the probability factorizes as:

$$p(x) = \frac{1}{Z} \prod_{h_i} \psi_{h_i}(x_{h_i}) \prod_{h_i} \psi_{h_i,h_i+1}(x_{h_i}, x_{h_i+1})$$

with

$$\psi_i(x_{h_i}) = \prod_{k=1}^{w} e^{\alpha x_{(i,k)}} \prod_{k=1}^{w-1} e^{\beta \mathbb{1}(x_{(i,k)} = x_{(i,k+1)})}$$

$$\psi_{h_i,h_i+1}(x_{h_i}, x_{h_i+1}) = \prod_{k=1}^{w} e^{\beta \mathbb{1}(x_{(i,k)} = x_{(i+1,k)})}$$


3500
3000
2500
2000
1500
1000
log(Z)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Beta

3.