Probabilistic Graphical Models
Homework 2

Antonin Berthon

November 2018

1 Exercise 1.1

A joint distribution $p(X) \in \mathcal{L}(G)$ satisfies the following factorization :

$$p(x, y, z, t) = p(x)p(y)p(z|x, y)p(t|z)$$

X and Y are not blocked by T at Z since Z is not in T, (X, Z, Y) is a v-structure but T is a descendant of Z. Therefore we do not have $X \perp\!\!\!\perp Y | T$ for any $p \in \mathcal{L}(G)$.
For example, take:

1. X, Y are two dices ranging from 1 to 6
2. $Z = X + Y$
3. $T = Z \leq 7$

In this case X and Y are clearly not independent when T is observed.

2 Exercise 1.2

a) Let X, Y, Z such that $Z \in \{0, 1\}$, $X \perp\!\!\!\perp Y | Z$ and $X \perp\!\!\!\perp Y$. Since $X \perp\!\!\!\perp Y$, we have:

$$\begin{aligned}
p(x, y) = p(x)p(y) &= \Big(p(x|z=0)p(z=0) + p(x|z=1)p(z=1)\Big)\Big(p(y|z=0)p(z=0) + p(y|z=1)p(z=1)\Big) \\
&= \Big(p(z=0)(p(x|z=0) - p(x|z=1)) + p(x|z=1)\Big)\Big(p(z=1)(p(y|z=1) - p(y|z=0)) + p(y|z=0)\Big) \\
&= A + p(x|z=1)p(y|z=0)\underbrace{(1 - p(z=1) - p(z=0))}_{=0} + p(z=1)p(x|z=1)p(y|z=1) \\
&\qquad + p(z=0)p(x|z=0)p(y|z=0) \\
&= A + p(x, y)
\end{aligned}$$

with

$$A = p(z=0)p(z=1)[p(x|z=0) - p(x|z=1)][p(y|z=1) - p(y|z=0)]$$

and the last equality coming from $X \perp\!\!\!\perp Y | Z$
Therefore, $A = 0$ which is equivalent to $p(x|z=0) = p(x|z=1)$ *or* $p(y|z=1) = p(y|z=0)$, *i.e.* $X \perp\!\!\!\perp Z$ or $Y \perp\!\!\!\perp Z$.

b) This statement is false in general. Let's consider $Z \in \{1, 2, 3\}$, and X, Y such that $X \perp\!\!\!\perp Y | Z$ and $X \perp\!\!\!\perp Y$. With computations similar (but slightly more complex) to the binary case, we find :

$$\begin{aligned}
0 &= p(z=0)p(z=1)[(p(x|z=0) - p(x|z=1))(p(y|z=1) - p(y|z=0))] \\
&\quad + p(z=1)p(z=2)[(p(x|z=1) - p(x|z=2))(p(y|z=2) - p(y|z=1))] \\
&\quad + p(z=2)p(z=0)[(p(x|z=2) - p(x|z=0))(p(y|z=0) - p(y|z=2))]
\end{aligned}$$

This equation can be satisfied with random variables X and Y that are not necessarily independent from Z. For instance, for $i = 0, 1, 2$, take $p(z = i) = \frac{1}{3}$; $y|z = i \sim \mathcal{B}(\pi_i)$ with $\pi_0 = \frac{1}{4}, \pi_1 = \frac{3}{4}, \pi_2 = \frac{1}{2}$; $x|z = i \sim \mathcal{B}(\mu_i)$ with $\mu_0 = \frac{1}{6}, \mu_1 = \frac{1}{6}, \mu_2 = \frac{2}{3}$. X and Y are clearly independant once Z is observed and neither of them is independent from Z.
However, $p(x, y) = \sum_z p(z)p(x, y|z) = \frac{1}{3}[\frac{1}{4}\frac{1}{6} + \frac{3}{4}\frac{1}{6} + \frac{1}{2}\frac{2}{3}] = \frac{1}{6}$, $p(x) = \frac{1}{3}$ and $p(y) = \frac{1}{3}(1 + \frac{1}{2}) = \frac{1}{2}$. We find that $p(x, y) = p(x)p(y)$, so $X \perp\!\!\!\perp Y$: we have found a counter example to the statement.

3 Exercise 2.1

Let $p \in \mathcal{L}(G)$ and $i, j \in V$ s.t. $i \to j \in E$ is a covered edge. We have:

$$p(x) = \prod_{k=1}^{n} p(x_k, x_{\pi_k}) = \left(\prod_{\substack{k=1 \\ k \neq i,j}}^{n} p(x_k, x_{\pi_k}) \right) p(x_i|x_{\pi_i}) p(x_j|x_i, x_{\pi_i})$$

However, the chain rule gives $p(x_{\pi_i})p(x_i|x_{\pi_i})p(x_j|x_i, x_{\pi_i}) = p(x_i, x_j, x_{\pi_i}) = p(x_{\pi_i})p(x_j|x_{\pi_i})p(x_i|x_j, x_{\pi_i})$.
Therefore, $p(x_i|x_{\pi_i})p(x_j|x_i, x_{\pi_i}) = p(x_j|x_{\pi_i})p(x_i|x_j, x_{\pi_i})$
Hence,

$$p(x) = \left(\prod_{\substack{k=1 \\ k \neq i,j}}^{n} p(x_k, x_{\pi_k}) \right) p(x_j|x_{\pi_i}) p(x_i|x_j, x_{\pi_i})$$

which proves that $p \in \mathcal{L}(G')$ and so $\mathcal{L}(G) \subset \mathcal{L}(G')$.
The same reasoning can be applied with $p \in \mathcal{L}(G')$ to find $\mathcal{L}(G') \subset \mathcal{L}(G)$. Hence, $\boxed{\mathcal{L}(G') = \mathcal{L}(G)}$.

4 Exercise 2.2

Since G is a directed tree, we know that it contains a single root node that do not have any parent (without loss of generality we will assume in the following that this root node is indexed by 1 : therefore $\pi_1 = \emptyset$ and we can write $f_1(x_1, x_{\pi_1}) = f_1(x_1)$). Also, aside from the root node, all of its nodes have exactly one parent.
Let $p \in \mathcal{L}(G)$. This means that there exists functions $(f_i)_i$ verifying $\forall i, f_i \geq 0$ and $\sum\limits_{x_i} f_i(x_i, x_{\pi_i}) = 1$, such that: $p(x) = \prod_{i=1}^{n} f_i(x_i, x_{\pi_i})$.
Let Φ be an application from $[2, n]$ to $V \times V$, $\Phi\colon i \longmapsto (i, \pi_i)$. Since all node expect the root have exactly one parent, the set of maximum cliques of G is $\mathcal{C}_{max} = \{\{i, \pi_i\},\ i \in V \backslash \{1\}\}$. Therefore, Φ is a bijection between $[2, n]$ and $\mathcal{C}_{max}$. Furthermore, let $\mathcal{C}_{min} = \{\{i\}, i \in V\}$. The full set of cliques $\mathcal{C}$ verifies $\mathcal{C} = \mathcal{C}_{min} \cup \mathcal{C}_{max}$ with $\mathcal{C}_{min} \cap \mathcal{C}_{max} = \emptyset$.
Therefore, we can write:

$$p(x) = f_1(x_1) \prod_{i \in [2,n]} f_i(x_i, x_{\pi_i}) \quad = f_1(x_1) \prod_{c \in \mathcal{C}_{max}} f_{\Phi^{-1}(c)}(x_{\Phi^{-1}(c)}, x_{\pi_{\Phi^{-1}(c)}}) = \prod_{c \in \mathcal{C}_{min}} \psi_c(x_c) \prod_{c \in \mathcal{C}_{max}} \psi_c(x_c)$$

where $\forall c \in \mathcal{C}_{max},\ x_c = (x_{\Phi^{-1}(c)}, x_{\pi_{\Phi^{-1}(c)}}),\ \psi_c = f_{\Phi^{-1}(c)}$ and $\forall c \in \mathcal{C}_{min},\ \psi_c(x_c) = \begin{cases} f_1(x_1) & \text{if } c = \{1\} \\ 1 & \text{otherwise} \end{cases}$.
Note that we do not need a normalization term $\frac{1}{Z}$ since $\sum\limits_{x'} p(x') = 1$ assures that $\sum\limits_{x'} \prod\limits_{c \in \mathcal{C}} \psi_c(x'_c) = 1$.

Therefore we have $\boxed{\mathcal{L}(G) \subset \mathcal{L}(G')}$.
Let $p \in \mathcal{L}(G')$: there exists potentials $\psi_c \geq 0$ for each $c \in \mathcal{C}$ such that $p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c)$.
With the same $\mathcal{C}_{min}$, $\mathcal{C}_{max}$ and Φ as previously we have:

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}_{min}} \psi_c(x_c) \prod_{c \in \mathcal{C}_{max}} \psi_c(x_c) = \frac{1}{Z} \prod_{i=1}^{n} \widetilde{f}_i(x_i, x_{\pi_i})$$

where $\widetilde{f}_i(x_i, x_{\pi_i}) = \begin{cases} \psi_{\{1\}}(x_{\{1\}}) & \text{if } i = 1 \\ \psi_{\{i\}}(x_{\{i\}}) \psi_{\Phi(i)}(x_{\Phi(i)}) & \text{otherwise} \end{cases}$
Moreover,

$$Z = \sum_{x'} \prod_{c \in \mathcal{C}} \psi_(x'_c) = \sum_{x'} \prod_{i=1}^{n} \widetilde{f}_i(x'_i, x_{\pi_i}) = \prod_{i=1}^{n} \sum_{x'_i} \widetilde{f}_i(x'_i, x_{\pi_i})$$

Therefore, by setting $\forall i \in [1, n],\ f_i(x_i, x_{\pi_i}) = \frac{\widetilde{f}_i(x_i, x_{\pi_i})}{\sum\limits_{x'_i} \widetilde{f}_i(x'_i, x_{\pi_i})}$, we have $p(x) = \prod_{i=1}^{n} f_i(x_i, x_{\pi_i})$ with $\forall i, f_i \geq 0$ and $\sum\limits_{x_i} f_i(x_i, x_{\pi_i}) = 1$. Hence, $\boxed{\mathcal{L}(G') \subset \mathcal{L}(G)}$.

5 Exercice 3.a

We initialize the K-mean algorithm by picking at random K points of the dataset and setting them as the initial centers. Depending on the number of clusters K chosen, the solution given by the algorithm will be more or less consistent with the random initializations. To assess this consistency for different K, we run the algorithm several times and we compute consistency metrics like the variance of the total distortions or the consistency of the positions of the centers. Those metrics are detailed in the Appendix.

It appears that the solution given by the algorithm is very consistent when the number of cluster is properly chosen (here $K = 4$), and that running the algorithm for an unsuitable K results in inconsistent solutions. On this dataset it appears that $K = 2$ also gives consistent solutions, since it can evenly divide the 4 clusters.

6 Exercice 3.b

Suppose that $z \in \{1; k\}, x|z = j \sim \mathcal{N}(\mu_j, \sigma_j^2 I_d)$. For this model, the M-step takes the form:

$$\forall t \geq 0, \forall j \in [1, k], \quad \boxed{\pi_{j,t+1} = \frac{\sum\limits_{i=1}^{n} \tau_i^j}{n}} \; \boxed{\mu_{j,t+1} = \frac{\sum\limits_{i=1}^{n} \tau_i^j x_i}{\sum\limits_{i=1}^{n} \tau_i^j}} \; \boxed{\sigma_{j,t+1}^2 = \frac{\sum\limits_{i=1}^{n} \tau_i^j (x_i - \mu_{j,t+1})^T (x_i - \mu_{j,t+1})}{\sum\limits_{i=1}^{n} \tau_i^j}}$$

with

$$\boxed{\tau_i^j = p_\theta(z_i = j|x_i) = \frac{\pi_j \sigma_j^{-d} \exp(-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^T (x_i - \mu_j))}{\sum\limits_{j'=1}^{k} \pi_{j'} \sigma_{j'}^{-d} \exp(-\frac{1}{2\sigma_{j'}^2}(x_i - \mu_{j'})^T (x_i - \mu_{j'}))}}$$

Computations in Appendix.

7 Exercice 3.c

We now release the hypothesis that the covariance matrices are proportional to the identity :
$z \in \{1; k\}, x|z = j \sim \mathcal{N}(\mu_j, \Sigma_j)$.
The M-step keeps the same form for τ, π and μ, but we the covariance matrix is now :

$$\boxed{\Sigma_{j,t+1} = \frac{\sum\limits_{i=1}^{n} \tau_i^j (x_i - \mu_{j,t+1})(x_i - \mu_{j,t+1})^T}{\sum\limits_{i=1}^{n} \tau_i^j}}$$

8 Exercice 3.d

- The algorithm with an anisotropic covariance matrix achieves a higher log-likelihood than the isotropic one on both the training and test data. This is not surprising considering the anisotropic distributions of the clusters.
- The anisotropic algorithm performs better on the training data than on the test data, which could indicate some overfitting.
- The isotropic algorithms performs a little bit better on the test data, which indicates that no overfitting is happening. This is not surprising considering the little flexibility allowed by its underlying model.

Log-likelihood		
Model	Train	Test
Isotropic	-3994	-3986
Anisotropic	-3511	-3614

9 Figures

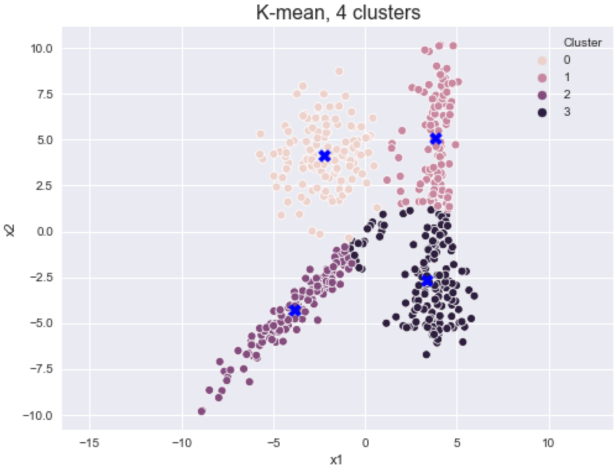
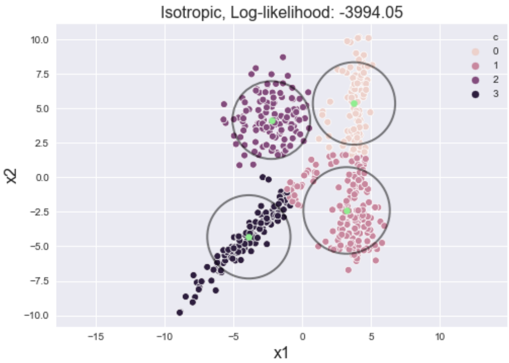
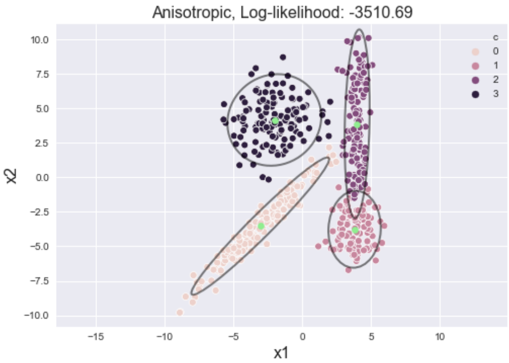


Figure 1: Clustering of the training data with the K-mean algorithm with $K = 4$. The blue crosses represent the center of the clusters and the points are grouped by their closest center with respect to the norm L2.



Clustering of the training data using the EM algorithm with isotropic covariance matrices. The crosses are the centers of the clusters, the circles show the 95% confidence interval of each cluster. The data points are assigned to the cluster for which they have the highest likelihood.



Clustering of the training data using the EM algorithm with general covariance matrices. The crosses are the centers of the clusters, the ellipses show the 95% confidence interval of each cluster. The data points are assigned to the cluster for which they have the highest likelihood.

10 Appendix

10.1 Exercice 3.a

To assess the consistency of the solution given by the K-mean algorithm, we compute two metrics:

- The variance of the distortions: for each solution, we compute the total distortion $J(\mu, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k \|x_i - \mu_k\|^2$ were $z_i^k = 1$ if x_i is in cluster k and 0 otherwise. We then form the variance of those distortions accross various execution to assess how much the quality of the solution varies at each execution.
- To assess how the positions of the cluster centers vary between different solutions, we run a K-mean algorithm on the dataset formed by the centers of each solution and compute the total distortion of those clusters.

Figure 2 shows those metrics after 20 executions for $K \in \{2, 7\}$.

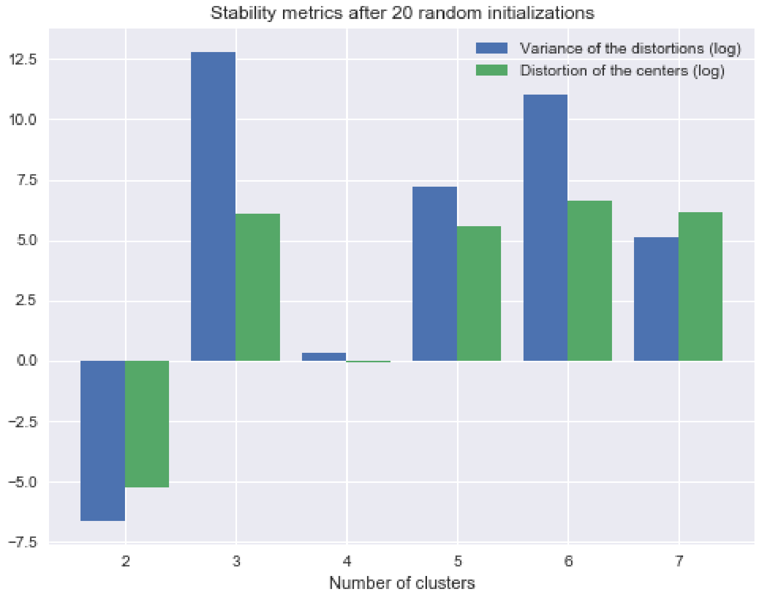


Figure 2: Logarithms of the variance of the distortion (in blue) and the distortions of the centers (in green) for $K = 2$ to 7. The algorithm seems much more consistent for $K = 2$ or $K = 4$ compared to other numbers of clusters.

10.2 Exercice 3.b

Suppose $z \in \{1; k\}, x|z = j \sim \mathcal{N}(\mu_j, \sigma_j^2 I_d)$. For given parameters $\theta = (\pi, \mu, \sigma)$, τ_i^j is given by:

$$\begin{aligned}
\tau_i^j = p_\theta(z_i = j|x_i) &= \frac{p_\theta(x_i|z_i = j)p(z_i = j)}{\sum\limits_{j'=1}^{k} p(x_i|z_i = j')} \\
&= \frac{\mathcal{N}(x_i|\mu_j, \sigma_j I_d)\pi_j}{\sum\limits_{j'=1}^{k} \mathcal{N}(x_i|\mu_{j'}, \sigma_{j'} I_d)\pi_{j'}} \\
&= \frac{\pi_j \sigma_j^{-d} \exp(-\frac{1}{2\sigma_j^2}(x_i - \mu_j)^T(x_i - \mu_j))}{\sum\limits_{j'=1}^{k} \pi_{j'} \sigma_{j'}^{-d} \exp(-\frac{1}{2\sigma_{j'}^2}(x_i - \mu_{j'})^T(x_i - \mu_{j'}))}
\end{aligned}$$

At the t-th iteration of the algorithm, the complete log-likelihood of the problem for parameters θ_t is

$$\begin{aligned}
\log p_{\theta_t}(x, z) &= \sum_{i=1} n \log p_{\theta_t}(x_i, z_i) \\
&= \sum_{i=1}^{n} p_{\theta_t}(z_i) + p_{\theta_t}(x_i|z_i) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} z_j^i \log \pi_{j,t} + \sum_{i=1}^{n}\sum_{j=1}^{k} z_j^i \log \mathcal{N}(x_i|\mu_{j,t}, \sigma_{j,t}^2 I_d)
\end{aligned}$$

with $z_j^i = \begin{cases} 1 & \text{if } z^i = j \\ 0 & \text{otherwise} \end{cases}$

During the E-step, we form the expectation of the complete likelihood with respect to the distribution of z given x. The terms $\log \pi_{j,t}$ and $\mathcal{N}(x_i|\mu_{j,t}, \sigma_{j,t}^2 I_d)$ are constant in the distribution of $z|x$, so we only have to apply the expectation $\mathbb{E}_{z|x}$ on the terms z_j^i : $\mathbb{E}_{z|x}(z_j^i) = p_{\theta_t}(z_i = j|x_i) = \tau_i^j(\theta_t)$.
Therefore, the M-step is computed by maximizing :

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \tau_i^j \log \pi_{j,t} + \sum_{i=1}^{n}\sum_{j=1}^{k} \tau_i^j \Big(-\frac{d}{2}\log((2\pi\sigma_{j,t}^2) - \frac{1}{2\sigma_{j,t}^2}(x - \mu_{j,t})^T(x - \mu_{j,t})\Big)$$

- To compute $\pi_{j,t+1}$, we solve the following optimization problem:

$$\begin{aligned}
\max_{\pi} \quad & \sum_{i=1}^{n}\sum_{j=1}^{k} \tau_i^j \log \pi_j \\
\text{s.t.} \quad & \sum_{j=1}^{k} \pi_j = 1
\end{aligned}$$

Since the objective function is concave, we can solve this problem by setting 0 the gradient of its Lagrangian, which takes the form : $\mathcal{L}(\pi, \lambda) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{k} \tau_i^j \log(\pi_j) - \lambda\big(\sum\limits_{j=1}^{k} \pi_j - 1\big)$ with $\lambda \in \mathbb{R}$.
We have

$$\mathcal{L}(\pi, \lambda) = 0 \Longleftrightarrow \forall j, \sum_{i=1}^{n} \frac{\tau_i^j}{\pi_j} - \lambda = 0 \Longleftrightarrow \forall j, \pi_j = \frac{1}{\lambda}\sum_{i=1}^{n} \tau_i^j$$

Since $\sum\limits_{j=1}^{k}\sum\limits_{i=1}^{n} \tau_i^j = n$, the constraint $\sum\limits_{j=1}^{k} \pi_j = 1$ gives $\lambda = n$, and therefore:

$$\forall j \in [1, k], \boxed{\pi_j^* = \frac{\sum\limits_{i=1}^{n} \tau_i^j}{n}}$$

- To compute $\mu_{j,t+1}$, we solve the following optimization problem:

$$\max_{\mu} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}\tau_i^j\big(-\frac{1}{2\sigma_{j,t}^2}(x-\mu_j)^T(x-\mu_j)\big)$$

Once again, the objective function is a concave function, so we can find μ by setting its gradient to 0:

$$\begin{aligned}\forall j, -\frac{1}{\sigma_{j,t}^2}\sum_{i=1}^{n}\tau_i^j(x_i-\mu_{j,t})=0 &\Longrightarrow \forall j, \sum_{i=1}^{n}\tau_i^j x_i = \sum_{i=1}^{n}\tau_i^j\mu_j \\ &\Longrightarrow \forall j, \boxed{\mu_j^* = \frac{\sum\limits_{i=1}^{n}\tau_i^j x_i}{\sum\limits_{i=1}^{n}\tau_i^j}}\end{aligned}$$

- To compute $\sigma_{j,t+1}$, we solve the following optimization problem:

$$\max_{\sigma} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}\tau_i^j\Big(-\frac{d}{2}\log\sigma_j-\frac{1}{2\sigma_j^2}(x-\mu_{j,t+1})^T(x-\mu_{j,t+1})\Big)$$

This time, the objective function is not concave. To solve this optimization problem, we introduce the variables $(a_j)_j$ defined as $\forall j,\ a_j=\frac{1}{\sigma_j^2}$
The optimization problem becomes :

$$\max_{a} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}\tau_i^j\Big(\frac{d}{4}\log a_j-\frac{a_j}{2}(x-\mu_{j,t+1})^T(x-\mu_{j,t+1})\Big)$$

The objective function is now concave, so we can find is maximum by setting is gradient to 0:

$$\begin{aligned}\forall j, \sum_{i=1}^{n}\tau_i^j\big(-\frac{d}{4a_j}-\frac{1}{2}(x-\mu_{j,t+1})^T(x-\mu_{j,t+1})\big) &\Longrightarrow -\frac{d}{2a_j}\sum_{i=1}^{n}\tau_i^j=\sum_{i=1}^{n}\tau_i^j(x-\mu_{j,t+1})^T(x-\mu_{j,t+1}) \\ &\Longrightarrow \forall j, \frac{1}{a_j^*}=\boxed{{\sigma_j^*}^2=\frac{\sum\limits_{i=1}^{n}\tau_i^j(x-\mu_{j,t+1})^T(x-\mu_{j,t+1})}{\sum\limits_{i=1}^{n}\tau_i^j}}\end{aligned}$$