In The Name OF God

Sharif University

Department of Electrical Engineering

Introduction to Machine Learning

Project Phase 1

Instructor:

Dr. Sajad Amini

Authors:

Sepehr Kazemi Ranjbar

99106599

Arman Lotfalikhani

99106599

2 Expectation Maximization

2.2.3 The M Step

Theory Question 1. In your own words, explain how the MM algorithm can deal with nonconvex optimization objective functions by considering simpler convex objective functions.

We do not have a general way to compute the global optimizer of the intended function, and the MM method only assures that almost all the time we converge to a local optimizer. Here are some methods that are used to get more accurate results:

1. we can start the MM method from many different random starting points, and choosing the best local optimizer as the best answer. Still, we may not find the global optimizer in this method.

2. There are special cases where the EM algorithm does not converge to a point at all. We Quote an example from [**Article**]:
   We consider the function

   $$f(\rho, \sigma^2) = 8\ln\sigma^2 + \frac{18}{\sigma^2} + 2\ln\big(1-\rho^2\big) + \frac{4}{\sigma^2(1-\rho^2)}$$

   to minimize over the interval $\sigma \geq 0 \quad |\rho| \leq 1$. In minimizing this function (which originates from the maximum-likelihood estimation of the variance and correlation coefficient of bivariate normal data with missing observations), the EM algorithm gives the majorizer function:

   $$g(\sigma^2, \rho, \sigma_n^2, \rho_n) = f(\rho, \sigma^2) + 2(\ln\frac{\sigma^2(1-\rho^2)}{\sigma_n^2(1-\rho_n^2)} + \frac{\sigma_n^2(1-\rho_n^2)}{\sigma^2(1-\rho^2)} - 1)$$

   This function has two global optimizers which are symmetric: $(\sigma_{n+1}^2, \rho_{n+1}) = (3, \pm\sqrt{2/3 - \sigma_n^2(1-\rho_n^2)/6}$. If we use $\sigma_0^2 = 3$ we will get:

   $$\rho_{n+1} = -sgn(\rho_n)\sqrt{\frac{1-\rho_n^2}{6}}$$

   Which will oscillate between $\pm\frac{1}{\sqrt{3}}$
   To remove such possibilities, one way is to add the function $\lambda||x-x_n||_2^2$ to our original majorizer. In this way, the MM algorithm is guaranteed to converge to a local minimizer of the cost function.

3. Another way is to use the generalized MM, which in each iteration, instead of one majorizer, considers a set of majorizer functions that need not touch the cost function.

Theory Question 2. Briefly explain how the formula for mixture models:

$$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} p_{\mathbf{Z}}(\mathbf{z_k};\boldsymbol{\theta}) p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{Z}=\mathbf{z_k};\boldsymbol{\theta}),$$

is the same as the sum over all possible values of Z(i) in equation (9). Explain why it's easier to optimize $p_{\mathbf{X},\mathbf{Z}}(\mathbf{x_n},\mathbf{z_n};\grave{})$ than $p_{\mathbf{X}}(\mathbf{x_n};\boldsymbol{\theta})$ in the context of mixture models.

We are assuming a hierarchical model for generating the data samples, so Z partitions the sample space of X. The according to the total probability law, we can write:

$$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{Z}=\mathbf{z_k};\boldsymbol{\theta}) = \sum_{k=1}^{K} p_{\mathbf{Z}}(\mathbf{z_k};\boldsymbol{\theta}) p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{Z}=\mathbf{z_k};\boldsymbol{\theta})$$

We have also used the definition of conditional probability. In order to convert this expression to the mixture model, we only need to set $p_{\mathbf{Z}}(\mathbf{z_k};\boldsymbol{\theta}) = \pi_k$ and assume X and Z are independent, each with distributions $p_k(X,\boldsymbol{\theta}_k)$ which are from the same distribution type. The main advantage of this new equation is we can use Jensen's inequality to expand the log of this sum to get a lower bound. Then, we can readily exploit the fact that all conditional distributions are of the same kind.

Theory Question 3. Read about variational inference (or variational bayesian methods) and compare it with the procedure we used for the EM algorithm (You might want to check Wikipedia for this!).

Consider a model with latent variables $\mathbf{z}$ and observations $\mathbf{x}$ and known parameters $\boldsymbol{\theta}$, actually $\boldsymbol{\theta}$ is known but if it's unknown we can add it to $\mathbf{z}$. Now we have to compute posterior :

$$p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}) = \frac{p(\mathbf{z},\mathbf{x}|\boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{x})}$$

Actually the denominator is not simple to compute because we can have several latent variables or maybe they are continuous, so the integral is not easy to compute. in these situations we have to change our procedure and that is the point *Variational inference* enter. the idea is that we have to estimate posterior with distributions $\{q_n\}_{n=1}^{N}$ which come from a distribution family like *Exponential*. actually we want distribution that minimize :

$$\mathrm{D}_{\mathbb{KL}}(q(\mathbf{z}|\boldsymbol{\psi})||p(\mathbf{z}|\mathbf{x}))$$

the $\boldsymbol{\psi}$ represent parameters of q distributions. now we expand the above equation:

$$\begin{aligned}\mathrm{D}_{\mathbb{KL}}(q(\mathbf{z}|\boldsymbol{\psi})||p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q(\mathbf{z}|\boldsymbol{\psi})}\left[\ln q(\mathbf{z}|\boldsymbol{\psi}) - \ln \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}\right] \\ &= \underbrace{\mathbb{E}_{q(\mathbf{z}|\boldsymbol{\psi})}\left[\ln q(\mathbf{z}|\boldsymbol{\psi}) - \ln p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \ln p_{\boldsymbol{\theta}}(\mathbf{z})\right]}_{\mathcal{L}(\boldsymbol{\psi}|\boldsymbol{\theta},\mathbf{x})} + \ln p_{\boldsymbol{\theta}}(\mathbf{x})\end{aligned}$$

The final term is hard to compute and acutually we don't need it because we want to minimize $\mathrm{D}_{\mathbb{KL}}(q(\mathbf{z}|\boldsymbol{\psi})||p(\mathbf{z}|\mathbf{x}))$ with respect to $\boldsymbol{\psi}$.
Another important assumption is that all latent variables are independent of each other, i.e.,

$$q(\mathbf{z}|\boldsymbol{\psi}) = \prod_{m=1}^{M} q_m(z_m)$$

that distribution q_m has parameters ψ_m. this is called mean field approximation, so we have:

$$\mathcal{L}(\boldsymbol{\psi}|\mathbf{x},\boldsymbol{\theta}) = -\int q(\mathbf{z}|\boldsymbol{\psi}) \ln p_\theta(\mathbf{x},\mathbf{z})\mathrm{d}\mathbf{z} - \sum_{m=1}^{M} \mathbb{H}(q_m)$$

for optimizing the above equation we use specefic coordinate ascent we called **CAVI**. at the first we write the mean field equation again:

$$\mathcal{L}(\boldsymbol{\psi}|\mathbf{x},\boldsymbol{\theta}) = \sum_{z_1}\sum_{z_2}\cdots\sum_{z_M} q_1(z_1)q_2(z_2)\cdots q_M(z_M)\ln p_{\boldsymbol{\theta}}(z_1,z_2,\ldots,z_M,\mathbf{x}) + \sum_{m=1}^{M}\mathbb{H}(q_m)$$

we optimize this function for q_i, so we have:

$$\mathcal{L}(\boldsymbol{\psi}|\mathbf{x},\boldsymbol{\theta}) = \sum_{z_i} q_i(z_i)\left[\ln \tilde{f}_i(z_i) - \ln(q_i(z_i))\right] + \text{const}$$

where

$$\tilde{f}_i(z_i) =$$
$$\exp\left[\sum_{z_1}\cdots\sum_{z_{i-1}}\sum_{z_{i+1}}\cdots\sum_{z_M} q_1(z_1)\cdots q_{i-1}(z_{i-1})q_{i+1}(z_{i+1})q_M(z_M)\ln p_{\boldsymbol{\theta}}(z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_M,\mathbf{x})\right]$$

finally we reach that:

$$\mathcal{L}(\boldsymbol{\psi}|\mathbf{x},\boldsymbol{\theta}) \propto \sum_{z_i} \mathrm{D}_{\mathbb{KL}}(q_i||\tilde{f}_i)$$

so for optimize that we set:

$$q_i(z_i) = \tilde{f}_i(z_i)$$

Or in the other word:

$$q_i(z_i) \propto \exp(\mathbb{E}_{q-i}\left[\ln p_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{x})\right])$$

where $\mathbb{E}\left[p_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{x})\right]$ take expectation through all variable except z_i.
in our procedure we directly compute posterior because we have just 3 clusters (in simulation) or finite cluster. so if our clustering problem was more complicated possibly we can't compute posterior, so we have to use this procedure.

3 EM Algorithm for GMM and CMM

3.1 EM for Gaussian Mixture Model

Theory Question 4. Compute estimate of parameters for Gaussian Mixture Models for N observed data $\{\mathbf{x}_i\}_{i=1}^{N}$.

1. Determine model parameters and initialize them.

   The parameters of model are:

   $$\{\boldsymbol{\mu}_k\}_{k=1}^{K}, \{\boldsymbol{\Sigma}_k\}_{k=1}^{K}, \{\pi_k\}_{k=1}^{K}$$

   Out problem is not convex, and has many local minimizers that are not the global minimizer of the cost function. An important consequence is we may converge to a fewer number of clusters, that is, some of the π_k converge to zero. As we verify experimentally, the algorithm is most sensitive to the initialization of $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ There are numerous ways of initializing the parameters. Some of the most popular methods are as follows:

   (a) Running the algorithm for many times with random initialization and choosing the one with the minimum log likelihood. For the initialization, we randomly break the dataset into K parts and use the mean and covariance matrix of the selected sets.

   (b) Using a k-means algorithm (Similar to GMM with fixed $Sigma_k = I$ and $\pi_k = 1/K$)

   (c) Choosing K random points of the dataset, according to [**Murphy**]

   (d) Using the Farthest point method, according to [**Murphy**]

   (e) Assuming some prior information about the means of distributions.

   We have implemented ways c,d and e.
   Farthest point method: First, we choose a random point to set μ_1. Then we choose the point with the most (Euclidian) distance to it as the second central point. After that, to each of the remaining points, we associate the minimum of its distance to mu_1 and mu_2 and choose the point with the maximum number. The initial minimization ensures that we do not choose a point close to mu_1 and far from m_2. The procedure is repeated $K-1$ times until we find all the center points. Assuming prior information: In this project, we assume that we know the first 200 points are from distribution 1 and so on. in this method, we initialize:

   $$\boldsymbol{\mu}_k^{(0)} = \frac{3}{N} \sum_{n=N_{k-1}}^{N_k} \mathbf{x}_n$$

and for all of the three methods we initalize as follows:

$$\boldsymbol{\Sigma}_k^{(0)} = I$$
$$\pi_k^{(0)} = \frac{1}{K}$$

2. Compute complete dataset likelihood.

We define $\mathbf{y_n} = [\mathbf{x}_n; \mathbf{z}_n]$, then $\{\mathbf{y}_n\}_{n=1}^N$ are independent so we have:

$$\begin{aligned}
\mathrm{LL}(\boldsymbol{\theta}) &= \ln p(\mathcal{D}|\boldsymbol{\theta}) = \ln \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) \\
&= \sum_{n=1}^{N} \ln p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln(p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})p(\mathbf{z}_n|\boldsymbol{\theta})) \\
&= \sum_{n=1}^{N} \ln \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}} + \sum_{n=1}^{N} \ln \prod_{k=1}^{K} \pi_k^{z_{nk}} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \ln \pi_k \\
&= -\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\ln(|\boldsymbol{\Sigma}_\mathbf{k}|) - 2 \ln \pi_k + (\mathbf{x}_n - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_\mathbf{k}}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)\right) - \frac{Nd}{2} \ln(2\pi)
\end{aligned}$$

where z_{nk} is one-hot encoding of $\mathbf{z}_n$.

3. Find closed-form solution for parameters using EM algorithm.

As discussed above we use equation (21) to derive parameters. actually in E step we set distributions $q_n(z_n) = p(z_n|x_n)$ but in Gaussian mixture model, we assume that $\{q_n\}_{n=1}^N$ are identical, we have just distribution as $\text{Cat}(z_n|\boldsymbol{\pi})$. at the first in E step we define:

$$q_{nk}^{(t)} = p(\mathbf{z}_n = k|\mathbf{x}_n, \boldsymbol{\theta}^{(t)}) = \frac{p(\mathbf{z}_n = k|\boldsymbol{\theta})p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})}{p(\mathbf{x}_n|\boldsymbol{\theta})}$$
$$= \frac{\pi_k^{(t)} p(\mathbf{x}_n|\boldsymbol{\theta}_k^{(t)})}{\sum_{k'=1}^K \pi_{k'}^{(t)} p(\mathbf{x}_n|\boldsymbol{\theta}_{k'}^{(t)})}$$

Now we have:

$$\begin{aligned}
l(\boldsymbol{\theta})^{(t)} &= \sum_{n=1}^N \mathbb{E}_{q_n^{(t)}}\{\ln[p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})p(z_n)]\} \\
&= \sum_{n=1}^N \mathbb{E}_{q_n^{(t)}}[\ln p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})] + \sum_{n=1}^N \mathbb{E}_{q_n^{(t)}}[\ln p(\mathbf{z}_n|\boldsymbol{\theta})] \\
&= \sum_{n=1}^N \mathbb{E}_{q_n^{(t)}}\left[\ln \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}\right] + \sum_{n=1}^N \mathbb{E}_{q_n^{(t)}}\left[\ln \prod_{k=1}^K \pi_k^{z_{nk}}\right] \\
&= \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}_{q_n^{(t)}}[z_{nk}] \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}_{q_n^{(t)}}[z_{nk}] \ln \pi_k \\
&= \sum_{n=1}^N \sum_{k=1}^K q_{nk}^{(t)} \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_{n=1}^N \sum_{k=1}^K q_{nk}^{(t)} \ln \pi_k \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K q_{nk}^{(t)} \left(\ln(|\boldsymbol{\Sigma}_\mathbf{k}|) + (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)\right) + \sum_{n=1}^N \sum_{k=1}^K q_{nk}^{(t)} \ln \pi_k + \text{const}
\end{aligned}$$

Now in M step we take gradient to obtain $\boldsymbol{\theta}^{(t+1)}$:

$$\frac{\partial l^{(t)}}{\partial \boldsymbol{\mu}_k} = 0 \Rightarrow \sum_{n=1}^N q_{nk}^{(t)} (\boldsymbol{\Sigma}_k^{-1} + (\boldsymbol{\Sigma}_k^{-1})^T)(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \Rightarrow \boxed{\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{n=1}^N \mathbf{x}_n q_{nk}^{(t)}}{\sum_{n=1}^N q_{nk}^{(t)}}}$$

$$\frac{\partial l^{(t)}}{\partial \boldsymbol{\Sigma}_\mathbf{k}} = 0 \Rightarrow \sum_{n=1}^N q_{nk}^{(t)} \boldsymbol{\Sigma}_k^{-T} - \sum_{n=1}^N q_{nk}^{(t)} \boldsymbol{\Sigma}_k^{-T} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-T} = 0$$

$$\Rightarrow \boxed{\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{n=1}^N q_{nk}^{(t)} (\mathbf{x}_n - \boldsymbol{\mu}_k^{(t+1)})(\mathbf{x}_n - \boldsymbol{\mu}_k^{(t+1)})^T}{\sum_{n=1}^N q_{nk}^{(t)}} = \frac{\sum_{n=1}^N q_{nk}^{(t)} \mathbf{x}_n \mathbf{x}_n^T}{\sum_{n=1}^N q_{nk}^{(t)}} - \boldsymbol{\mu}_k^{(t+1)} (\boldsymbol{\mu}_k^{(t+1)})^T}$$

for finding $\{\pi_k\}_{k=1}^N$ we have to use the Lagrange multiplier method because we have the constraint, $g(\boldsymbol{\pi}) = \sum_{k=1}^K \pi_k - 1 = 0$:

$$\frac{\partial \left(l^{(t)} + \lambda g(\boldsymbol{\pi})\right)}{\partial \pi_k} = 0 \Rightarrow \frac{1}{\pi_k} \sum_{n=1}^N q_{nk} + \lambda = 0$$

$$\frac{\partial \left(l^{(t)} + \lambda g(\boldsymbol{\pi})\right)}{\partial \lambda} = 0 \Rightarrow \sum_{k=1}^K \pi_k = 1$$

$$\Rightarrow \lambda = -N \Rightarrow \boxed{\pi_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^N q_{nk}^{(t)}}$$

finally we estimate $q_{nk}^{(t)}$ in E step of its distribution and then use above results to estimate parameter vector $\boldsymbol{\theta}$.

3.2 EM for Categorical Mixture Model

Theory Question 4. Compute estimate of parameters for Categorical Mixture Models for N observed data $\{\mathbf{x}_i\}_{i=1}^N$.

1. Determine model parameters and initialize them.

   The model parameters are:

   $$\{\pi_k\}_{k=1}^K,\ \{\{\theta_{kc}\}_{c=1}^C\}_{k=1}^K$$

   we initialize them as:

   $$\pi_k^{(0)} = \frac{1}{K},\ \theta_{kc}^{(0)} = \frac{\text{number of c labels in k'th data}}{\text{total labels of k'th data}}$$

   we use uniform distribution for pi_k, because we have no prior knowledge. also we set every θ_{kc} proportional to number of c labels in k'th data as like GMMs.

2. Compute complete dataset likelihood.

   We compute complete log likelihood of $p(\{\mathbf{x}_n\}_{n=1}^N, \{\mathbf{z}_n\}_{n=1}^N|\boldsymbol{\theta})$, also we assume that $\mathbf{x}_n, \mathbf{z}_n$ are one-hot encoded:

   $$\begin{aligned}
   \ln p(\mathcal{D}|\boldsymbol{\theta}) &= \ln \prod_{n=1}^N p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) = \sum_{n=1}^N \ln p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) \\
   &= \sum_{n=1}^N \ln \left[p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}) p(\mathbf{z}_n|\boldsymbol{\theta})\right] \\
   &= \sum_{n=1}^N \ln \left[\prod_{k=1}^K \text{Cat}(\mathbf{x}_n|\boldsymbol{\theta}_k)^{z_{nk}} \prod_{k=1}^K \pi_k^{z_{nk}}\right]
   \end{aligned}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln \prod_{c=1}^{C} \theta_{kc}^{x_{nc}} + \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln \pi_k$$
$$= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \sum_{c=1}^{C} x_{nc} \ln \theta_{kc} + \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln \pi_k$$

3. Find closed-form solution for parameters using EM algorithm.

We already defined q_{nk} in last question so we have:

$$\begin{aligned} l(\boldsymbol{\theta})^{(t)} &= \sum_{n=1}^{N} \mathbb{E}_{q_n^{(t)}} \{\ln[p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})p(z_n)]\} \\ &= \sum_{n=1}^{N} \mathbb{E}_{q_n^{(t)}} [\ln p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta})] + \sum_{n=1}^{N} \mathbb{E}_{q_n^{(t)}} [\ln p(\mathbf{z}_n|\boldsymbol{\theta})] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q_n^{(t)}} [z_{nk}] \ln \mathrm{Cat}(\mathbf{x}_n|\boldsymbol{\theta}_k) + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{q_n^{(t)}} [z_{nk}] \ln \pi_k \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}^{(t)} \sum_{c=1}^{C} x_{nc} \ln \theta_{kc} + \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}^{(t)} \ln \pi_k \end{aligned}$$

now we optimize the above equation in M step, for that we have to define constraint, $g(\boldsymbol{\theta_k}) = \sum_{c=1}^{C} \theta_{kc} - 1 = 0$, then we have:

$$\frac{\partial \left(l(\boldsymbol{\theta})^{(t)} + \lambda_k g(\boldsymbol{\theta}_k)\right)}{\partial \theta_{kc}} = 0 \Rightarrow \frac{1}{\theta_{kc}} \sum_{n=1}^{N} q_{nk} x_{nc} + \lambda_k = 0$$

$$\sum_{c=1}^{C} \theta_{kc} = 1 \Rightarrow \lambda_k = -\sum_{n=1}^{N} q_{nk} \sum_{c=1}^{C} x_{nc} \Rightarrow \boxed{\theta_{kc}^{(t+1)} = \frac{\sum_{n=1}^{N} q_{nk}^{(t)} x_{nc}}{\sum_{n=1}^{N} q_{nk}^{(t)} \sum_{c=1}^{C} x_{nc}}}$$

Now we use constraint, $f(\boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k - 1 = 0$, we see that this is like the Gaussian case, so we just mention the result:

$$\boxed{\pi_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} q_{nk}^{(t)}}$$

4 EM Algorithm in Real Applications

As is requested in the project description in Telegram, we have included out report for this part in the jupyter notebook file

References

[1] Kevin P.Murphy (2012) *Machine Learning-A Probabilistic Perspective*, MIT Press.

[2] Kenneth Lange, Joong-Ho Won, Alfonso Landeros, and Hua Zhou1 :*Nonconvex Optimization via MM Algorithms: Convergence Theory* Wiley StatsRef