Introduction to Machine Learning

Project Phase 1

MohammadPouya Toroghi

Ilia Hashemi Rad

Theory Questions

Question 1

Solution:

The **MM (Majorization-Minimization) algorithm** is designed to handle complex non-convex optimization problems by iteratively simplifying the problem using convex surrogate functions. The key idea is to replace the original, often non-convex, objective function $l(\theta)$ with a simpler surrogate function $Q(\theta, \theta^{(t)})$ at each iteration. This surrogate function satisfies the following conditions:

- **Tangency Condition**: $Q(\theta^{(t)}, \theta^{(t)}) = l(\theta^{(t)})$, meaning the surrogate and the original function coincide at the current parameter estimate.
- **Lower Bound Condition**: $Q(\theta, \theta^{(t)}) \leq l(\theta)$ for all θ, ensuring that the surrogate underestimates the original function.

These properties ensure that the MM algorithm produces a sequence of iterates $\theta^{(t)}$ where each update improves the original objective function by solving a simpler problem. The update rule is:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta, \theta^{(t)}),$$

and this guarantees that the objective function $l(\theta)$ increases monotonically at each step.

Handling Non-Convexity: For non-convex objective functions, directly maximizing $l(\theta)$ can be difficult because of multiple local maxima. The MM algorithm tackles this by optimizing a surrogate function $Q(\theta, \theta^{(t)})$, which is often chosen to be convex. Convex functions are much easier to optimize because they have a unique global maximum, allowing the algorithm to avoid the difficulties associated with non-convex optimization.

Although the MM algorithm guarantees convergence to a local maximum, it may not reach the global maximum in the case of non-convex objectives. However, by carefully choosing surrogate functions that are convex, the algorithm simplifies the optimization problem, making it easier to find a good solution iteratively.

Flexibility in Surrogate Function Selection: The MM algorithm allows flexibility in constructing the surrogate function $Q(\theta, \theta^{(t)})$. Various techniques, such as Jensen's inequality, Cauchy-Schwarz inequality, or a second-order Taylor expansion, can be used to derive convex surrogates. This flexibility enables the algorithm to be adapted to different types of problems, improving the chances of efficient and effective optimization.

Conclusion: The MM algorithm handles non-convex optimization by iteratively constructing convex or simpler surrogate functions, optimizing them, and ensuring monotonic improvement in the original objective. While convergence is to a local maximum in non-convex settings, the flexibility in choosing surrogates and the ease of solving convex problems make the MM algorithm a powerful approach for complex optimization tasks.

Question 2

Solution:

The mixture model formula can be written as:

$$p(y; \theta) = \sum_{k=1}^{K} p_Z(z_k; \theta) p_{Y|Z}(y | Z = z_k; \theta)$$

This formula expresses the marginal likelihood of y by summing over all possible values of the latent variable $Z = z_k$. In a dataset with N observations, the likelihood for all data points $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\}$ becomes:

$$p_Y(\mathbf{y}; \theta) = \prod_{i=1}^{N} p_Y(y^{(i)}; \theta) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} p_{Y,Z}(y^{(i)}, Z^{(i)} = k; \theta) \right)$$

We can decompose the joint probability $p_{Y,Z}(y^{(i)}, Z^{(i)} = k; \theta)$ using:

$$p_{Y,Z}(y^{(i)}, Z^{(i)} = k; \theta) = p_Z(z_k; \theta) p_{Y|Z}(y^{(i)} | Z^{(i)} = k; \theta)$$

Thus, the overall likelihood becomes:

$$p_Y(\mathbf{y}; \theta) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} p_Z(z_k; \theta) p_{Y|Z}(y^{(i)} | Z^{(i)} = k; \theta) \right)$$

This is equivalent to summing over all possible values of the latent variable $Z^{(i)}$. Now, the reason why it is easier to optimize $p_{Y,Z}(y_n, z_n; \theta)$ rather than $p_Y(y_n; \theta)$ is as follows:

Directly optimizing $p_Y(y_n; \theta)$ is difficult because it involves summing over all possible latent variables Z, which makes the marginal likelihood complex and harder to optimize. On the other hand, optimizing the joint likelihood $p_{Y,Z}(y_n, z_n; \theta)$ is simpler because:

1. The latent variables Z are treated as observed when optimizing the joint likelihood. This simplifies the problem by avoiding the need to marginalize over Z.

2. The Expectation-Maximization (EM) algorithm is commonly used to handle this. In the E-step, we compute the expected value of the latent variables, and in the M-step, we maximize the joint likelihood $p_{Y,Z}(y_n, z_n; \theta)$, which is easier to handle computationally.

Thus, optimizing $p_{Y,Z}(y_n, z_n; \theta)$ is easier than optimizing $p_Y(y_n; \theta)$ because we avoid the complexities of marginalizing over the latent variables.

Question 3

Solution:

Variational Inference (VI) and the Expectation-Maximization (EM) algorithm are both used to approximate a true posterior distribution, but they employ different approaches and assumptions.

The EM algorithm iteratively maximizes the likelihood of observed data through two steps: the Expectation (E-step) and the Maximization (M-step). In the E-step, the algorithm computes the expected values of the latent variables, given the current estimates of the model parameters. In the M-step, the parameters are updated to maximize the expected log-likelihood of the observed data. EM assumes that latent variables are unobserved, and the goal is to estimate both the latent variables and the parameters. EM is widely used for problems where the direct optimization of the likelihood is difficult due to missing or incomplete data.

On the other hand, Variational Inference (VI) is a deterministic optimization method that minimizes the Kullback-Leibler (KL) divergence between the true posterior distribution and a simpler, tractable distribution (such as a Gaussian). Unlike EM, VI does not estimate specific latent variables but instead estimates an entire distribution over them. This is achieved by approximating the posterior distribution over the latent variables and parameters, allowing for fully Bayesian estimation. VI can thus be seen as a generalization of EM, moving from Maximum A Posteriori (MAP) estimation, which finds the most probable value of each parameter, to approximating the entire posterior distribution.

A key difference is that while EM provides point estimates for parameters, VI computes a full distribution, making it more flexible and applicable to complex probabilistic models where a full Bayesian treatment is desired. However, the trade-off is that VI requires specifying an approximating family of distributions, which introduces additional assumptions and might limit its flexibility depending on the choice of the distribution.

Question 4

Solution: 1:

Assume that we have a dataset of N observed d-dimensional points $\{\boldsymbol{x}_i\}_{i=1}^{N}$. The dataset follows a mixture of K Multivariate Normal distributions such that:

$$p(\boldsymbol{x}|\Theta) = \sum_k \pi_k \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are the prior probability, mean vector, and covariance matrix of the k-th Multivariate Gaussian model, respectively.
Our task is to estimate the set of parameters $(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for each Multivariate Gaussian model using the dataset.

Clearly, in the first step, we should initialize the mentioned parameters. Here, we choose an algorithm to do so. In this algorithm, we initialize the parameters π_k uniformly:

$$\pi_k^0 = \frac{1}{K} \qquad \forall k; \quad 1 \le k \le K$$

which means each point in the dataset is equally likely to belong to the k-th Multivariate Gaussian model. To initialize the Gaussian models' parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, we first divide the dataset randomly into K clusters with $M = \frac{N}{K}$ data points. (Note that if the value of M is not an integer, we would divide the dataset into clusters such that all the clusters have approximately the same sizes.) Then, using the MLE method, we have:

$$\begin{cases} \boldsymbol{\mu}_k^0 = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{x}_j^{(k)} \\ \boldsymbol{\Sigma}_k^0 = \frac{1}{M}\sum_{j=1}^{M} (\boldsymbol{x}_j^{(k)} - \boldsymbol{\mu}_k^0)(\boldsymbol{x}_j^{(k)} - \boldsymbol{\mu}_k^0)^T \end{cases} \qquad \forall k; \quad 1 \le k \le K$$

where $\boldsymbol{x}_j^{(k)}$ denotes the j-th point of the k-th cluster.

Solution: 2:

In this section, we are supposed to determine the likelihood of the complete dataset. According to the descriptions in the previous part, we have:

$$p(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}) = \prod_i p(\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}) = \prod_i \sum_k p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta}) = \prod_i \sum_k p(z_i = k) p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\Theta}) = \prod_i \sum_k \pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$= \prod_i \sum_k p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta}) \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta})} = \prod_i \sum_k \omega_{i,k} \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\omega_{i,k}}$$

In which, using Bayes' Rule, we have:

$$\omega_{i,k} \triangleq p^*(z_i = k|\boldsymbol{x}_i, \boldsymbol{\Theta}) = \frac{p(z_i = k) p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\Theta})}{\sum_{k'} p(z_i = k') p(\boldsymbol{x}_i|z_i = k', \boldsymbol{\Theta})} = \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

Therefore, the complete log-likelihood function is determined as follows (this part is for the next section):

$$L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}) = \sum_i \log \sum_k \omega_{i,k} \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\omega_{i,k}} \ge \sum_i \sum_k \omega_{i,k} \log \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\omega_{i,k}} \qquad *$$

Solution: 3:

Assume that the latent parameters at iteration t are defined as calculated previously:

$$\omega_{i,k}^t \triangleq p^*(z_i = k|\boldsymbol{x}_i, \boldsymbol{\Theta}^t) = \frac{\pi_k^t \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)}{\sum_{k'} \pi_{k'}^t \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Sigma}_{k'}^t)}.$$

Using the results from the previous parts and the reference book, we can define an instance iterative target function (the lower bound function) to work with:

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) = \sum_i \sum_k \omega_{i,k}^t \log \frac{\pi_k \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\omega_{i,k}^t}$$

$$= \sum_i \sum_k \omega_{i,k}^t \log \left(\frac{\pi_k}{\omega_{i,k}^t \sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \; \exp\left[-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \; \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right] \right).$$

This function obviously satisfies the two constraints: $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) \leq L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta})$ (based on *) and $Q(\boldsymbol{\Theta}^t, \boldsymbol{\Theta}^t) = L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}^t)$ (based on the proof in the source).

We expand the result further to use it in subsequent steps:

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) = \sum_i \sum_k \omega_{i,k}^t \left[\log \pi_k - \log \omega_{i,k}^t - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \; \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right]. \tag{1}$$

Now we proceed with the E-step and M-step:

1. **E-step**: The latent parameters to be estimated in each iteration of the E-step are determined in closed form:

   $$\omega_{i,k}^t = \frac{\pi_k^t \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)}{\sum_{k'} \pi_{k'}^t \, \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{k'}^t, \boldsymbol{\Sigma}_{k'}^t)},$$

   where ω is an $N \times K$ matrix.

2. **M-step**: In this step, we update the Θ parameters by maximizing $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)$ with respect to the parameters:

   $$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t).$$

   Using the expanded equation in (1), we conclude:

   $$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} \sum_i \sum_k \omega_{i,k}^t \left[\log \pi_k - \log \omega_{i,k}^t - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \; \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right].$$

   - Updating $\boldsymbol{\mu}_k$:

     $$\boldsymbol{\mu}_k^{t+1} = \arg\max_{\boldsymbol{\mu}_k} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t).$$

     To do so, by taking the partial derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)$ with respect to $\boldsymbol{\mu}_k$, we have:

     $$\left. \frac{\partial Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)}{\partial \boldsymbol{\mu}_k} \right|_{\boldsymbol{\mu}_k = \boldsymbol{\mu}_k^{t+1}} = \sum_i \omega_{i,k}^t \left. \frac{\partial \left[-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \; \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right]}{\partial \boldsymbol{\mu}_k} \right|_{\boldsymbol{\mu}_k = \boldsymbol{\mu}_k^{t+1}} = 0. \tag{2}$$

     On the other hand, it can be shown that if $\boldsymbol{Y}$ is a symmetric matrix, for any two vectors x and a we have:

     $$\frac{\partial (\boldsymbol{x} - \boldsymbol{a})^T \boldsymbol{Y} (\boldsymbol{x} - \boldsymbol{a})}{\partial \boldsymbol{x}} = -2\boldsymbol{Y}(\boldsymbol{x} - \boldsymbol{a}).$$

     So, using the fact that the covariance matrix and consequently its inverse are symmetric, from (2) we conclude:

     $$\sum_i \omega_{i,k}^t \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{t+1}) = 0 \Rightarrow \sum_i \omega_{i,k}^t \boldsymbol{x}_i = \boldsymbol{\mu}_k^{t+1} \sum_i \omega_{i,k}^t$$

$$\Rightarrow \boldsymbol{\mu}_k^{t+1} = \frac{\sum_i \omega_{i,k}^t \boldsymbol{x}_i}{\sum_i \omega_{i,k}^t}.$$

- Updating $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\Sigma}_k^{t+1} = \arg\max_{\boldsymbol{\Sigma}_k} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t).$$

To do so, by taking the partial derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)$ with respect to $\boldsymbol{\Sigma}_k^{-1}$, we have:

$$\frac{\partial Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)}{\partial \boldsymbol{\Sigma}_k^{-1}}\bigg|_{\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}_k^{t+1}} = \sum_i \omega_{i,k}^t \frac{\partial \left[-\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right]}{\partial \boldsymbol{\Sigma}_k^{-1}}\bigg|_{\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}_k^{t+1}} = 0. \tag{3}$$

On the other hand, for the symmetric matrices $\boldsymbol{\Sigma}$ and its inverse and for any vector a, we have:

$$\frac{\partial \log|\boldsymbol{\Sigma}|}{\partial \boldsymbol{\Sigma}^{-1}} = \frac{\partial \log|\boldsymbol{\Sigma}|}{\partial|\boldsymbol{\Sigma}|}\frac{\partial|\boldsymbol{\Sigma}|}{\partial \boldsymbol{\Sigma}^{-1}} = \frac{1}{|\boldsymbol{\Sigma}|}(-|\boldsymbol{\Sigma}|\boldsymbol{\Sigma}^T) = -\boldsymbol{\Sigma}^T,$$

$$\frac{\partial \boldsymbol{a}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{a}}{\partial \boldsymbol{\Sigma}^{-1}} = (\boldsymbol{x} - \boldsymbol{\mu}_k)(\boldsymbol{x} - \boldsymbol{\mu}_k)^T.$$

So, using these results, from (3) we conclude:

$$\sum_i \omega_{i,k}^t \left[\boldsymbol{\Sigma}_k^{t+1} - (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\right] = 0 \Rightarrow \sum_i \omega_{i,k}^t \boldsymbol{\Sigma}_k^{t+1} = \sum_i \omega_{i,k}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

$$\Rightarrow \boldsymbol{\Sigma}_k^{t+1} = \frac{\sum_i \omega_{i,k}^t (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{\sum_i \omega_{i,k}^t}.$$

- Updating π_k: Similarly, using the Lagrange multiplier method and the constraint $\sum_k \pi_k = 1$, it can be shown that:

$$\pi_k^{t+1} = \frac{1}{N}\sum_i \omega_{i,k}^t.$$

Question 5

Solution: 1:

Assume we have a dataset of N observed d-dimensional points $\{\boldsymbol{x}_i\}_{i=1}^N$. The dataset follows a mixture of K Categorical distributions such that:

$$p(\boldsymbol{x}|\Theta) = \sum_k \pi_k \ \mathrm{Cat}(\boldsymbol{x}|\boldsymbol{\theta}_k)$$

where π_k and $\boldsymbol{\theta}_k$ are the prior probability and the parameter vector of the kth Categorical model. Each $\boldsymbol{\theta} = (\theta_{k,1}, \cdots, \theta_{k,d})$ is a probability distribution over the labels $\boldsymbol{\mathcal{A}} = \{1, \cdots, d\}$.
For example, if $x_i = c$ with $1 \le c \le d$, we can say $p(x_i|\boldsymbol{\theta}_k) = \theta_{k,c}$.
Since each data point is d-dimensional, we can represent $\boldsymbol{x}_i$ as a d-dimensional vector. For instance, if $x_i = 3$, we can express it by turning on the third component of the data point vector ($\boldsymbol{x}_i = (0, 0, 1, 0, \cdots, 0)$ and $x_{i,3} = 1$).
Our task is to estimate the set of parameters $(\pi_k, \boldsymbol{\theta}_k)$ for each Categorical model using the dataset. Clearly, the first step is to initialize these parameters. Here, we choose an algorithm to do so.

In this algorithm, we initialize the parameters π_k uniformly:

$$\pi_k^0 = \frac{1}{K} \quad \forall k; \quad 1 \le k \le K$$

This means each point in the dataset is equally likely to belong to the kth Categorical model.

To initialize the Categorical models' parameter vector $\boldsymbol{\theta}_k$, we first divide the dataset randomly into K clusters with $M = \frac{N}{K}$ data points. (If M is not an integer, we divide the dataset into clusters such that all clusters have approximately the same sizes.) Then, using the MLE method, we have:

$$\theta^0_{k,j} = \frac{N^{(k)}_j}{M} \quad \forall k; \quad 1 \leq k \leq K$$

where $N^{(k)}_j$ denotes the number of times the jth component of the data point vectors in the kth cluster is turned on. Equivalently,

$$\boldsymbol{\theta}^0_k = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{x}^{(k)}_i$$

Solution: 2:

In this section, we aim to determine the complete dataset likelihood. According to the previous part, we have:

$$p(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}) = \prod_i p(\boldsymbol{x}_i, z_i|\boldsymbol{\Theta}) = \prod_i \sum_k p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta}) = \prod_i \sum_k p(z_i = k) p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\Theta}) = \prod_i \sum_k \pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$$

$$= \prod_i \sum_k p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta}) \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\Theta})} = \prod_i \sum_k \omega_{i,k} \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\omega_{i,k}}$$

where, using Bayes' Rule, we have:

$$\omega_{i,k} \triangleq p^*(z_i = k|\boldsymbol{x}_i, \boldsymbol{\Theta}) = \frac{p(z_i = k) p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\Theta})}{\sum_{k'} p(z_i = k') p(\boldsymbol{x}_i|z_i = k', \boldsymbol{\Theta})} = \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\sum_{k'} \pi_{k'} \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_{k'})}$$

Therefore, the complete log likelihood function is determined as follows (this part is for the next section):

$$L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}) = \sum_i \log \sum_k \omega_{i,k} \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\omega_{i,k}} \geq \sum_i \sum_k \omega_{i,k} \log \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\omega_{i,k}} \qquad *$$

Solution: 3:

Assume that the latent parameters at iteration t are defined as calculated before:

$$\omega^t_{i,k} \triangleq p^*(z_i = k|\boldsymbol{x}_i, \boldsymbol{\Theta}^t) = \frac{\pi^t_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}^t_k)}{\sum_{k'} \pi^t_{k'} \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}^t_{k'})}$$

Using the results from previous parts and the source book, we define an iterative target function (the lower bound function):

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) = \sum_i \sum_k \omega^t_{i,k} \log \frac{\pi_k \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k)}{\omega^t_{i,k}}$$

$$= \sum_i \sum_k \omega^t_{i,k} \log \left(\frac{\pi_k \ \prod_{j=1}^d (\theta_{k,j})^{x_{i,j}}}{\omega^t_{i,k}} \right)$$

This function satisfies two constraints: $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) \leq L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta})$ (based on *) and $Q(\boldsymbol{\Theta}^t, \boldsymbol{\Theta}^t) = L(\boldsymbol{\mathcal{D}}|\boldsymbol{\Theta}^t)$ (as proven in the source).

We expand the result further for use in the next steps:

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t) = \sum_i \sum_k \omega^t_{i,k} \left[\log \pi_k - \log \omega^t_{i,k} + \underbrace{\sum_{j=1}^d x_{i,j} \log \theta_{k,j}}_{\boldsymbol{x}_i \cdot \log \boldsymbol{\theta}_k} \right] \qquad (1)$$

Now, we proceed to the E-step and M-step:

1. **E-step**: The latent parameters to be estimated in each iteration of the E-step are determined in a closed form:
$$\omega_{i,k}^t = \frac{\pi_k^t \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_k^t)}{\sum_{k'} \pi_{k'}^t \ \mathrm{Cat}(\boldsymbol{x}_i|\boldsymbol{\theta}_{k'}^t)}$$
where ω is an $N \times K$ matrix.

2. **M-step**: In the M-step, we update the Θ parameters by maximizing $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)$:
$$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^t)$$
Using the expanded equation in (1), we get:
$$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} \sum_i \sum_k \omega_{i,k}^t \left[\log \pi_k - \log \omega_{i,k}^t + \boldsymbol{x}_i \cdot \log \boldsymbol{\theta}_k\right]$$
π_k and $\boldsymbol{\theta}_k$ can be estimated separately:
   (a) Update prior probabilities π_k: We assume the Lagrange multiplier technique to preserve the constraint $\sum_k \pi_k = 1$. Hence,
$$\pi_k^{t+1} = \frac{1}{N} \sum_i \omega_{i,k}^t$$
   (b) Update Categorical parameters $\boldsymbol{\theta}_k$: Similarly, we use the Lagrange multiplier to preserve the constraint $\sum_j \theta_{k,j} = 1$. Hence,
$$\theta_{k,j}^{t+1} = \frac{\sum_i \omega_{i,k}^t x_{i,j}}{\sum_i \omega_{i,k}^t}$$

Finally, the algorithm iterates between these steps until convergence. Thus, the mixture of Categorical distributions model is successfully estimated using the EM algorithm.