NOTES ON THE EXPECTATION-MAXIMIZATION (EM) ALGORITHM

IORDAN GANEV

1. Introduction and Set-up

These notes provide an exposition of the expectation-maximization (EM) algorithm for clustering. One can regard this algorithm as an unsupervised learning analogue of Linear Discriminant Analysis. The main reference for these notes is [For19, Section 9.2]. We focus on the theoretical justification for the algorithm rather than implementation details.

1.1. **The underlying process.** Suppose we have d-dimensional data that we would like to classify into k clusters. We imagine that the underlying sampling procedure is a (hidden) two-step process:

(1) A cluster is chosen. For $j = 1, \dots, k$, the probability of choosing the j-th cluster is unknown, and denoted by π_j. Note that $\sum_{j=1}^{k} \pi_j = 1$.
(2) Once a cluster is chosen, a sample is taken from a distribution corresponding to that cluster. Write $f_j(\mathbf{x}|\theta_j)$ be the probability density function of cluster j, where $\mathbf{x} \in \mathbb{R}^d$. We assume the formula for f_j is known, but that the parameters θ_j are unknown.

We group the unknown parameters into a tuple $\Theta = (\theta_1, \dots, \theta_k, \pi_1, \dots, \pi_k)$. The main question we would like to address is:

> $\rightarrow$ Assuming this process is valid (with all its unknown parameters), can we tell which cluster each sample came from?

We examine two of the main examples:

Example 1.1 (Mixture of normals)**.** Suppose each cluster is normally distributed, and let $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$ be the means. Suppose further that the covariance matrix of each cluster is known. We can apply a transformation to all the data that makes all the covariance matrices simultaneously the identity. This means that we can reduce to the case where each cluster has density function given by:

$$f_j(\mathbf{x}|\boldsymbol{\mu}_j z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T \left(\mathbf{x} - \boldsymbol{\mu}_j\right)\right)$$

In this case, the tuple of unkown parameters is given by: $\Theta = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \pi_1, \ldots, \pi_k)$. Note that $\boldsymbol{\mu}_j \in \mathbb{R}^d$ and each π_j belongs to $[0,1]$, with the condition that $\sum_{j=1}^k \pi_j = 1$. Therefore, the space of all possible parameters is $\left(\mathbb{R}^d\right)^k \times [0,1]^{k-1}$ and has dimension $dk + k - 1$.

Example 1.2 (Topic model)**.** Suppose each cluster is distributed according to a multinomial distribution with unknown probabilities $\mathbf{p}_j, \ldots, \mathbf{p}_k$. In other words we have:

$$f_j\left(\mathbf{x}|\mathbf{p}_j\right) = \frac{\left(\sum_{v=1}^d x_v\right)!}{x_1! \cdots x_d!} \prod_{u=1}^d p_{ju}^{x_u}$$

where $\mathbf{p}_j = (p_{ji}, \ldots, p_{jk})$ and $\sum_{u=1}^d p_{ju} = 1$ for all j. In this case, the tuple of unknown parameters is $\Theta = (\mathbf{p}_1, \ldots, \mathbf{p}_k, \pi_1, \ldots, \pi_k)$. Note that each $\mathbf{p}_j$ is a vector in $\mathbb{R}^d$ whose entries are non-negative and sum to one. Meanwhile, each π_j belongs to $[0,1]$, with the condition that $\sum_{j=1}^k \pi_j = 1$. Therefore, the space of all possible parameters is $\left([0,1]^{(d-1)}\right)^k \times [0,1]^{k-1}$ and has dimension $dk - 1$.

1.2. **One-hot vectors.** What does it mean to classify our data into k clusters? Recall the one-hot vectors in $\mathbb{R}^k$:

$$e_1 = (1, 0, \ldots, 0)\ ,\ e_2 = (0, 1, 0, \ldots, 0)\ ,\ \ldots\ e_k = (0, \ldots, 0, 1)$$

These are also the standard basis vectors for $\mathbb{R}^k$. Giving a classification of our data is the same as assigning a one-hot vector to each sample point; the coordinate of the single 1 indicates which cluster we've assigned the sample to. More formally, let $X \in \mathbb{R}^{N \times d}$ be the data matrix. As usual, the i-th row is a d-dimensional sample, denoted $\mathbf{x}_i \in \mathbb{R}^d$, and there are N samples total. We have:

Definition 1.3. A classification of the data $X \in \mathbb{R}^{N \times d}$ is a matrix $\Delta \in \mathbb{R}^{N \times k}$, each row of which is a one-hot vector.

Given a cluster $j \in \{1, \ldots, k\}$, we observe that:

$$p(e_j|\Theta) = \pi_j \qquad p(\mathbf{x}|\Theta, e_j) = f_j(\mathbf{x}|\theta_j) \qquad p(\mathbf{x}, e_j|\Theta) = f_j(\mathbf{x}|\theta_j)\pi_j$$

Indeed, the first equality reflects the fact that the cluster j is chosen with probability π_j, the second equality reflects the fact that the density function of $\mathbf{x}$ given that it comes from the j-th cluster is f_j, and the third equality is an application of the definition of conditional probability: $p(\mathbf{x}, e_j|\Theta) = p(\mathbf{x}|\Theta, e_j)p(e_j|\Theta)$. Using the law of total probability and conditioning on the one-hot vectors, we arrive at a formula for the density of $\mathbf{x}$:

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^k p(\mathbf{x}|e_j, \Theta)p(e_j|\Theta) = \sum_{j=1}^k f_j(\mathbf{x}|\theta_j)\pi_j \tag{1}$$

Next, using the definition of conditional probability, we observe that the conditional probability of cluster e_j given a point $\mathbf{x}$ is:

$$p(e_j|\mathbf{x},\Theta) = \frac{p(\mathbf{x},e_j|\Theta)}{p(\mathbf{x}|\Theta)} = \frac{f_j(\mathbf{x}|\theta_j)\pi_j}{\sum_{\ell=1}^k f_\ell(\mathbf{x}|\theta_\ell)\pi_\ell}.$$

Finally, for $i = 1,\dots,N$, and $j = 1,\dots,k$, let $w_{ij}^{(\Theta)}$ be the conditional probability of the i-th sample point $\mathbf{x}_i$ belonging to cluster e_j, that is:

$$w_{ij}^{(\Theta)} = p(e_j|\mathbf{x}_i,\Theta) = \frac{f_j(\mathbf{x}_i|\theta_j)\pi_j}{\sum_{\ell=1}^k f_\ell(\mathbf{x}_i|\theta_\ell)\pi_\ell}.$$

2. The EM algorithm

If there are N sample points in our data set, then the total number of ways to classify the data into k clusters is equal to k^N. Indeed, there are k choices for each of the N samples. Our task is to select a particular classification, and the EM algorithm provides a method for doing so. Here is the algorithm:

2.1. **The algorithm.**

- **Initialization**: Start with an estimate
$$\Theta^{(0)} = (\theta_1^{(0)},\dots,\theta_k^{(0)},\pi_1^{(0)},\dots,\pi_k^{(0)})$$
for the unknown parameters Θ.

- **Loop**: For $n = 0,1,2,\dots$, recursively define a new estimate $\Theta^{(n+1)}$ from $\Theta^{(n)}$ as follows:
  - **E-Step**: Take the expected value of the log-likelihood function for Θ given a classification, where each classification is weighted by its density given $\Theta^{(n)}$. This results in a function of Θ that we denote by $Q^{(n)}$:
$$Q^{(n)}(\Theta) = \mathbb{E}_{\Delta\sim p(\Delta|\Theta^{(n)},X)}[\mathcal{L}(\Theta;X,\Delta)]$$
This function admits an explicit formula (see Proposition 3.5 below):
$$Q^{(n)}(\Theta) = \sum_{i=1}^N\sum_{j=1}^k w_{ij}^{(n)}\left(\log f_j(\mathbf{x}_i|\theta_j) + \log\pi_j\right) \tag{2}$$
Note that $w_{ij}^{(n)} = p(e_j|\mathbf{x}_i,\Theta^{(n)}) = \frac{f_j(\mathbf{x}_i|\theta_j^{(n)})\pi_j^{(n)}}{\sum_{\ell=1}^k f_\ell(\mathbf{x}_i|\theta_\ell^{(n)})\pi_\ell^{(n)}}$ can be computed directly from the current estimate $\Theta^{(n)}$. (While the algorithm does not require computation of the log-likelihood function $\mathcal{L}(\Theta;X,\Delta)$ per se, this function also admits an explicit formula, see Proposition 3.2.)

– **M-Step**: In the function $Q^{(n)}$, we regard Θ as ranging over all possible parameter values; each is a candidate for the new estimate of the parameters. The value $Q^{(n)}(\Theta)$ is the 'score' of a candidate or paremeters Θ. This score varies depending on our current estimate $\Theta^{(n)}$. We update the estimate as the candidate with the largest score:

$$\Theta^{(n+1)} = \arg\max_{\Theta} Q^{(n)}(\Theta)$$

At this point, the sample $\mathbf{x}_i$ is tentatively assigned to the cluster j for which $w_{ij}^{(n+1)}$ is highest.

– **Check for convergence**: If the difference $\|\Theta^{(n+1)} - \Theta^{(n)}\|$ between $\Theta^{(n+1)}$ and $\Theta^{(n)}$ is larger than a predetermined tolerance level ϵ, then iterate the loop. Otherwise, let $\hat{\Theta} = \Theta^{(n+1)}$ be our final estimate of the parameters, and exit the loop.

• **Output**: Declare the final assignment of classes to be:

$$\mathbf{x}_i \text{ belongs to the cluster } \arg\max_j w_{ij}^{(\hat{\Theta})}$$

Equivalently, $\mathbf{x}_i$ is assigned to the j maximizing $f_j(\mathbf{x}_i|\hat{\theta}_j)\hat{\pi}_j$.

2.2. **The update rule.** In certain cases, one can perform the M-step by setting gradients equal to zero and computing an explicit formula for the update rule. We illustrate this in several cases.

Lemma 2.1. *Fix $j \in \{1, \ldots, k\}$. The update of π_j in the M-step is given by: $\pi_j^{(n+1)} = \frac{w_{ij}^{(n)}}{N}$.*

Proof. Ignoring terms that do not contain any π_j's, and recalling that the π_j sum to 1, we reduce the M-step for the π_j's to the following:

$$\arg\max_{(\pi_1,\ldots,\pi_k)} \sum_{i=1}^{N}\sum_{j=1}^{k} w_{ij}^{(n)} \log \pi_j \qquad \text{subject to} \quad \sum_{j=1}^{N} \pi_j = 1.$$

The corresponding Lagrangian is

$$L(\pi_1, \ldots, \pi_k, \lambda) = \sum_{i=1}^{N}\sum_{j=1}^{k} w_{ij}^{(n)} \log \pi_j + \lambda \left(1 - \sum_{j=1}^{N} \pi_j\right).$$

Setting the gradient equal to zero, one deduces that $\lambda = N$ and then arrives at the claimed update rule. □

Example 2.2 (Mixture of normals)**.** Suppose we are in the setting of Example 1.1, where each cluster is normally distributed with unknown mean $\boldsymbol{\mu}_j$. In this case, the computation of $w_{ij}^{(n)}$ simplifies to:

$$w_{ij}^{(n)} = \frac{\exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_j^{(n)}\right)^T\left(\mathbf{x}_i - \boldsymbol{\mu}_j^{(n)}\right)\right)\pi_j^{(n)}}{\sum_{\ell=1}^{k}\exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_\ell^{(n)}\right)^T\left(\mathbf{x}_i - \boldsymbol{\mu}_\ell^{(n)}\right)\right)\pi_\ell^{(n)}}.$$

Using the definition of f_j, and ignoring terms that do not contain any $\boldsymbol{\mu}_j$'s, we reduce the M-step for the $\boldsymbol{\mu}_j$'s to the following:

$$\boldsymbol{\mu}_j^{(n+1)} = \arg\max_{\boldsymbol{\mu}_j}\sum_{i=1}^{N} w_{ij}^{(n)}\left(-\frac{1}{2}\left(\mathbf{x}_i - \boldsymbol{\mu}_j\right)^T\left(\mathbf{x}_i - \boldsymbol{\mu}_j\right)\right)$$

Setting the gradient with respect to $\boldsymbol{\mu}_j$, one arrives at the update rule:

$$\boldsymbol{\mu}_j^{(n+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(n)}\mathbf{x}_i}{\sum_{i=1}^{N} w_{ij}^{(n)}}.$$

Example 2.3 (Topic model)**.** Suppose we are in the setting of Example 1.2, where each cluster is multinomially distributed with unknown probabilities $\mathbf{p}_j$. In this case, the computation of $w_{ij}^{(n)}$ simplifies to:

$$w_{ij}^{(n)} = \frac{\prod_{u=1}^{d}\left(p_{ju}^{(n)}\right)^{x_{iu}}\pi_j^{(n)}}{\sum_{\ell=1}^{k}\prod_{u=1}^{d}\left(p_{\ell u}^{(n)}\right)^{x_{iu}}\pi_\ell^{(n)}}.$$

Using the definition of f_j, ignoring terms that do not contain any $\mathbf{p}_j$'s, and recalling that the entries of each $\mathbf{p}_j$ sum to 1, we reduce the M-step for the $\mathbf{p}_j$'s to the following:

$$\mathbf{p}_j^{(n+1)} = \arg\max_{\mathbf{p}_j}\sum_{i=1}^{N}\sum_{u=1}^{d} w_{ij}^{(n)}\, x_{iu}\log(p_{ju}) \qquad \text{subject to } \sum_{u=1}^{d} p_{ju} = 1$$

Setting the gradient of the corresponding Lagrangian equal to zero, one arrives at the update rule:

$$\mathbf{p}_j^{(n+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(n)}\mathbf{x}_i}{\sum_{i=1}^{N} w_{ij}^{(n)} s_i}$$

where $s_i = \sum_{u=1}^{d} x_{iu} = \mathbf{x}_i^T\mathbb{1}_d$ is the sum of all the entries in $\mathbf{x}_i$.

3. Justification of the E-step

This section is devoted to proving Equation 2, which is the heart of the E-step of the algorithm. We first collect facts about the joint density function and the likelihood function.

3.1. **The joint density function.** For a one-hot vector $\delta \in \mathbb{R}^k$, we write δ as a tuple: $\delta = (\delta_1, \ldots, \delta_k)$ where exactly one of the δ_j is equal to one and the rest are zero.

Lemma 3.1. *Let $\mathbf{x} \in \mathbb{R}^d$ and $\delta \in \mathbb{R}^k$ a one-hot vector. Then the joint density of $\mathbf{x} \in \mathbb{R}^d$ and the classifying vector δ is given by:*

$$p(\mathbf{x}, \delta|\Theta) = \prod_{j=1}^{k} \left(f_j(\mathbf{x}|\theta_j)\pi_j\right)^{\delta_j}$$

Proof. Given a cluster $\ell \in \{1, \ldots, k\}$, we have already argued that:

$$p(\mathbf{x}, e_\ell|\Theta) = p(\mathbf{x}, e_\ell|\Theta)p(e_\ell|\Theta) = f_\ell(\mathbf{x}|\theta_\ell)\pi_\ell$$

The formula in the lemma follows from observing that $\delta = e_\ell$ for some ℓ, and the right-hand side have all factors equal to one except for the ℓ-th one. □

3.2. **The likelihood function.** We now give a formula for the log likelihood function of the parameters Θ given data X and a classification Δ. Write δ_{ij} for the i, j entry of the matrix $\Delta \in \mathbb{R}^{N\times k}$, and δ_i for the i-th row of Δ.

Proposition 3.2. *Given data X and a classification Δ, the log-likelihood function of Θ is:*

$$\mathcal{L}(\Theta; \Delta, X) = \sum_{i=1}^{N}\sum_{j=1}^{k} \delta_{ij}\left(\log f_j(\mathbf{x}_i|\theta_j) + \log \pi_j\right)$$

Proof. The independence of the samples implies that the density of X and Δ given Θ is the product of the individual densities: $p(X, \Delta|\Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i, \delta_i|\Theta)$. The log likelihood function is therefore:

$$\begin{aligned}
\mathcal{L}(\Theta; \Delta, X) &= \log p(X, \Delta|\Theta) = \log\left(\prod_{i=1}^{N} p(\mathbf{x}_i, \delta_i|\Theta)\right) \\
&= \log\left(\prod_{i=1}^{N}\prod_{j=1}^{k} \left(f_j(\mathbf{x}_i|\theta_j)\pi_j\right)^{\delta_{ij}}\right) = \sum_{i=1}^{N}\sum_{j=1}^{k} \log\left(\left(f_j(\mathbf{x}_i|\theta_j)\pi_j\right)^{\delta_{ij}}\right) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{k} \delta_{ij}\left(\log f_j(\mathbf{x}_i|\theta_j) + \log \pi_j\right)
\end{aligned}$$

where we use Lemma 3.1 for $p(\mathbf{x}_i, \delta_i|\Theta) = \left(f_j(\mathbf{x}_i|\theta_j)\pi_j\right)^{\delta_{ij}}$. □

3.3. **The function Q.** The density of Δ given X and Θ is given by:

$$p(\Delta|X, \Theta) = \prod_{i=1}^{N} p(\delta_i|\mathbf{x}_i, \Theta) = \prod_{i=1}^{N} \frac{p(\mathbf{x}_i, \delta_i|\Theta)}{p(\mathbf{x}_i|\Theta)} \tag{3}$$

where the first equality follows from the independence of the samples, and the second from the definition of conditional probability. For the purposes of this section, we do

not need to be more explicit about this density at the moment. If necessary, one may use Equation 1 and Lemma 3.1 to write out the last expression in terms of the f_j and π_j.

Remark 3.3. One can write the density of Δ given X and Θ in terms of the w_{ij} as follows. The rows of Δ are $e_{j_1}, e_{j_2}, \ldots, e_{j_N}$ for some $j_1, j_2, \ldots, j_N$, where each $j_i \in \{1, \ldots, k\}$. With this notation in hand, we may write:

$$p(\Delta|X,\Theta) = \prod_{i=1}^{N} p(e_{j_i}|\mathbf{x}_i,\Theta) = \prod_{i=1}^{N} w_{i,j_i}^{(\Theta)} \tag{4}$$

Given data $X \in \mathbb{R}^{N\times d}$, we define a function Q that takes in two sets of parameters Θ_1 and Θ_2, and gives the expected value of the log-likelihood of the Θ_1 with density of classifications specified by Θ_2:

Definition 3.4. For parameters Θ_1 and Θ_2, we set:

$$Q(\Theta_1,\Theta_2) = E_{\Delta\sim p(\Delta|X,\Theta_2)}[\mathcal{L}(\Theta_1;\Delta,X)] \tag{5}$$

In practice, the set of parameters Θ_2 will be a known estimate that we can use to compute the density of Δ explicitly. Meanwhile, the set of parameters Θ_1 will be a dummy variable. Note that in the EM algorithm, we have $Q^{(n)}(\Theta) = Q(\Theta,\Theta^{(n)})$. The 'M-step' of the EM algorithm will seek to maximize Q with respect to $\Theta_1 = \Theta$ with $\Theta_2 = \Theta^{(n)}$ fixed. The following proposition justifies Equation 2 appearing in the E-step of the EM-algorithm.

Proposition 3.5. *Given data $X \in \mathbb{R}^{N\times d}$, we have:*

$$Q(\Theta,\hat{\Theta}) = \sum_{i=1}^{N}\sum_{j=1}^{k} \left(\log f_j(\mathbf{x}_i|\theta_j) + \log \pi_j\right) w_{ij}^{(\hat{\Theta})}$$

where $\Theta = (\theta_1, \ldots, \theta_k, \pi_1, \ldots, \pi_k)$ and $\hat{\Theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_k, \hat{\pi}_1, \ldots, \hat{\pi}_k)$.

Proof. For $i = 1, \ldots, N$ and $j = 1, \ldots, k$, we make the following abbreviations:

$$c_{ij} = \log f_j(\mathbf{x}_i|\theta_j) + \log \pi_j \qquad \text{and} \qquad w_{ij} = w_{ij}^{(\hat{\Theta})} = \frac{f_j(\mathbf{x}_i|\hat{\theta}_j)\hat{\pi}_j}{\sum_{\ell=1}^{k} f_\ell(\mathbf{x}_i|\hat{\theta}_\ell)\hat{\pi}_\ell}$$

We aim to show that

$$Q(\Theta,\hat{\Theta}) = \sum_{i=1}^{N}\sum_{j=1}^{k} c_{ij} w_{ij}. \tag{6}$$

From Proposition 3.2, we have that the log-likelihood function of Θ is given by $\mathcal{L}(\Theta;\Delta,X) = \sum_{i=1}^{N}\sum_{j=1}^{k} \delta_{ij} c_{ij}$, and hence:

$$\begin{aligned} Q(\Theta,\hat{\Theta}) &= E_{\Delta\sim p(\Delta|X,\hat{\Theta})}[\mathcal{L}(\Theta;\Delta,X)] = \sum_{\Delta} \mathcal{L}(\Theta;\Delta,X)\, p(\Delta|\hat{\Theta},X) \\ &= \sum_{\Delta}\sum_{i=1}^{N}\sum_{j=1}^{k} \delta_{ij} c_{ij}\, p(\Delta|\hat{\Theta},X) \end{aligned}$$

where Δ ranges over all the possible k^N classification matrices. We now rearrange the sum so that, for each i and j, we can isolate only those Δ whose i-th row is the one-hot vector e_j:

$$\begin{aligned}\sum_{\Delta}\sum_{i=1}^{N}\sum_{j=1}^{k}\delta_{ij}c_{ij}p(\Delta|\hat{\Theta},X) &= \sum_{i=1}^{N}\sum_{j=1}^{k}c_{ij}\sum_{\Delta}\delta_{ij}p(\Delta|\hat{\Theta},X)\\ &= \sum_{i=1}^{N}\sum_{j=1}^{k}c_{ij}\sum_{\Delta:\delta_i=e_j}p(\Delta|\hat{\Theta},X)\end{aligned}$$

This is valid since $\delta_{ij} = 1$ only if $\delta_i = e_j$, and otherwise $\delta_{ij} = 0$. We now claim that $\sum_{\Delta:\delta_i=e_j} p(\Delta|\hat{\Theta},X) = w_{ij}$. To this end, we first make the following deductions:

$$\begin{aligned}\sum_{\Delta:\delta_i=e_j}p(\Delta|\hat{\Theta},X) &= \sum_{\Delta:\delta_i=e_j}\prod_{u=1}^{N}p(\delta_u|\hat{\Theta},x_u)\\ &= \sum_{\Delta:\delta_i=e_j}p(e_j|\hat{\Theta},\mathbf{x}_i)\prod_{u=1\ ;\ u\neq i}^{N}p(\delta_u|\hat{\Theta},x_u)\\ &= p(e_j|\hat{\Theta},\mathbf{x}_i)\sum_{\Delta_{-i}}p(\Delta_{-i}|\hat{\Theta},X_{-i})\end{aligned}$$

where the first equality follows from the independence of the samples; the second from the fact that we are only interested in the terms where $\delta_i = e_j$; the third from factoring, and using the notation X_{-i} to denote the matrix X with the i-th row removed, and similarly for Δ_{-i}. Now, the definition of conditional probability together with Lemma 3.1 imply that

$$p(e_j|\hat{\Theta},\mathbf{x}_i) = \frac{p(\mathbf{x}_i,e_j|\hat{\Theta})}{p(\mathbf{x}_i|\hat{\Theta})} = \frac{f_j(\mathbf{x}_i|\hat{\theta}_j)\hat{\pi}_j}{\sum_{\ell=1}^{k}f_\ell(\mathbf{x}_i|\hat{\theta}_\ell)\hat{\pi}_\ell} = w_{ij}.$$

Meanwhile, $\sum_{\Delta_{-i}} p(\Delta_{-i}|\hat{\Theta},X_{-i})$ is the sum over all values of a (discrete) density function, so is equal to one. We conclude that $\sum_{\Delta:\delta_i=e_j} p(\Delta|\hat{\Theta},X) = w_{ij}$. Therefore, $Q(\Theta,\hat{\Theta}) = \sum_{i=1}^{N}\sum_{j=1}^{k} c_{ij}w_{ij}$, which is what we wanted to show. □

References

[For19] David Forsyth, *Applied Machine Learning*, 1st ed. 2019 edition, Springer, Cham, Switzerland, 2019.