APMLA: Lecture 3

Caterina De Bacco and Isabel Valera

1 Bayesian Mixture Models

In this lecture, we are going to extend the mixture models introduced in the previous lecture to treat the mixing coefficients (and the likelihood parameters) as random variables, such that the resulting Bayesian Mixture Model (BMM) can be written as:

$$p(\mathbf{x}, z, \boldsymbol{\pi}, \boldsymbol{\Theta}) = p(\mathbf{x}|z, \boldsymbol{\Theta})p(z|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\Theta}),$$

where $\mathbf{x}$ is the observed variable, z is the component assignment, $\boldsymbol{\pi}$ is the vector of mixing coefficients and $\boldsymbol{\Theta}$ are the likelihood parameters. In the case of GMMs, $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ and $p(\mathbf{x}|z_n) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$. Here, we select as prior distribution for the mixing coefficients, $p(\boldsymbol{\pi})$, a Dirichlet distribution parametrized by the K-dimensional vector $\boldsymbol{\alpha} = (\alpha_1, \dots \alpha_K)$:

$$p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})}\prod_k \pi_k^{\alpha_k - 1}, \tag{1}$$

where $\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^K \alpha_k)}$ is the normalization constant.

Generative model. In such case, one may generate N samples from a Bayesian mixture model as:

1. Sample mixing coefficients $\boldsymbol{\pi} \sim Dir(\boldsymbol{\alpha})$
2. Sample likelihood parameters $\boldsymbol{\Theta}$ from its prior distribution.
   For example, in the case of a GMM, $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ can be sampled from a Normal-inverse-Wishart distribution, i.e., $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \sim \mathrm{NIW}(\boldsymbol{\mu}_0, \lambda, \boldsymbol{\Psi}, \nu)$. Alternatively, if we assume the covariance matrices to be equal and known for all clusters, i.e., $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_x$, the likelihood parameters correspond only to the mean vectors, i.e., $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_k\}_{k=1}^K$. In this case, we may consider a Normal prior $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$.
3. For n=1,..., N
   (a) Sample component/cluster indicator as $z_n \sim Cat(\boldsymbol{\pi})$.
   (b) Sample the observation indicator as $\mathbf{x}_n \sim p(\mathbf{x}|z_n, \boldsymbol{\Theta})$. In the case of the GMM, $p(\mathbf{x}|z_n, \boldsymbol{\Theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$

Figure 1 shows the graphical model for a Bayesian mixture model with Gaussian likelihood, i.e., a Bayesian GMM, with unknown cluster means $\boldsymbol{\mu}_k$.

In practice, we observe a dataset $\{\mathbf{x}_n\}_{n=1}^N$ (but not the cluster assigments). Then, under the assumption of a BMM as generative model for our data, we aim to infer (the posterior distribution of) the model parameters $(\boldsymbol{\pi}, \boldsymbol{\Theta})$ and also cluster assignments $\{z_n\}_{n=1}^N$. Note that the assignments $\{z_n\}_{n=1}^N$ can be interpreted as the cluster that each observation belong to, and therefore can help us to better understand the latent structure in the data. Common applications of clustering include topic modeling and community detection, where we aim to group "similar" documents and users into, topics and communities respectively.

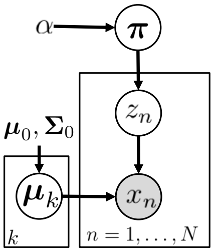

Figure 1: Graphical model for the Bayesian GMM. *Observations:* Note that conditioned on the cluster assignments $\{z_i\}_{i \neq N}$, the mixing coefficients $\boldsymbol{\pi}$ and the observations $\{\mathbf{x}_n\}_{n=1}^N$ are conditionally independent.

In the following sections, we introduce three variations of the Gibbs sampling algorithm to infer the posterior distribution of Bayesian mixture models.

2 Inferring BMMs via Gibbs sampling

In this section, we will first introduce the *Gibbs sampling* algorithm in general, and then we will show how to particularize it to approximate the posterior distribution $p\left(\pi, \boldsymbol{\Theta}, \{z_n\}_{n=1}^N | \{\mathbf{x}_n\}_{n=1}^N\right)$.

Introduction to Gibbs sampling. Gibbs sampling is a simple and widely applicable Markov chain Monte Carlo (MCMC) algorithm and can be seen as a special case of the Metropolis-Hastings algorithm with acceptance probability equal to one. For a brief and formal overview of MCMC methods please refer to Section 11.2 of Bishop (2006) or to Chapters 23&24 from **?**. Informally, Gibbs sampling is an iterative algorithm that allows to obtain samples of the $p(\boldsymbol{z}) = p(z_1, \dots, z_M)$ by iteratively sampling from the conditional distributions $p(z_i | \{z_j\}_{j \neq i})$.

More formally, consider the distribution $p(\boldsymbol{z}) = p(z_1, \dots, z_M)$ from which we wish to sample, and suppose that we have chosen some initial state for the Markov chain, i.e. an initial state (value for the variables) $\boldsymbol{z}^{(0)} = (z_1^{(0)}, \dots, z_M^{(0)})$. Here, $\boldsymbol{z}$ represents the set of random variables whose joint distribution we want to sample from, and $\boldsymbol{z}^{(0)}$ is an initial instantiation of the vector $\boldsymbol{z}$. Each step of the Gibbs sampling procedure involves replacing the value of one of the variables by a value drawn from the distribution of that variable *conditioned* on the values of the remaining variables, i.e., $z_i \sim p(z_i | \{z_j\}_{j \neq i})$. Thus we replace z_i by the new sampled value. This procedure is repeated either by cycling through the variables in some particular order or by choosing the variable to be updated at each step at random from some distribution. As an example, imagine we want to sample from the distribution $p(z_1, z_2, z_3)$ and at the τ-th iteration of the sampler, we have drawn the values $\left(z_1^{(\tau)}, z_2^{(\tau)}, z_3^{(\tau)}\right)$. Then we can obtain an $\tau + 1$ sample of $p(z_1, z_2, z_3)$ by first sampling $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)})$; then sampling $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)})$; and finally $z_3^{(\tau+1)} \sim p(z_3 | z_1^{(\tau+1)}, z_2^{(\tau+1)})$.

An important property of Gibbs sampling, like any other MCMC method, is that it ensures convergence to the target distribution, i.e., it ensures that the obtained samples $\left(z_1^{(\tau)}, z_2^{(\tau)}, z_3^{(\tau)}\right)$ with $\tau \rightarrow \infty$ are valid samples from the target distribution $p(z_1, z_2, z_3)$. In practice, one starts the iterating over the algorithm and after a burn-in period $\tau_{\text{burn-in}}$ (to be selected by the practitioner), we assume that the samples $\left(z_1^{(\tau)}, z_2^{(\tau)}, z_3^{(\tau)}\right)$ with $\tau \geq \tau_{\text{burn-in}}$ are sampled from the target distribution.

Algorithm 1. We may thus use the Gibbs sampling algorithm to approximate (using samples) the posterior distribution of our BMM, $p\left(\pi, \boldsymbol{\Theta}, \{z_n\}_{n=1}^N | \{\mathbf{x}_n\}_{n=1}^N\right)$, as follows:

1. Initialize the cluster assignments $\{z_n^{(0)}\}_{n=1}^N$ and the likelihood parameters $\boldsymbol{\Theta}^{(0)}$.

2. For $\tau = 1, \dots, N_{it}$ (N_{it} refers to the total number of iterations of gthe sampler)
    (a) Sample mixing coefficients from its posterior distribution $\boldsymbol{\pi}^{(\tau)} \sim p(\boldsymbol{\pi}|\{z_n^{(\tau-1)}\}_{n=1}^N)$.
    (b) Sample likelihood parameters from its posterior distribution $\boldsymbol{\Theta}^{(\tau)} \sim p(\boldsymbol{\Theta}|\{z_n^{(\tau-1)}\}_{n=1}^N, \{\mathbf{x}_n\}_{n=1}^N)$.
    (c) For n=1,..., N
    - Sample $z_n^{(\tau)} \sim p\left(z_n|\mathbf{x}_n, \boldsymbol{\pi}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)}\right)$, by making use of the fact that

$$p\left(z_n = k|\mathbf{x}_n, \boldsymbol{\pi}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)}\right) \propto \pi_k p(\boldsymbol{x}_n|z_n = k, \boldsymbol{\Theta}),$$

and $\sum_{k=1}^K p\left(z_n = k|\mathbf{x}_n, \boldsymbol{\pi}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)}\right) = 1$.

Note that the above algorithm requires to sample from the posterior distribution of the likelihood parameters, $p(\boldsymbol{\Theta}|\{z_n\}_{n=1}^N, \{\mathbf{x}_n\}_{n=1}^N)$, and the mixing coefficients, $p(\boldsymbol{\pi}|\{z_n\}_{n=1}^N)$.

Next we compute the posterior distribution of the mixing weights, which may be obtained as:

$$p\left(\boldsymbol{\pi}|z_1, \dots, z_N\right) = \frac{p\left(z_1, \dots, z_N|\boldsymbol{\pi}\right)p(\boldsymbol{\pi})}{p\left(z_1, \dots, z_N\right)} \quad .$$

Here, we can compute the probability of the cluster assignments $p(z_1, \dots, z_N|\boldsymbol{\pi})$ as

$$p\left(z_1, \dots, z_N|\boldsymbol{\pi}\right) = \left(\prod_n p(z_n|\boldsymbol{\pi})\right) = \frac{N!}{\prod m_k!}\prod_k \pi_k^{m_k} = Multinomial(z_1, \dots, z_N|\boldsymbol{\pi}),$$

being $m_k = \sum_{n=1}^N [z_n = k]$ the number of samples assigned to component k.

As a result, the posterior distribution of $\boldsymbol{\pi}$ can be written in closed-form as:

$$p\left(\boldsymbol{\pi}|z_1, \dots, z_N\right) = \frac{\Gamma\left(\sum_k \alpha_k + m_k\right)}{\prod_k \Gamma\left(\alpha_k + m_k\right)}\prod_k \pi_k^{\alpha_k - 1 + m_k} = \text{Dirichlet}\left(\boldsymbol{\pi}|\alpha_1 + m_1, \dots, \alpha_K + m_K\right) \quad . \tag{2}$$

In words, the posterior distribution of the mixing weights of a mixture model with Dirichlet prior and multinomial likelihood, is in turn also Dirichlet distribution. Note, that the above posterior distribution is independent of the observed variables $\{\mathbf{x}_n\}_{n=1}^N$, since as shown in the graphical model for the Bayesian Gaussian mixture model in Figure 1, the mixing coefficients $\boldsymbol{\pi}$ and the observations are independent conditioned on the component/cluster assignments $\{z_n\}_{n=1}^N$.

Let's now particularize steps (b) and (c) for GMM, with unknown cluster means $\boldsymbol{\mu}_k$ (for simplicity and without loss of generality, we here assume the covariance matrices to be equal and known for all clusters, i.e., $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_x$). In such case, our likelihood parameters are $\boldsymbol{\Theta}_k = \{\boldsymbol{\mu}_k\}_{k=1}^K$, for which we assume a Normal prior distribution, such that the posterior for the likelihood parameters is given (after the tedious calculations of the posterior derived in the lecture 1) by:

$$p\left(\boldsymbol{\mu}_k|\mathbf{x}_n, \{z_n\}_{n=1}^N\right) = \mathcal{N}\left(\boldsymbol{\mu}_k|\left(\boldsymbol{\Sigma}_0^{-1} + m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\bar{\mathbf{x}}_k\right), \left(\boldsymbol{\Sigma}_0^{-1} + m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\right)^{-1}\right), \tag{3}$$

where $m_k = \sum_{n=1}^N [z_n = k]$ is the number of observations assigned to cluster k, and $\bar{\mathbf{x}}_k = \frac{1}{m_k}\sum_{n=1}^N \mathbf{x}_n[z_n = k]$ is the empirical mean of the observations assigned to cluster k.

Finally, the cluster assignment probabilities are given in this case by

$$p\left(z_n = k|\mathbf{x}_n, \{z_i\}_{i \neq n}\right) \propto \pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{4}$$

3 Exchangeability

While in our generative model $p(z_n = k|\boldsymbol{\pi}) = \pi_k$—which implies that conditioned on $\boldsymbol{\pi}$, the samples/realizations $\{z_n\}_{n=1}^N$ are independent and identically distributes (i.i.d.)—, this is not the case when we marginalize out the mixing coefficients. In particular, we can show that the probability of assigning the n-th sample of our generative model to cluster k after marginalizing out the mixing coefficients is given by:

$$p\left(z_n|\{z_i\}_{i=1}^{n-1}\right) = \int p\left(z_n, \pi|\{z_i\}_{i=1}^{n-1}\right) d\pi = \int p(z_n|\pi)p\left(\pi|\{z_i\}_{i=1}^{n-1}\right) d\pi, \tag{5}$$

where the term $p\left(\pi|\{z_i\}_{i=1}^{n-1}\right)$ corresponds to posterior distribution of π after observing $\{z_i\}_{i=1}^{n-1}$ and can be obtained as shown in (2). Solving the integral above we obtain:

$$p\left(z_n = k|\{z_i\}_{i=1}^{n-1}\right) = \frac{\sum_{i=1}^{n-1}[z_i = k] + \alpha_k}{n - 1 + \sum_k \alpha_k}, \tag{6}$$

which, as expected, introduces statistical dependencies between the different samples of the component assignment variable z.

Question: If the component/clusters assignments are not i.i.d., should I worry about the order of the samples?

Answer: **NO**. The probability of a sequence is invariant to permutations. This property is known as ***exchangeability***.
Exercise: Proof that $p\left(z_1 = 1, z_2 = 1, z_3 = 1, z_4 = 2, z_5 = 3\right) = p\left(z_1 = 3, z_2 = 1, z_3 = 2, z_4 = 1, z_5 = 1\right)$.

Exchangeability thus allows us not to worry about the data order and treat every data point as if it were the last one that we have seen. Instead of just conditioning on the first $n-1$ data points, we can pretend the n-th data point is the last one we saw, such that

$$p\left(z_n = k|\{z_i\}_{i\neq n}\right) = \frac{\sum_{i\neq n}[z_i = k] + \alpha_k}{N - 1 + \sum_k \alpha_k}, \tag{7}$$

As a consequence, we can write the posterior probability of the n-th sample belonging to cluster k after marginalizing out the mixing coefficients π as:

$$\begin{aligned} p\left(z_n = k|\mathbf{x}_n, \{z_i\}_{i\neq n}, \boldsymbol{\Theta}\right) &\propto p\left(z_n = k|\{z_i\}_{i\neq n}\right) p(\mathbf{x}_n|z_n = k) \\ &\propto \frac{\sum_{i\neq n}[z_i = k] + \alpha_k}{N - 1 + \sum_k \alpha_k} p(\mathbf{x}_n|z_n = k, \boldsymbol{\Theta}), \end{aligned} \tag{8}$$

which needs to be normalized so that $\sum_{k=1}^{K} p\left(z_n = k|\mathbf{x}_n, \{z_i\}_{i\neq n}\right) = 1$ to be a valid probability mass function (pmf).

Algorithm 2. We may now use the above result to derive a Gibbs sampling algorithm to approximate the distribution $p\left(\boldsymbol{\Theta}, \{z_n\}_{n=1}^N|\{\mathbf{x}_n\}_{n=1}^N\right)$, as follows:

1. Initialize the cluster assignments $\{z_n^{(0)}\}_{n=1}^N$ and the model parameters $\boldsymbol{\Theta}^{(0)}$.
2. For $\tau = 1, \ldots, N_{it}$
   (a) For n=1,. . . , N
   - Sample $z_n^{(\tau)} \sim p\left(z_n = k|\mathbf{x}_n, \{z_i^{(\tau)}\}_{i<n}, \{z_i^{(\tau-1)}\}_{i>n}, \boldsymbol{\Theta}^{(\tau-1)}\right)$ as in Eq. 8.

   (b) Sample likelihood model parameters $\boldsymbol{\Theta}^{(\tau)} \sim p(\boldsymbol{\Theta}|\{z_n^{(\tau)}\}_{n=1}^N, \{\mathbf{x}_n\}_{n=1}^N)$.

Note that the above algorithm requires to sample from the posterior distribution of the likelihood parameters $p(\boldsymbol{\Theta}|\{z_n^{(\tau)}\}_{n=1}^N, \{\mathbf{x}_n\}_{n=1}^N)$.

In the case of the GMM, we can rewrite Eq. 8 as:

$$p\left(z_n = k|\mathbf{x}_n, \{z_i\}_{i\neq n}, \boldsymbol{\Theta}\right) \propto \frac{\sum_{i\neq n}[z_i = k] + \alpha_k}{N - 1 + \sum_k \alpha_k} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{9}$$

and the posterior for the likelihood parameters per cluster $\boldsymbol{\mu}_k$ (for simplicity and without loss of generality, we here assume the covariance matrices to be equal and known for all clusters, i.e., $\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}_x$) assuming a Normal prior distribution is given by:

$$p\left(\boldsymbol{\mu}_k|\mathbf{x}_n,\{z_n\}_{n=1}^N\right)=\mathcal{N}\left(\boldsymbol{\mu}_k|\left(\boldsymbol{\Sigma}_0^{-1}+m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0+m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\bar{\mathbf{x}}_k\right),\left(\boldsymbol{\Sigma}_0^{-1}+m_k\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}\right)^{-1}\right) \tag{10}$$

where $m_k=\sum_{n=1}^N[z_n=k]$ is the number of observations assigned to cluster k, and $\bar{\mathbf{x}}_k=\frac{1}{m_k}\sum_{n=1}^N\mathbf{x}_n[z_n=k]$ is the empirical mean of the observations assigned to cluster k.

Observation: The above algorithm will in general converge faster the Algorithm 1, since we do not need to iterate over the K-dimensional vector of mixing coefficients. Moreover, conditioned on a sample of the cluster assignments $(z_1,\ldots,z_N)$, we may always recover the posterior of the mixing coefficients, $p\left(\boldsymbol{\pi}|z_1,\ldots,z_N\right)$, using Eq. 2.

Unfortunately, the above algorithms may still present poor mixing properties and that require a very large number of iterations to converge, specially in high dimensional data. To solve that limitation, one may decide to *collapse* the Gibbs sampler by marginalizing out also the likelihood parameters. In such case, the target posterior distribution will corresponds to:

$$\begin{aligned}
p\left(\{z_n\}_{n=1}^N|\{\mathbf{x}_n\}_{n=1}^N\right)&=\int p\left(\boldsymbol{\Theta},\{z_n\}_{n=1}^N|\{\mathbf{x}_n\}_{n=1}^N\right)d\boldsymbol{\Theta}\\
&\propto\int p\left(\{\mathbf{x}_n\}_{n=1}^N|\{z_n\}_{n=1}^N,\boldsymbol{\Theta}\right)p(\{z_n\}_{n=1}^N)p\left(\boldsymbol{\Theta}\right)d\boldsymbol{\Theta}\\
&\propto p(\{z_n\}_{n=1}^N)\int p\left(\{\mathbf{x}_n\}_{n=1}^N|\{z_n\}_{n=1}^N,\boldsymbol{\Theta}\right)p\left(\boldsymbol{\Theta}\right)d\boldsymbol{\Theta}\\
&\propto p(\{z_n\}_{n=1}^N)p\left(\{\mathbf{x}_n\}_{n=1}^N|\{z_n\}_{n=1}^N\right)
\end{aligned}.$$

where, as in previous cases, the normalization constant cannot be computed analitically.

Algorithm 3. We can thus again rely on Gibbs sampling to approximate the marginal posterior distribution $p\left(\{z_n\}_{n=1}^N|\{\mathbf{x}_n\}_{n=1}^N\right)$ as:

1. Initialize the cluster assignments $\{z_n^{(0)}\}_{n=1}^N$.
2. For $\tau=1,\ldots,N_{it}$
   (a) For n=1,…, N
   - Sample $z_n^{(\tau)}\sim p\left(z_n=k|\{\mathbf{x}_n\}_{n=1}^N,\{z_i^{(\tau)}\}_{i<n},\{z_i^{(\tau-1)}\}_{i>n}\right)$, which is given by

   $$p\left(z_n=k|\{\mathbf{x}_n\}_{n=1}^N,\{z_i\}_{i\neq n}\right)\propto p\left(z_n=k|\{z_i\}_{i\neq n}\right)p(\mathbf{x}_n|z_n=k,\{\boldsymbol{x}_i,z_i\}_{i\neq n}), \tag{11}$$

   where the $p\left(z_n=k|\{z_i\}_{i\neq N}\right)$ is already computed in Eq. 7 and $p(\mathbf{x}_n|z_n=k,\{\boldsymbol{x}_i,z_i\}_{i\neq n})$ is the marginal likelihood (a.k.a., *posterior predictive*).

For a GMM with unknown cluster means $\boldsymbol{\mu}_k\ \sim(\boldsymbol{\mu}_0=0,\boldsymbol{I})$, and the covariance matrices to be equal and known for all clusters, i.e., $\boldsymbol{\Sigma}_k=\sigma_x^2\boldsymbol{I}$, the posterior predictive can be computed as:

$$\begin{aligned}
p(\mathbf{x}_n|z_n=k,\{\boldsymbol{x}_i,z_i\}_{i\neq n})&\propto\int\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\sigma_x^2\boldsymbol{I})\mathcal{N}\left(\boldsymbol{\mu}_k|\frac{m_k}{m_k+\sigma_x^2}\bar{\mathbf{x}}_k,(m_k+\sigma_x^2)^{-1}\boldsymbol{I}\right)d\boldsymbol{\mu}_k\\
&=\mathcal{N}\left(\boldsymbol{\mu}_k|\frac{m_k}{m_k+\sigma_x^2}\bar{\mathbf{x}}_k,\frac{\sigma_x^2}{m_k+\sigma_x^2}\boldsymbol{I}\right),
\end{aligned} \tag{12}$$

where here $m_k=\sum_{i\neq n}[z_i=k]$ and $\bar{\mathbf{x}}_k=\frac{1}{m_k}\sum_{i\neq n}[z_i=k]\mathbf{x}_i$ do not account for the n-th sample.

References

C. M. Bishop, *Pattern recognition and machine learning* (Springer, 2006).