# Machine Learning: Lecture 20

Isabel Valera

(Notes from Advanced Probabilistic ML course by Prof. Valera)

## 1 Gaussian Mixture Models (GMMs)

**Gaussian Mixture Distribution.** In general, the superposition of $K$ Gaussian distributions can be formulated as the following probabilistic model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1}$$

which is called *Gaussian mixture distribution*. Here, each Gaussian density $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sigma_x)$ is often referred as a *component* of the mixture and is characterized by its mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$; the parameters $\pi_k$ are called *mixing coefficients* and should fulfil $\sum_{k=1}^{K} \pi_k = 1$ (with $0 \leq \pi_k \leq 1$) in order for the resulting $p(\mathbf{x})$ to be a valid probability density function (pdf), i.e. $p(\mathbf{x}) \geq 0$ for all $\mathbf{x}$ and $\int p(\mathbf{x}) d\mathbf{x} = 1$ (refer to Section 2.3.9 of Bishop (2006) for details on the proof). We also remark that the mixing coefficients $\pi_k$ correspond to the prior probability of *picking* the $k$-th component in the mixture.

**Gaussian Mixture Model (GMM).** Let us know introduce a categorical latent variable $z \in \{1, \dots, K\}$, such that the joint distribution of the observed variable $\mathbf{x}$ and the latent variable $z$ factorizes as:

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

where $p(z = k) = \pi_k$ and $p(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Alternatively, we can write $p(\mathbf{x}|z) = \prod_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{[z=k]}$, where $[z = k]$ returns one iff $z$ takes value $k$, and zero otherwise.

The marginal distribution of the observed variable $\mathbf{x}$ is given by:

$$p(\mathbf{x}) = \sum_{z=1,\dots,K} p(\mathbf{x}, z) = \sum_{z=1,\dots,K} p(z)p(\mathbf{x}|z) = \sum_{k=1,\dots,K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

which shows that the marginal distribution of $\mathbf{x}$ is indeed a Gaussian mixture distribution as defined in Eq. (1).

As a result, one can easily generate samples $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ from a Gaussian mixture distribution by using the generative process of the GMM that consist in two steps: i) first sample each latent variable $z_n$ from a Categorical distribution with category probabilities $\boldsymbol{\pi} = (\pi_1, \dots \pi_k)$; ii) then sample the corresponding observation $\mathbf{x}_n$ from $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma}_{z_n})$. The graphical model corresponding to this generative model is shown in Figure 3.

## 2 MLE for GMMs: Introduction to the E-M algorithm

Next, we assume that we observed N i.i.d. samples from a GMM with unknown parameters, and show how to obtain the MLE solution for the parameters of the GMM.

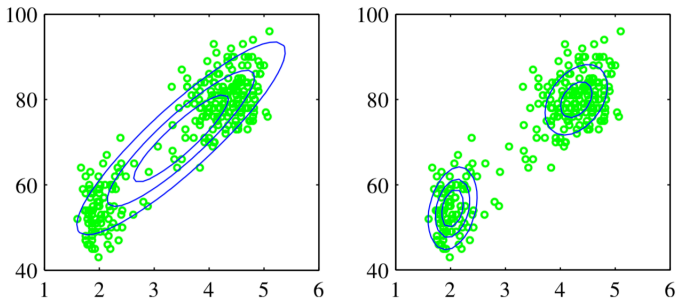


**Figure 1:** The 'Old Faithful' dataset (Figure 2.21 from Bishop (2006)).

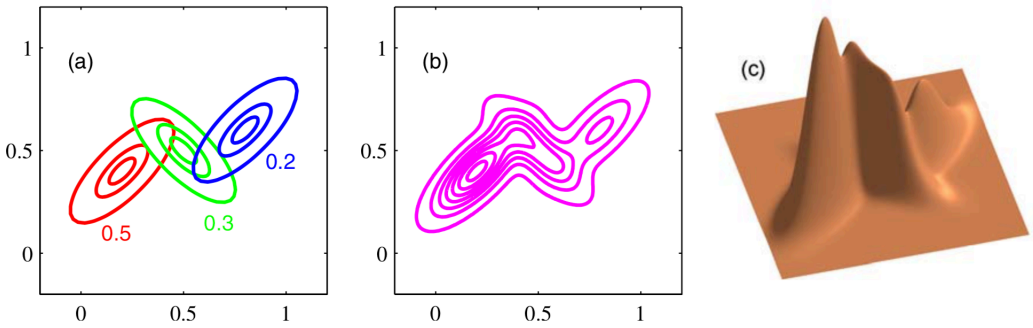


**Figure 2:** Illustration of a mixture of 3 Gaussians in a two-dimensional space (Figure 2.23 from Bishop (2006)).

**Maximum Likelihood Estimation.** As shown before, the Gaussian mixture distribution is characterized by the parameters $\boldsymbol{\pi} = (\pi_1, \dots \pi_k)$, $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. One possibility is to estimate them by *maximum likelihood estimation* (MLE) given the observed dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ as:

$$\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K = \underset{\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K), \tag{2}$$

where

$$\mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K) = \ln p(\mathbf{X}|\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K) = \sum_{n=1}^N \ln \left( \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \tag{3}$$

**Obs1:** unfortunately, in contrast to the MLE solution for the Gaussian distribution, the above optimization problem does not have closed-form solution (since there is a sum inside the logarithm). **Obs2:** there exist more than one equivalent MLE solutions. Specifically, for a $K$-component mixture we will have a total of $K!$ equivalent MLE solutions corresponding to the $K!$ ways of assigning $K$ sets of parameters to $K$ components.

Nevertheless, as done for the Gaussian distribution, we set the derivative with respect to the mean parameter $\boldsymbol{\mu}_k$ to zero, i.e.,

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K)}{\partial \mu_k} = \sum_{n=1}^N \gamma_k(\mathbf{x}_n) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \equiv 0,$$

where we have defined

$$\gamma_k(\mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4}$$

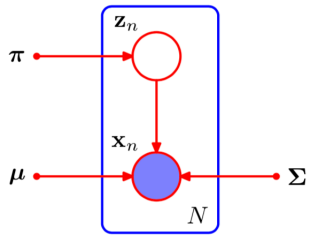


**Figure 3:** Graphical model for the GMM (Figure 9.6 from Bishop (2006)).

which corresponds to the posterior probability of $z_n = k$, i.e., the posterior probability that the observation $\mathbf{x}_n$ has been sampled from the component (cluster) $k$.

Then we can write the MLE result for $\boldsymbol{\mu}_k$ as:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n)\, \mathbf{x}_n, \tag{5}$$

where we have defined

$$N_k = \sum_{n=1}^{N} \gamma_k(\mathbf{x}_n), \tag{6}$$

which can be interpreted as the effective number of points assigned to cluster $k$.

Following a similar procedure for the covariance matrix, we obtain:

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{z_n}(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top. \tag{7}$$

Finally, we also would like to find the mixing coefficients $\boldsymbol{\pi}$ that maximize the log-likelihood. However, in this case we need to ensure that $\sum_{k=1}^{K} \pi_k = 1$. We do so by using a Lagrange multiplier to account for the constraint directly in the objective function as:

$$\hat{\mathcal{L}}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}, \lambda) := \mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}) + \lambda \left( \sum_{k=1}^{K} \pi_k - 1 \right).$$

We now take the derivative with respect to $\pi_k$ and set it to zero:

$$\frac{\partial \hat{\mathcal{L}}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}, \lambda)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda = \sum_{n=1}^{N} \frac{\gamma_k(\mathbf{x}_n)}{\pi_k} + \lambda \equiv 0 \quad .$$

If we now multiply both sides of the above expression by $\pi_k$ and take the sum over $k$ to enforce the constraint $\sum_{k=1}^{K} \pi_k = 1$ and $\sum_{k=1}^{K} N_k = N$, we find that $\lambda = -N$. Substituting it inside Eq. (8) and multiplying each term by $\pi_k$ (this time without taking the sum over $k$) yields:

$$\pi_k = \frac{N_k}{N}. \tag{8}$$

**Obs**: notice that Equations 5, 7 and 8 do not lead to a closed-form solution since they all depend on the posterior probability of $z_n = k$ given $\mathbf{x}_n$, i.e. $\gamma_k(\mathbf{x}_n)$, which is also unknown. Remarkably, $\gamma_k(\mathbf{x}_n)$ in turn depends on the likelihood and therefore on the parameters we are trying to estimate. Thus the question that arise here is, *how can we obtain the MLE of parameters of a Gaussian mixture distribution?*

**Expectation-Maximization (EM) algorithm.** A good alternative is to propose an *iterative* algorithm for finding a solution to the MLE problem: informally, this iterates between two steps: i) estimating the posterior probabilities $\{\gamma_k(\mathbf{x}_n)\}_{k=1}^K$ *conditioned* on the current values of the parameters $\boldsymbol{\pi} = (\pi_1, \dots \pi_k)$, $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$; ii) then maximize the parameters $\boldsymbol{\pi} = (\pi_1, \dots \pi_k)$, $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ *conditioned* on the current estimates of $\{\gamma_k(\mathbf{x}_n)\}_{k=1}^K$. This algorithm results indeed in the particularisation of the Expectation-Maximization (EM) algorithm for the GMM.

More specifically, one may find an MLE solution for the GMM by the following Algorithm:

1. Initialize the GMM parameters $\boldsymbol{\pi} = (\pi_1, \dots \pi_k)$, $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ and evaluate the log-likelihood $\mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K)$.
2. **E-step.** Evaluate the posterior probabilities for $z_n = k$ for all $k$, i.e., $\{\gamma_k(\mathbf{x}_n)\}_{k=1}^K$ according to Eq. 4.
3. **M-step.** For the new value of $\{\gamma_k(\mathbf{x}_n)\}_{k=1}^K$, re-estimate the GMM parameters as:
   (a) Update $\{\boldsymbol{\mu}_k^{new}\}_{k=1}^K$ as in Eq. 5.
   (b) Update $\{\boldsymbol{\Sigma}_k^{new}\}_{k=1}^K$ as in Eq. 7, using the new values for the mean parameters $\{\boldsymbol{\mu}_k^{new}\}_{k=1}^K$.
   (c) Update the probabilities $\{\pi_k^{new}\}_{k=1}^K$ as in Eq. 8.
4. Evaluate the log-likelihood $\mathcal{L}(\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K)$ and check for convergence (in log-likelihood or parameter estimates). If the convergence criterium is not achieved, return to step 2.

# 3 The E-M algorithm, in general

In general, the E-M algorithm may be applied to find the MLE solution $\theta_{MLE} = \text{argmax}_{\boldsymbol{\theta}} \ln p(\mathbf{X}|\boldsymbol{\theta})$, where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ are the data and $\boldsymbol{\theta}$ the model parameters. To this end, we just need to assume a generative model $p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$, where $\mathbf{Z}$ is the set of latent variables (e.g. in a mixture model, the component/cluster assignments $\mathbf{Z} = (z_1, \dots, z_N)$), such that the log-likelihood is given by

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \quad .$$

Note though that the set of latent variables $\mathbf{Z}$ is unknown; we can only access them through their posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$.

Therefore, as in the GMM case, we can iterate between the following two steps:

1. **E-step.** Estimate the log-likelihood of some general parameters $\boldsymbol{\theta}$ by taking the expectation with respect to the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})$ as

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) \geq \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}),$$

   where $\boldsymbol{\theta}^{old}$ are the parameters from the previous iteration[1].
2. **M-step.** Update the model parameters via log-likelihood maximization as

$$\boldsymbol{\theta}^{new} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}),$$

   and set $\boldsymbol{\theta}^{old} = \boldsymbol{\theta}^{new}$.

[1]The above inequality can be obtained by taking into account that the logarithm is a concave function and then using Jensen's inequality. As $Q$ is a lower-bound of $\ln p(\mathbf{X}|\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$, maximizing $Q$ in turns maximizes $\ln p(\mathbf{X}|\boldsymbol{\theta})$ . The above inequality comes from the derivation by Dempster et al. (1977): "Maximum likelihood from incomplete data via the EM algorithm."

**Obs1**: note that the E-M algorithm assumes a tractable E-step.

**Obs2**: the E-M algorithm ensures that at each iteration the log-likelihood is increased until convergence to a local optimum (this can be rigorously proven, see e.g. Bishop (2006)). This does not guarantee to reach the *global* optimum though. In practice, one runs the EM algorithm $R$-times with different parameter initialisations, which can potentially converge to $R$ different local optima. Finally, one chooses, for instance, the one with best likelihood at convergence.

**Obs3**: the E-M algorithm can be also used to find the MAP solution assuming a prior distribution $p(\boldsymbol{\theta})$. In such case, the M-step maximizes $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \ln p(\boldsymbol{\theta})$ (note: $p(\boldsymbol{\theta}|\mathbf{X}) \propto p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ ).

# References

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