Mixture Models and EM

STATS 305B: Applied Statistics II

Scott Linderman

February 10, 2025

Recap

We have a lot of tools in our toolkit now! We learned about exponential family distributions, which form the building blocks of more complex models.

We also learned about Bayesian inference algorithms, like MCMC and VI, to infer posterior distributions of more complex models

Today we'll start learning about **latent variable models** for complex datasets. We'll start with the simplest latent variable model — **mixture models**.

Outline

- Specifying a mixture model
- K-Means and MAP estimation
- Expectation Maximization (EM) and MLE
- Connecting EM and VI

Motivation

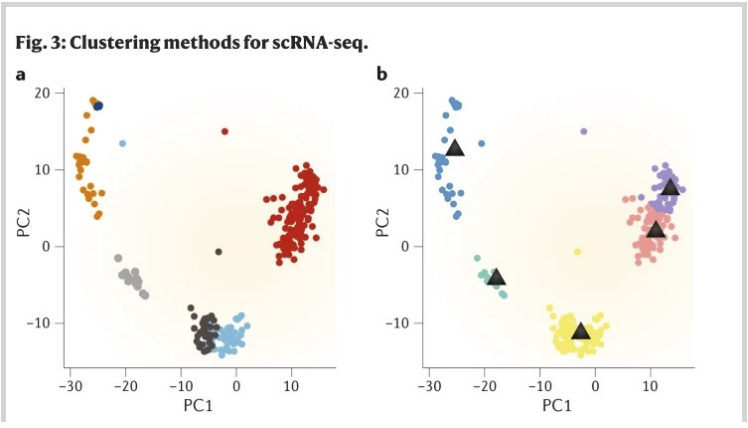

Figure: Single Cell RNA Sequencing. *From {cite:t}*`Kiselev2019-bt`.

Motivation

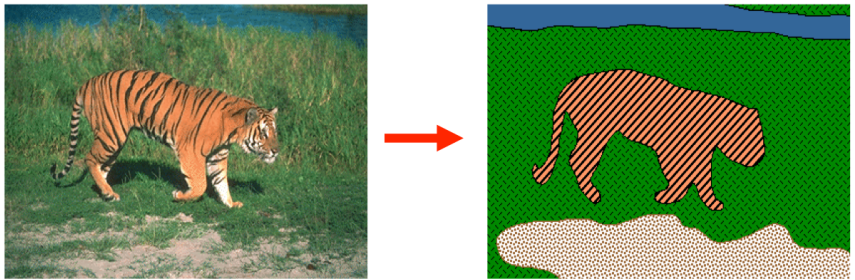

Figure: Foreground/background segmentation. *From https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html*

Motivation

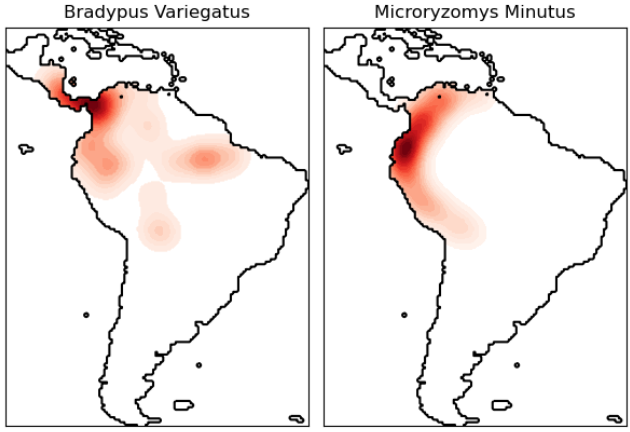

Figure: Kernel density estimate. *From Scikit-learn KDE Demo.*

Mixture Models

Let,

- N denote the number of data points
- K denote the number of mixture components (i.e., clusters)
- $\boldsymbol{x}_n \in \mathbb{R}^D$ denote the n-th data point
- $z_n \in \{1, \ldots, K\}$ be a latent variable denoting the cluster assignment of the n-th data point

The model is parameterized by,

- $\boldsymbol{\theta}_k$, the natural parameters of cluster k.
- $\boldsymbol{\pi} \in \Delta_{K-1}$, the cluster proportions (probabilities).

Mixture Models

The generative model is as follows:

1. Sample the assignment of each data point:

$$z_n \overset{\text{iid}}{\sim} \text{Cat}(\boldsymbol{\pi}) \quad \text{for } n = 1, \ldots, N$$

2. Sample data points given their assignments:

$$\boldsymbol{x}_n \sim p(\boldsymbol{x} \mid \boldsymbol{\theta}_{z_n}) \quad \text{for } n = 1, \ldots, N$$

Joint distribution

The joint distribution of the data and latent variables is,

$$
\begin{aligned}
p(\{(z_n, \boldsymbol{x}_n)\}_{n=1}^N) &= \prod_{n=1}^N \mathrm{Cat}(z_n \mid \pi)\, p(\boldsymbol{x}_n \mid z_n) \\
&= \prod_{n=1}^N \prod_{k=1}^K [\pi_k\, p(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k)]^{\mathbb{I}[z_n = k]}
\end{aligned}
$$

Exponential family mixture models

Assume an exponential family likelihood of the form,

$$p(\boldsymbol{x} \mid \boldsymbol{\eta}_k) = h(\boldsymbol{x}_n) \exp\left\{\langle t(\boldsymbol{x}_n), \boldsymbol{\eta}_k \rangle - A(\boldsymbol{\eta}_k)\right\}$$

Example: Gaussian Mixture Model (GMM)

Assume the conditional distribution of $\boldsymbol{x}_n$ is a Gaussian with mean $\boldsymbol{\theta}_k \in \mathbb{R}^D$ and identity covariance:

$$p(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k) = \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I})$$

Since we are assuming identity covariance, the sufficient statistics of the Gaussian are $t(\boldsymbol{x}) = \boldsymbol{x}$ and the natural parameters are simply $\boldsymbol{\eta}_k = \boldsymbol{\theta}_k$.

Two Inference Algorithms

Let's stick with the Gaussian mixture model for now. Suppose we observe data points $\{\boldsymbol{x}_n\}_{n=1}^N$ and want to infer the assignments $\{z_n\}_{n=1}^N$ and estimate the means $\{\boldsymbol{\theta}_k\}_{k=1}^K$. Here are two intuitive algorithms.

K-Means

Suppose we knew the cluster assignments, z_n. Then it would be straightforward to estimate the cluster means: we could simply use the maximum likelihood estimate, $\hat{\boldsymbol{\theta}}_{\text{MLE}} = \frac{1}{N_k} \sum_{n:z_n=k} \boldsymbol{x}_n$, where $N_k = \sum_n \mathbb{I}[z_n = k]$ is the number of data poitns in cluster k.

Likewise, if we knew the cluster means, it would be straightforward to compute the *maximum a posteriori* (MAP) estimate of z_n: we would assign each data point to the nearest cluster. If we alternate these two steps, we obtain the **K-Means algorithm**:

K-Means

Repeat until convergence:

1. For each $n = 1, \ldots, N$, fix the means $\boldsymbol{\theta}$ and set,

$$\begin{aligned} z_n &= \hat{z}_{n,\text{MAP}} \\ &= \arg \max_{k \in \{1,\ldots,K\}} \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I}) \\ &= \arg \min_{k \in \{1,\ldots,K\}} \|\boldsymbol{x}_n - \boldsymbol{\theta}_k\|_2 \end{aligned}$$

2. For each $k = 1, \ldots, K$, fix all assignments $\boldsymbol{z}$ and set,

$$\begin{aligned} \boldsymbol{\theta}_k &= \hat{\boldsymbol{\theta}}_{\text{MLE}} \\ &= \frac{1}{N_k} \sum_{n=1}^{K} \mathbb{I}[z_n = k] \boldsymbol{x}_n \end{aligned}$$

Comments on K-Means

- **Question:** What does this algorithm implicitly assume about the cluster probabilities π? How would you modify this algorithm to incorporate and estimate π?
- **Connection between K-Means and MAP estimation** Note that if we put an improper uniform prior on θ_k, we could think of this entire algorithm as MAP estimation of $\boldsymbol{\theta}$ and $\boldsymbol{z}$ via **coordinate ascent**!
- K-Means made **hard assignments** of data points to clusters in each iteration. That sounds a little extreme — do you really want to attribute a datapoint to a single class when it is right in the middle of two clusters? What could we do instead?

Expectation Maximization (EM) for a GMM

Repeat until convergence:

1. For each data point n and component k, compute the **responsibility**:

$$\omega_{nk} = \frac{\pi_k \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I})}{\sum_{j=1}^{K} \pi_j \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_j, \boldsymbol{I})}$$

2. For each component k, update the mean:

$$\boldsymbol{\theta}_k^\star = \frac{1}{N_k} \sum_{n=1}^{K} \omega_{nk} \boldsymbol{x}_n$$

This is the **Expectation-Maximization (EM) algorithm**. As we will show, EM yields an estimate that maximizes the *marginal* likelihood of the data.

Theoretical Motivation for EM

Rather than maximizing the **joint probability**, EM is maximizing the **marginal probability**,

$$\begin{aligned}\log p(\{\boldsymbol{x}_n\}_{n=1}^N; \boldsymbol{\theta}) &= \log \sum_{z_1=1}^K \cdots \sum_{z_N=1}^K p(\{\boldsymbol{x}_n, z_n\}_{n=1}^N; \boldsymbol{\theta}) \\ &= \log \prod_{n=1}^N \sum_{z_n=1}^K p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) \\ &= \sum_{n=1}^N \log \sum_{z_n=1}^K p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})\end{aligned}$$

For discrete mixtures (with small enough K) we can evaluate the log marginal probability. We can usually evaluate its gradient too, so we could just do gradient ascent to find $\boldsymbol{\theta}^*$. However, EM typically obtains faster convergence rates.

The ELBO from another angle

The key idea is to obtain a lower bound on the marginal probability,

$$\begin{aligned}\log p(\{\boldsymbol{x}_n\}_{n=1}^N; \boldsymbol{\theta}) &= \sum_{n=1}^N \log \sum_{z_n} p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) \\ &= \sum_{n=1}^N \log \sum_{z_n} q(z_n) \frac{p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})}{q(z_n)} \\ &= \sum_{n=1}^N \log \mathbb{E}_{q(z_n)} \left[\frac{p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})}{q(z_n)} \right]\end{aligned}$$

where $q(z_n)$ is any distribution on $z_n \in \{1, \ldots, K\}$ such that $q(z_n)$ is **absolutely continuous** w.r.t. $p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})$.

The ELBO from another angle

Jensen's Inequality: states that,

$$f(\mathbb{E}[Y]) \geq \mathbb{E}\left[f(Y)\right]$$

if f is a **concave function**, with equality iff f is linear.

Applied to the log marginal probability, Jensen's inequality yields,

$$\begin{aligned}
\log p(\{\boldsymbol{x}_n\}_{n=1}^N; \boldsymbol{\theta}) &= \sum_{n=1}^N \log \mathbb{E}_{q_n}\left[\frac{p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})}{q_n(z_n)}\right] \\
&\geq \sum_{n=1}^N \mathbb{E}_{q_n}\left[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) - \log q_n(z_n)\right] \\
&\triangleq \mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}]
\end{aligned}$$

where $\boldsymbol{q} = (q_1, \ldots, q_N)$ is a tuple of distributions, one for each latent variable z_n.

The ELBO from another angle

This is called the **evidence lower bound**, or **ELBO** for short. It is a function of $\boldsymbol{\theta}$ and a *functional* of $\boldsymbol{q}$, since each q_n is a probability density function. We can think of *EM as coordinate ascent on the ELBO*, alternating between updating the parameters $\boldsymbol{\theta}$ and the posteriors $q.

M-step: Gaussian case

Suppose we fix $\boldsymbol{q}$. Since each z_n is a discrete latent variable, q_n must be a probability mass function. Let it be denoted by,

$$q_n = [\omega_{n1}, \ldots, \omega_{nK}]^\top.$$

(These will be the **responsibilities** from before.)

Note that $q_n(k) = \Pr(z_n = k) = \mathbb{E}[\mathbb{I}[z_n = k]] = \omega_{nk}$ is the probability that the n-th data point belongs to cluster k.

M-step: Gaussian case

Now, recall our basic model, $\boldsymbol{x}_n \sim \mathrm{N}(\boldsymbol{\theta}_{z_n}, \boldsymbol{I})$,

$$
\begin{aligned}
\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \sum_{n=1}^{N} \mathbb{E}_{q_n}[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})] + c \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_n}[\mathbb{I}[z_n = k] \log p(\boldsymbol{x}_n, \boldsymbol{\theta}_k)] + c \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_{nk} \log p(\boldsymbol{x}_n, \boldsymbol{\theta}_k) + c \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_{nk} \left[\boldsymbol{x}_n^\top \boldsymbol{\theta}_k - \tfrac{1}{2} \boldsymbol{\theta}_k^\top \boldsymbol{\theta}_k \right] + c
\end{aligned}
$$

M-step: Gaussian case

Zooming in on just $\boldsymbol{\theta}_k$,

$$\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] = \boldsymbol{h}_k^\top \boldsymbol{\theta}_k - \tfrac{1}{2} \boldsymbol{\theta}_k^\top \boldsymbol{J}_k \boldsymbol{\theta}_k$$

where

$$\boldsymbol{h}_k = \sum_{n=1}^{N} \omega_{nk} \boldsymbol{x}_n \qquad \boldsymbol{J}_k = \sum_{n=1}^{N} \omega_{nk} \boldsymbol{I}$$

Taking derivatives and setting to zero yields,

$$\boldsymbol{\theta}_k^\star = \boldsymbol{J}_k^{-1} \boldsymbol{h}_k = \frac{\sum_{n=1}^{N} \omega_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} \omega_{nk}}.$$

These are the same as the EM updates shown above!

E-step: Gaussian case

As a function of q_n, for discrete Gaussian mixtures with identity covariance,

$$\begin{aligned}\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \mathbb{E}_{q_n}\left[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) - \log q_n(z_n)\right] + c \\ &= \sum_{k=1}^{K} \omega_{nk}\left[\log \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I}) + \log \pi_k - \log \omega_{nk}\right] + c\end{aligned}$$

where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^\top$ is the vector of prior cluster probabilities.

We also have two constraints: $\omega_{nk} \geq 0$ and $\sum_k \omega_{nk} = 1$. Let's ignore the non-negative constraint for now (it will automatically be satisfied anyway) and write the Lagrangian with the simplex constraint,

$$\mathscr{J}(\boldsymbol{\omega}_n, \lambda) = \sum_{k=1}^{K} \omega_{nk}\left[\log \mathrm{N}(\boldsymbol{x}_n \mid \theta_k, \boldsymbol{I}) + \log \pi_k - \log \omega_{nk}\right] - \lambda\left(1 - \sum_{k=1}^{K} \omega_{nk}\right)$$

E-step: Gaussian case

Taking the partial derivative wrt ω_{nk} and setting to zero yields,

$$\begin{aligned}\frac{\partial}{\partial \omega_{nk}} \mathscr{L}(\boldsymbol{\omega}_n, \lambda) &= \log \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I}) + \log \pi_k - \log \omega_{nk} - 1 + \lambda = 0 \\ \Rightarrow \log \omega^\star_{nk} &= \log \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I}) + \log \pi_k + \lambda - 1 \\ \Rightarrow \omega^\star_{nk} &\propto \pi_k \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I})\end{aligned}$$

Enforcing the simplex constraint yields,

$$\omega^\star_{nk} = \frac{\pi_k \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k, \boldsymbol{I})}{\sum_{j=1}^K \pi_j \mathrm{N}(\boldsymbol{x}_n \mid \boldsymbol{\theta}_j, \boldsymbol{I})},$$

just like above.

Note that

$$\omega^\star_{nk} \propto p(z_n = k)\, p(\boldsymbol{x}_n \mid z_n = k, \boldsymbol{\theta}) = p(z_n = k \mid \boldsymbol{x}_n, \boldsymbol{\theta}).$$

That is, the responsibilities equal the posterior probabilities!

The ELBO is tight after the E-step

Equivalently, q_n equals the posterior, $p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$. At that point, the ELBO simplifies to,

$$
\begin{aligned}
\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \sum_{n=1}^{N} \mathbb{E}_{q_n}\left[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) - \log q_n(z_n)\right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})}\left[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta}) - \log p(z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta})\right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{p(z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta})}\left[\log p(\boldsymbol{x}_n; \boldsymbol{\theta})\right] \\
&= \sum_{n=1}^{N} \log p(\boldsymbol{x}_n; \boldsymbol{\theta}) \\
&= \log p(\{\boldsymbol{x}_n\}_{n=1}^{N}, \boldsymbol{\theta})
\end{aligned}
$$

The ELBO is tight after the E-step

We can view the EM algorihtm as a **minorize-maximize (MM)** algorithm where we iteratively lower bound the ELBO and and then maximize the lower bound.

M-step: General Case

Now let's consider the general Bayesian mixture with exponential family likelihoods.

While we're at it, let's also add conjugate priors $p(\boldsymbol{\theta})$ with pseudo-counts ν and pseudo-observations $\boldsymbol{\chi}$. As a function of $\boldsymbol{\theta}$,

$$\begin{aligned}
\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \mathbb{E}_{q_n}[\log p(\boldsymbol{x}_n, z_n \mid \boldsymbol{\theta})] + \mathrm{c} \\
&= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_{nk} \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta}_k) + \mathrm{c} \\
&= \sum_{k=1}^{K} \left[\boldsymbol{\chi}^\top \boldsymbol{\theta}_k - \nu A(\boldsymbol{\theta}_k) \right] + \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_{nk} \left[t(\boldsymbol{x}_n)^\top \boldsymbol{\theta}_k - A(\boldsymbol{\theta}_k) \right] + \mathrm{c}
\end{aligned}$$

M-step: General Case

Zooming in on just $\boldsymbol{\theta}_k$,

$$\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] = \boldsymbol{\chi}_k'^{\top} \boldsymbol{\theta}_k - \nu_k' A(\boldsymbol{\theta}_k)$$

where

$$\boldsymbol{\chi}_k' = \boldsymbol{\chi} + \sum_{n=1}^{N} \omega_{nk} t(\boldsymbol{x}_n) \qquad \nu_k' = \nu + \sum_{n=1}^{N} \omega_{nk}$$

Taking derivatives and setting to zero yields,

$$\boldsymbol{\theta}_k^* = [\nabla A]^{-1} \left(\frac{\boldsymbol{\chi}_k'}{\nu_k'} \right)$$

Recall that $\nabla A^{-1} : \mathscr{M} \mapsto \Omega$ is a mapping from mean parameters to natural parameters (and the inverse exists for minimal exponential families). Thus, the generic M-step above amounts to finding the natural parameters $\boldsymbol{\theta}_k^*$ that yield the expected sufficient statistics $\boldsymbol{\chi}_k' / \nu_k'$ by inverting the gradient mapping.

E-step: General Case

In our first pass, we assumed q_n was a finite pmf. More generally, q_n will be a probability density function, and optimizing over functions usually requires the *calculus of variations*. (Ugh!)

However, note that we can write the ELBO in a slightly different form,

$$\begin{aligned}
\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \mathbb{E}_{q_n}\left[\log p(\boldsymbol{x}_n, z_n \mid \boldsymbol{\theta}) - \log q_n(z_n)\right] \\
&= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \mathbb{E}_{q_n}\left[\log p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) + \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta}) - \log q_n(z_n)\right] \\
&= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} \left[\log p(\boldsymbol{x}_n \mid \boldsymbol{\theta}) - D_{\mathrm{KL}}\left(q_n(z_n) \,\|\, p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})\right)\right] \\
&= \log p(\{\boldsymbol{x}_n\}_{n=1}^{N}, \boldsymbol{\theta}) - \sum_{n=1}^{N} D_{\mathrm{KL}}\left(q_n(z_n) \,\|\, p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})\right)
\end{aligned}$$

E-step: General Case

where $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ denote the **Kullback-Leibler divergence**. (Note that we included the prior on $\boldsymbol{\theta}$ since we are treating it as a random variable with a prior in this general case.)

Recall, the KL divergence is defined as,

$$D_{\mathrm{KL}}(q(z) \,\|\, p(z)) = \int q(z) \log \frac{q(z)}{p(z)} \, \mathrm{d}z.$$

It gives a notion of how similar two distributions are, but it is *not a metric!* (It is not symmetric.) Still, it has some intuitive properties: 1. It is non-negative, $D_{\mathrm{KL}}(q(z) \,\|\, p(z)) \geq 0$. 2. It equals zero iff the distributions are the same, $D_{\mathrm{KL}}(q(z) \,\|\, p(z)) = 0 \iff q(z) = p(z)$ almost everywhere.

E-step: General Case

Maximizing the ELBO wrt q_n amounts to minimizing the KL divergence to the posterior $p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$,

$$\begin{aligned}\mathscr{L}[\boldsymbol{\theta}, \boldsymbol{q}] &= \log p(\boldsymbol{\theta}) + \sum_{n=1}^{N} [\log p(\boldsymbol{x}_n \mid \boldsymbol{\theta}) - D_{\mathrm{KL}}(q_n(z_n) \,\|\, p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}))] \\ &= -D_{\mathrm{KL}}(q_n(z_n) \,\|\, p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})) + c\end{aligned}$$

As we said, the KL is minimized when $q_n(z_n) = p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$, so the optimal update is,

$$q_n^\star(z_n) = p(z_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}),$$

just like we found above.

Conclusion

Mixture models are basic building blocks of statistics, and our first encounter with **discrete latent variable models (LVMs)**. (Where have we seen continuous LVMs so far?) Mixture models have widespread uses in both density estimation (e.g., kernel density estimators) and data science (e.g., clustering). Next, we'll talk about how to extend mixture models to cases where the cluster assignments are correlated in time.