Deep Learning - Foundations and Concepts

Chapter 15. Discrete Latent Variables

nonlineark@github

April 6, 2025

Outline

K-means clustering

Problem

Suppose we have a data set $\{x^1, \ldots, x^N\}$ consisting of N observations of a D-dimensional Euclidean variable x. Partition the data set into some number K of clusters, where we will suppose for the moment that the value of K is given.

K-means clustering

Problem'

Find:

- K cluster centers: $\mu_1, \ldots, \mu_K \in \mathbb{R}^D$.
- N data point assignment: $r^1, \ldots, r^N \in \{\mathrm{e}_1, \ldots, \mathrm{e}_K\}$.

such that the error function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^n ||x^n - \mu_k||^2$$

which represents the sum of the squares of the distances of each data point to its assigned cluster center, is minimized.

K-means clustering

We can do this through an iterative procedure:

1. Choose some initial values for the $\{\mu_k\}$.
2. E step: Minimize J with respect to the $\{r_k^n\}$, keeping the $\{\mu_k\}$ fixed.
3. M step: Minimize J with respect to the $\{\mu_k\}$, keeping the $\{r_k^n\}$ fixed.
4. Go to step 2 until convergence.

K-means clustering

Consider the E step. It's easy to see that we should assign the nth data point to the closest cluster center:

$$r_k^n = \begin{cases} 1, & \text{if } k = \text{argmin}_j ||x^n - \mu_j||^2 \\ 0, & \text{otherwise} \end{cases}$$

For the M step:

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{n=1}^{N} r_k^n (x^n - \mu_k)^T$$

$$\mu_k = \frac{\sum_{n=1}^{N} r_k^n x^n}{\sum_{n=1}^{N} r_k^n}$$

so μ_k is equal to the mean of all the data points x_n assigned to cluster k.

K-means clustering

Algorithm 1: K-means algorithm

$\{r_k^n\} \leftarrow 0$;
repeat
  $\{^{\text{old}}r_k^n\} \leftarrow \{r_k^n\}$;
  for $n \leftarrow 1$ **to** N **do**
    $k \leftarrow \text{argmin}_j ||x^n - \mu_j||^2$;
    $r_k^n \leftarrow 1$;
    $r_{j \neq k}^n \leftarrow 0$;
  end
  for $k \leftarrow 1$ **to** K **do**
    $\mu_k \leftarrow \frac{\sum_{n=1}^N r_k^n x^n}{\sum_{n=1}^N r_k^n}$;
  end
until $\{r_k^n\} = \{^{\text{old}}r_k^n\}$;
return $\{\mu_k\}, \{r_k^n\}$;

K-means clustering

When updating the prototype vectors, we can also derive a sequential update in which, for each data point x^n in turn, we update the nearest prototype μ_k using:

$$^{\text{new}}\mu_k = {}^{\text{old}}\mu_k + \frac{1}{N_k}(x^n - {}^{\text{old}}\mu_k)$$

where N_k is the number of data points that have so far been used to update μ_k.

Image segmentation

Using the K-means algorithm to perform (toy) image segmentation:

- Each pixel in an image is a point in a three-dimensional space comprising the intensities of the red, blue and green channels.
- We treat each pixel in the image as a separate data point.
- We can apply the K-means algorithm to these pixels, and redraw the image in which we replace each pixel by the center μ_k to which that pixel has been assigned.

Image segmentation

Figure: Application of the K-means clustering algorithm to image segmentation

Image segmentation

Using the K-means algorithm to perform lossy data compression:

- For each of the N data points, we store only the identity k of the cluster to which it is assigned.
- We also store the values of the K cluster centers $\{\mu_k\}$.

This framework is often called vector quantization, and the vectors $\{\mu_k\}$ are called codebook vectors.

Mixtures of Gaussians

Formulation of Gaussian mixtures in terms of discrete latent variables:

- Let z be a K-dimensional binary random variable having a 1-of-K representation:
  - $p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$, where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.
- Let x be a random variable whose distribution given a particular value for z is a Gaussian:
  - $p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x; \mu_k, \Sigma_k)^{z_k}$.

Mixtures of Gaussians

We see that the marginal distribution for x is given by:

$$p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$$

which is a Gaussian mixture. We are now able to work with the joint distribution $p(x, z)$ instead of the marginal distribution $p(x)$, and this will lead to significant simplifications.

Mixtures of Gaussians

Let's calculate $\gamma(z_k) = p(z_k = 1|x)$:

$$p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{k'=1}^{K} p(z_{k'} = 1)p(x|z_{k'} = 1)} = \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(x; \mu_{k'}, \Sigma_{k'})}$$

We will view π_k as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed x. $\gamma(z_k)$ can also be viewed as the responsibility that component k takes for explaining the observation x.

Likelihood function

Suppose we have a data set of observations $\{x^1, \ldots, x^N\}$, and we wish to model this data using a mixture of Gaussians. The log of the likelihood function is given by:

$$L = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x^n; \mu_k, \Sigma_k))$$

Likelihood function

We see that:

- Due to the presence of the summation over k that appears inside the logarithm, when maximizing this log likelihood function, we will no longer obtain a closed-form solution.
- The maximization of the log likelihood function is not a well-posed problem, because singularities will occur whenever one of the Gaussian components collapses onto a specific data point.
- Identifiability issue: For any given (non-degenerate) point in the space of parameter values, there will be a further $K! - 1$ additional points all of which give rise to exactly the same distribution.

Maximum likelihood

Let's find the conditions that must be satisfied at a maximum of the log likelihood function:

$$0 = \frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N} \gamma(z_k^n)(x^n - \mu_k)^T \Sigma_k^{-1} \implies \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n) x^n$$

$$0 = \frac{\partial L}{\partial \Lambda_k}(H) = \frac{1}{2} \sum_{n=1}^{N} \gamma(z_k^n) \mathrm{tr}((\Sigma_k - (x^n - \mu_k)(x^n - \mu_k)^T)H)$$

$$\implies \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n)(x^n - \mu_k)(x^n - \mu_k)^T$$

$$\lambda = \frac{\partial L}{\partial \pi_k} = \frac{N_k}{\pi_k} \implies \pi_k = \frac{N_k}{N}$$

where $N_k = \sum_{n=1}^{N} \gamma(z_k^n)$. We can interpret N_k as the effective number of points assigned to cluster k.

Maximum likelihood

We can maximize the log likelihood function through an iterative procedure:

1. Choose some initial values for the means, covariances and mixing coefficients.
2. E step: Use the current values for the parameters to evaluate the posterior probabilities.
3. M step: Use these probabilities to re-estimate the means, covariances and mixing coefficients.
4. Go to step 2 until convergence.

Maximum likelihood

Algorithm 2: EM algorithm for a Gaussian mixture model

repeat

  for $n \leftarrow 1$ **to** N **do**

    for $k \leftarrow 1$ **to** K **do**

      $\gamma(z_k^n) = \frac{\pi_k \mathcal{N}(x^n; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(x^n; \mu_{k'}, \Sigma_{k'})}$;

    end

  end

  for $k \leftarrow 1$ **to** K **do**

    $N_k \leftarrow \sum_{n=1}^{N} \gamma(z_k^n)$;

    $\mu_k \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n) x^n$;

    $\Sigma_k \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n)(x^n - \mu_k)(x^n - \mu_k)^T$;

    $\pi_k \leftarrow \frac{N_k}{N}$;

  end

  $L \leftarrow \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x^n; \mu_k, \Sigma_k))$;

until *convergence*;

return $\{\mu_k\}, \{\Sigma_k\}, \{\pi_k\}$;

Expectation-maximization algorithm

Let's consider the EM algorithm under the more general situation:

- There are N observed data points: $x^1, \ldots, x^N \in \mathbb{R}^D$.
- The corresponding discrete latent variables $z^1, \ldots, z^N \in \mathbb{R}^K$ use a 1-of-K representation.
- The set of all model parameters is denoted by θ.

Expectation-maximization algorithm

The log likelihood function is given by:

$$L = \sum_{n=1}^{N} \log p(x^n; \theta) = \sum_{n=1}^{N} \log(\sum_{z^n} p(x^n, z^n; \theta))$$

The presence of the summation inside the logarithm results in complicated expressions for the maximum likelihood solution.

Expectation-maximization algorithm

The EM algorithm tries to maximize the log likelihood function through an iterative procedure:

1. Choose some starting value for the parameters θ_0.
2. E step: Calculate the posterior distribution of the latent variables $p(z^n|x^n;\theta^{\text{old}})$, so that we can form the expected value of the complete-data log likelihood under this posterior distribution $\mathcal{Q}(\theta,\theta^{\text{old}}) = \sum_{n=1}^{N}\sum_{z^n} p(z^n|x^n;\theta^{\text{old}}) \log p(x^n,z^n;\theta)$.
3. M step: We maximize this expectation and determine the revised parameter estimate $\theta^{\text{new}} = \text{argmax}_\theta \mathcal{Q}(\theta,\theta^{\text{old}})$.
4. Go to step 2 until convergence.

Expectation-maximization algorithm

Algorithm 3: General EM algorithm

repeat

$\mathcal{Q}(\theta, \theta^{\text{old}}) \leftarrow \sum_{n=1}^{N} \sum_{z^n} p(z^n | x^n; \theta^{\text{old}}) \log p(x^n, z^n; \theta)$;

$\theta^{\text{new}} \leftarrow \text{argmax}_\theta \mathcal{Q}(\theta, \theta^{\text{old}})$;

$L \leftarrow \sum_{n=1}^{N} \log p(x^n; \theta^{\text{new}})$;

$\theta^{\text{old}} \leftarrow \theta^{\text{new}}$;

until *convergence*;

return θ^{new};

Expectation-maximization algorithm

- The use of the expectation may seem somewhat arbitrary, we will see the motivation for this choice when we give a deeper treatment of EM in Section 15.4.
- In the definition of $\mathcal{Q}(\theta, \theta^{\text{old}})$, the logarithm acts directly on the joint distribution $p(x^n, z^n; \theta)$, and so the corresponding M step maximization will be tractable.
- The EM algorithm has the property that each cycle of EM will increase the incomplete-data log likelihood, as we will see in Section 15.4.

Gaussian mixtures

Application of this latent-variable view of EM to the specific case of a Gaussian mixture model. For the E step:

$$p(z^n = \mathrm{e}_k | x^n; \theta^{\mathrm{old}}) = p(z_k^n = 1 | x^n; \theta^{\mathrm{old}})$$

$$= \frac{p(z_k^n = 1, x^n; \theta^{\mathrm{old}})}{\sum_{k'=1}^K p(z_{k'}^n = 1, x^n; \theta^{\mathrm{old}})} = \frac{\pi_k^{\mathrm{old}} \mathcal{N}(x^n; \mu_k^{\mathrm{old}}, \Sigma_k^{\mathrm{old}})}{\sum_{k'=1}^K \pi_{k'}^{\mathrm{old}} \mathcal{N}(x^n; \mu_{k'}^{\mathrm{old}}, \Sigma_{k'}^{\mathrm{old}})} = \gamma(z_k^n)$$

$$\mathcal{Q}(\theta, \theta^{\mathrm{old}}) = \sum_{n=1}^N \sum_{z^n} p(z^n | x^n; \theta^{\mathrm{old}}) \log p(x^n, z^n; \theta)$$

$$= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_k^n)(\log \pi_k + \log \mathcal{N}(x^n; \mu_k, \Sigma_k))$$

Gaussian mixtures

For the M step, we fix θ^{old} thus $\gamma(z_k^n)$, and maximize $\mathcal{Q}(\theta, \theta^{\text{old}})$ with respect to θ:

$$N_k = \sum_{n=1}^{N} \gamma(z_k^n)$$

$$0 = \frac{\partial \mathcal{Q}}{\partial \mu_k} = \sum_{n=1}^{N} \gamma(z_k^n)(x^n - \mu_k)^T \Sigma_k^{-1} \implies \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n) x^n$$

$$0 = \frac{\partial \mathcal{Q}}{\partial \Lambda_k}(H) = \frac{1}{2} \sum_{n=1}^{N} \gamma(z_k^n) \text{tr}((\Sigma_k - (x^n - \mu_k)(x^n - \mu_k)^T) H)$$

$$\implies \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n)(x^n - \mu_k)(x^n - \mu_k)^T$$

$$\lambda = \frac{\partial \mathcal{Q}}{\partial \pi_k} = \frac{N_k}{\pi_k} \implies \pi_k = \frac{N_k}{N}$$

Relation to K-means

We can derive the K-means algorithm as a particular limit of EM for Gaussian mixtures. Consider a Gaussian mixture model in which:

- The mixing coefficients are fixed to $\frac{1}{K}$.
- The covariance matrices of the mixture components are given by ϵI, where ϵ is a fixed constant.

Relation to K-means

Consider the limit $\epsilon \to 0+$. For the E step:

$$\gamma(z_k^n) = \frac{\exp(-\frac{||x^n - \mu_k^{\text{old}}||^2}{2\epsilon})}{\sum_{k'=1}^{K} \exp(-\frac{||x^n - \mu_{k'}^{\text{old}}||^2}{2\epsilon})}$$

$$\to \begin{cases} 1, & \text{if } k = \text{argmin}_j ||x^n - \mu_j^{\text{old}}||^2 \\ 0, & \text{otherwise} \end{cases}$$

Thus, in this limit, we obtain a hard assignment of data points to clusters, just as in the K-means algorithm, so that $\gamma(z_k^n) \to r_k^n$.

Relation to K-means

The expected complete-data log likelihood becomes:

$$\epsilon\mathcal{Q}(\theta, \theta^{\text{old}}) = \epsilon \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_k^n)(-\log K + \log \mathcal{N}(x^n; \mu_k, \epsilon I))$$

$$\rightarrow -\frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^n ||x^n - \mu_k||^2$$

We see that in this limit, maximizing the expected complete-data log likelihood is equivalent to minimizing the error measure J for the K-means algorithm.

Relation to K-means

For the M step:

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_k^n) x^n}{\sum_{n=1}^{N} \gamma(z_k^n)} \rightarrow \frac{\sum_{n=1}^{N} r_k^n x^n}{\sum_{n=1}^{N} r_k^n}$$

The EM re-estimation equation for the μ_k then reduces to the K-means result.

Mixtures of Bernoulli distributions

As a further example of mixture modelling and to illustrate the EM algorithm in a different context, we now discuss mixtures of descrete binary variables described by Bernoulli distributions:

- Let z be a K-dimensional binary random variable having a 1-of-K respresentation:
  - $p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$, where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.
- Let $x \in \mathbb{R}^D$ be a set of D binary variables, each of which is governed by a Bernoulli distribution given a particular value for z:
  - $\text{Bern}(x; \mu) = \prod_{d=1}^{D} \mu_d^{x_d} (1 - \mu_d)^{1-x_d}$.
  - $p(x|z) = \prod_{k=1}^{K} \text{Bern}(x; \mu^k)^{z_k}$.

Mixtures of Bernoulli distributions

Application of the EM algorithm to the specific case of a Bernoulli mixture model. For the E step:

$$\gamma(z_k^n) = p(z^n = \mathrm{e}_k | x^n; \theta^{\text{old}}) = p(z_k^n = 1 | x^n; \theta^{\text{old}})$$

$$= \frac{p(z_k^n = 1, x^n; \theta^{\text{old}})}{\sum_{k'=1}^K p(z_{k'}^n = 1, x^n; \theta^{\text{old}})} = \frac{\pi_k^{\text{old}} \text{Bern}(x^n; (\mu^k)^{\text{old}})}{\sum_{k'=1}^K \pi_{k'}^{\text{old}} \text{Bern}(x^n; (\mu^{k'})^{\text{old}})}$$

$$\mathcal{Q}(\theta, \theta^{\text{old}}) = \sum_{n=1}^N \sum_{z^n} p(z^n | x^n; \theta^{\text{old}}) \log p(x^n, z^n; \theta)$$

$$= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_k^n)(\log \pi_k + \log \text{Bern}(x^n; \mu^k))$$

Mixtures of Bernoulli distributions

For the M step, we fix θ^{old} thus $\gamma(z_k^n)$, and maximize $\mathcal{Q}(\theta, \theta^{\text{old}})$ with respect to θ:

$$N_k = \sum_{n=1}^{N} \gamma(z_k^n)$$

$$0 = \frac{\partial \mathcal{Q}}{\partial \mu_d^k} = \frac{1}{\mu_d^k(1-\mu_d^k)} \sum_{n=1}^{N} \gamma(z_k^n)(x_d^n - \mu_d^k)$$

$$\implies \mu_d^k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n) x_d^n \implies \mu^k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^n) x^n$$

$$\lambda = \frac{\partial \mathcal{Q}}{\partial \pi_k} = \frac{N_k}{\pi_k} \implies \pi_k = \frac{N_k}{N}$$

Evidence lower bound

We now present an even more general perspective on the EM algorithm by deriving a lower bound on the log likelihood function, which is known as the evidence lower bound or ELBO. Again, the assumptions are:

- There is a probabilistic model in which we denote all the observed variables by X and all the hidden variables by Z. The model is governed by a set of parameters denoted by θ.
- Direct optimization of $p(X;\theta)$ is difficult.
- Optimization of the complete-data likelihood function $p(X,Z;\theta)$ is significantly easier.

Evidence lower bound

For any distribution $q(Z)$ defined over the latent variables, we have:

$$
\begin{aligned}
\log p(X;\theta) &= \sum_Z q(Z) \log p(X;\theta) \\
&= \sum_Z q(Z) \log \frac{p(X,Z;\theta)}{p(Z|X;\theta)} \\
&= \sum_Z q(Z) \log \frac{\frac{p(X,Z;\theta)}{q(Z)}}{\frac{p(Z|X;\theta)}{q(Z)}} \\
&= \sum_Z q(Z) \log \frac{p(X,Z;\theta)}{q(Z)} - \sum_Z q(Z) \log \frac{p(Z|X;\theta)}{q(Z)} \\
&= \mathcal{L}(q,\theta) + \mathrm{KL}(q(Z)||p(Z|X;\theta))
\end{aligned}
$$

Evidence lower bound

We see that $\mathrm{KL}(q(Z)||p(Z|X;\theta))$ is the Kullback-Leibler divergence between $q(Z)$ and the posterior distribution $p(Z|X;\theta)$, thus $\mathrm{KL}(q(Z)||p(Z|X;\theta)) \geq 0$, with equality if and only if $q(Z) = p(Z|X;\theta)$. It also follows that:

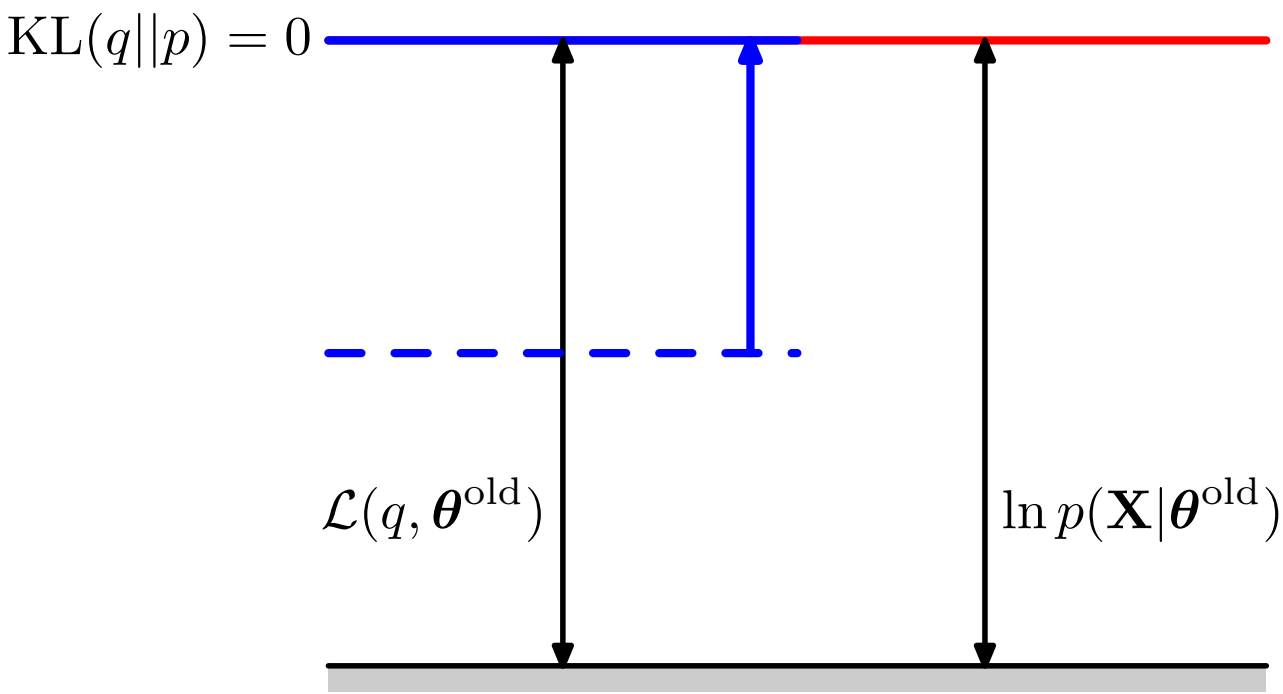
$$\log p(X;\theta) \geq \mathcal{L}(q,\theta)$$

In other words that $\mathcal{L}(q,\theta)$ is a lower bound on $\log p(X;\theta)$.

EM revisited

In the E step, the lower bound $\mathcal{L}(q, \theta^{\text{old}})$ is maximized with respect to $q(Z)$ while holding θ^{old} fixed. It's easy to see that the solution to this maximization problem is $q(Z) = p(Z|X;\theta)$.

Figure: Illustration of the E step of the EM algorithm



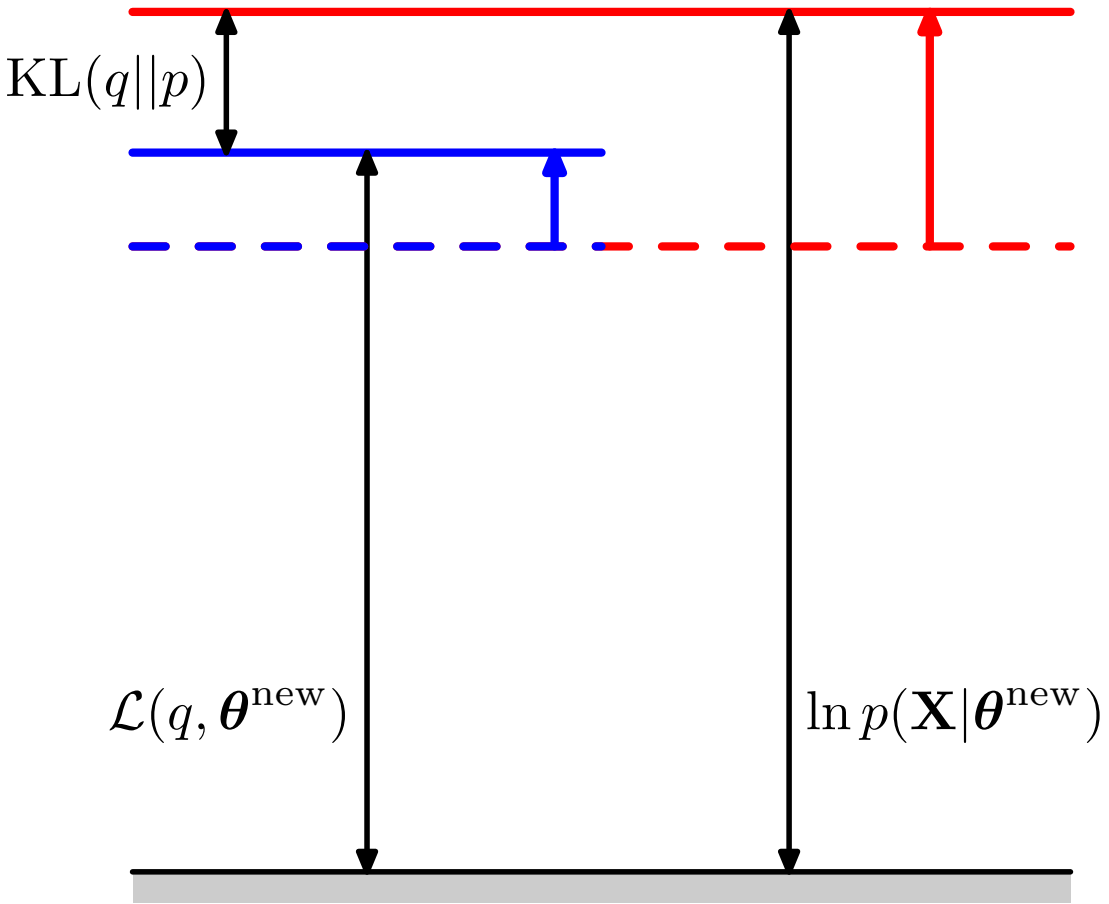
EM revisited

In the M step, the distribution $q(Z)$ is held fixed and the lower bound $\mathcal{L}(q, \theta)$ is maximized with respect to θ to give some new value θ^{new}:

- This will cause the lower bound $\mathcal{L}$ to increase, which will necessarily cause the corresponding log likelihood function to increase.
- Because $q(Z) = p(Z|X; \theta^{\text{old}})$, it will not equal the new posterior distribution $p(Z|X; \theta^{\text{new}})$, and hence there will be a non-zero Kullback-Leibler divergence.

EM revisited

Figure: Illustration of the M step of the EM algorithm



EM revisited

Let's take a closer look at the function we try to maximize in the M step:

$$\begin{aligned}\mathcal{L}(q,\theta) &= \sum_Z p(Z|X;\theta^{\text{old}}) \log \frac{p(X,Z;\theta)}{p(Z|X;\theta^{\text{old}})} \\ &= \sum_Z p(Z|X;\theta^{\text{old}}) \log p(X,Z;\theta) - \sum_Z p(Z|X;\theta^{\text{old}}) \log p(Z|X;\theta^{\text{old}}) \\ &= \mathcal{Q}(\theta,\theta^{\text{old}}) + \text{const}\end{aligned}$$

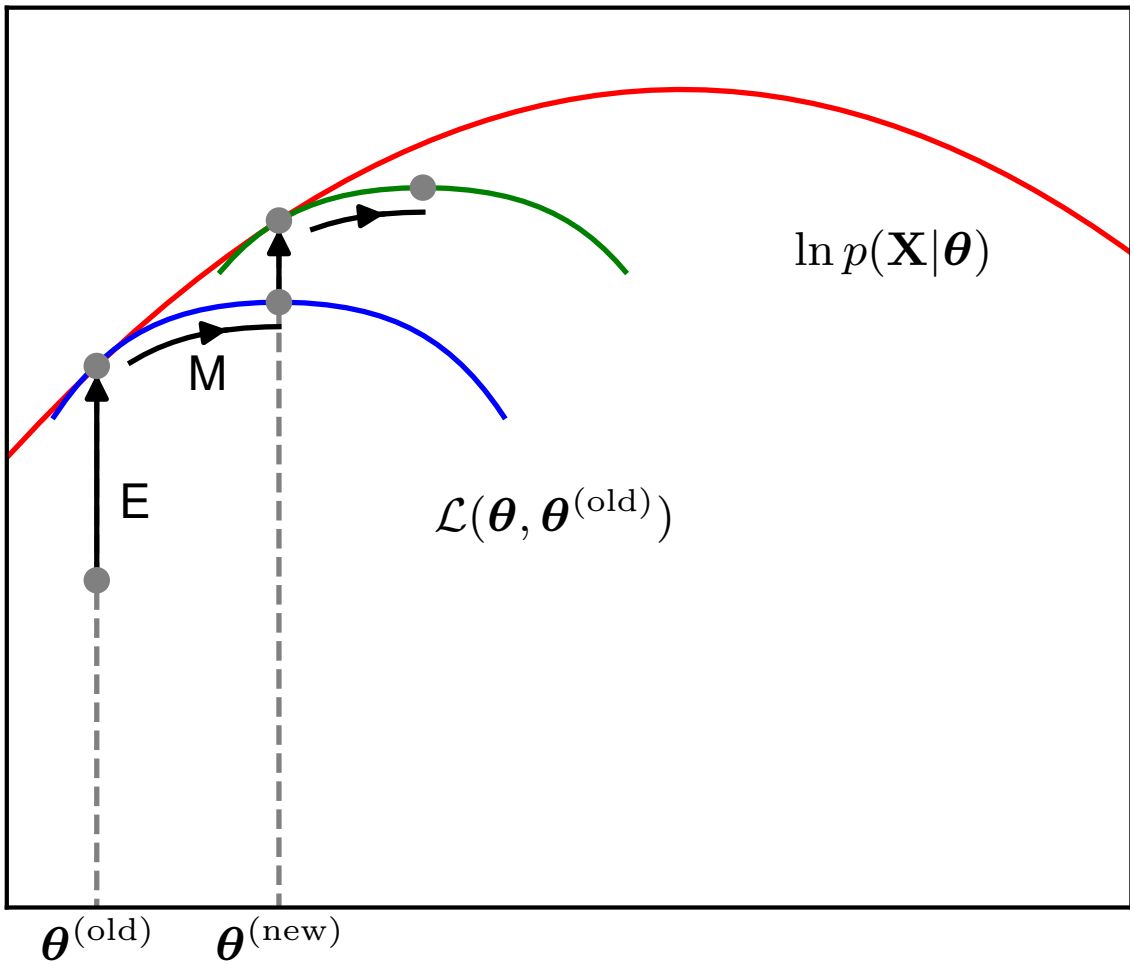
EM revisited

We see that:

- Here we recognize $\mathcal{Q}(\theta, \theta^{\text{old}})$ as the expected complete-data log likelihood, and it is therefore the quantity that is being maximized in the M step.
- The variable θ over which we are optimizing appears only inside the logarithm, and by our assumption, the complete-data likelihood function $p(X, Z; \theta)$ is significantly easier to optimize.

EM revisited

Figure: The operation of the EM algorithm viewed in the space of parameters



EM revisited

After each E step, the lower bound curve $\mathcal{L}(q, \theta)$ becomes tangent to the log likelihood curve $\log p(X; \theta)$ at θ^{old}:

$$\begin{aligned}
\left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} \mathcal{L}(q, \theta) &= \sum_Z \left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} p(Z|X; \theta^{\text{old}}) \log \frac{p(X, Z; \theta)}{p(Z|X; \theta^{\text{old}})} \\
&= \frac{1}{p(X; \theta^{\text{old}})} \sum_Z \left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} p(X, Z; \theta) \\
&= \frac{1}{p(X; \theta^{\text{old}})} \left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} \sum_Z p(X, Z; \theta) \\
&= \frac{1}{p(X; \theta^{\text{old}})} \left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} p(X; \theta) \\
&= \left.\frac{\partial}{\partial \theta}\right|_{\theta=\theta^{\text{old}}} \log p(X; \theta)
\end{aligned}$$

Independent and identically distributed data

For the particular case of an i.i.d. data set, we have $p(X, Z; \theta) = \prod_{n=1}^{N} p(x^n, z^n; \theta)$. Further, we have:

$$
\begin{aligned}
p(X; \theta) &= \sum_{Z} p(X, Z; \theta) \\
&= \sum_{z^1} \cdots \sum_{z^N} p(x^1, z^1; \theta) \cdots p(x^N, z^N; \theta) \\
&= \sum_{z^1} p(x^1, z^1; \theta) \cdots \sum_{z^N} p(x^N, z^N; \theta) \\
&= p(x^1; \theta) \cdots p(x^N; \theta)
\end{aligned}
$$

$$
p(Z|X; \theta) = \frac{p(X, Z; \theta)}{p(X; \theta)} = \frac{\prod_{n=1}^{N} p(x^n, z^n; \theta)}{\prod_{n=1}^{N} p(x^n; \theta)} = \prod_{n=1}^{N} p(z^n|x^n; \theta)
$$

Independent and identically distributed data

$$\mathcal{Q}(\theta, \theta^{\text{old}}) = \sum_{Z} p(Z|X;\theta^{\text{old}}) \log p(X, Z;\theta)$$

$$= \sum_{z^1} \cdots \sum_{z^N} p(z^1|x^1;\theta^{\text{old}}) \cdots p(z^N|x^N;\theta^{\text{old}}) \sum_{n=1}^{N} \log p(x^n, z^n;\theta)$$

$$= \sum_{n=1}^{N} \sum_{z^1} \cdots \sum_{z^N} p(z^1|x^1;\theta^{\text{old}}) \cdots p(z^N|x^N;\theta^{\text{old}}) \log p(x^n, z^n;\theta)$$

$$= \sum_{n=1}^{N} \sum_{z^n} p(z^n|x^n;\theta^{\text{old}}) \log p(x^n, z^n;\theta)$$

This is exactly the form that was given when we formally introduced the EM algorithm in Section 15.3. We see that the form given here is more general.

Parameter priors

We can also use the EM algorithm to maximize the postierior distribution $p(\theta|X)$ for models in which we have introduced a prior $p(\theta)$ over the parameters:

$$\begin{aligned}\log p(\theta|X) &= \log p(X,\theta) - \log p(X) \\ &= \log p(X|\theta) + \log p(\theta) - \log p(X) \\ &= \mathcal{L}(q,\theta) + \mathrm{KL}(q(Z)||p(Z|X,\theta)) + \log p(\theta) - \log p(X) \\ &\geq \mathcal{L}(q,\theta) + \log p(\theta) - \log p(X)\end{aligned}$$

The E step is the same as the standard EM algorithm, while the M-step equations are modified through the introduction of the prior term $\log p(\theta)$.

Generalized EM

For complex models it may be the case that either the E step or the M step, or indeed both, remain intractable:

- The generalized EM algorithm addresses the problem of an intractable M step:
  - One way would be to use gradient-based iterative optimization algorithm during the M step.
  - The expectation conditional maximization algorithm involves making several constrained optimizations within each M step.
- We can similarly generalize the E step of the EM algorithm by performing a partial, rather than complete, optimization of $\mathcal{L}(q, \theta)$ with respect to $q(Z)$.

Sequential EM

Incremental form of EM:

- In the E step, we just re-evaluate the responsibilities for one data point.
- In the M step, update the required sufficient statistics incrementally.

Sequential EM

For intance, for a Gaussian mixture, suppose we perform an update for data point m:

$$\gamma^{\text{new}}(z_k^m) = \cdots$$

$$N_k^{\text{new}} = N_k^{\text{old}} - \gamma^{\text{old}}(z_k^m) + \gamma^{\text{new}}(z_k^m)$$

$$\mu_k^{\text{new}} = \mu_k^{\text{old}} + \frac{\gamma^{\text{new}}(z_k^m) - \gamma^{\text{old}}(z_k^m)}{N_k^{\text{new}}}(x^m - \mu_k^{\text{old}})$$

The corresponding results for the covariances and the mixing coefficients are analogous.