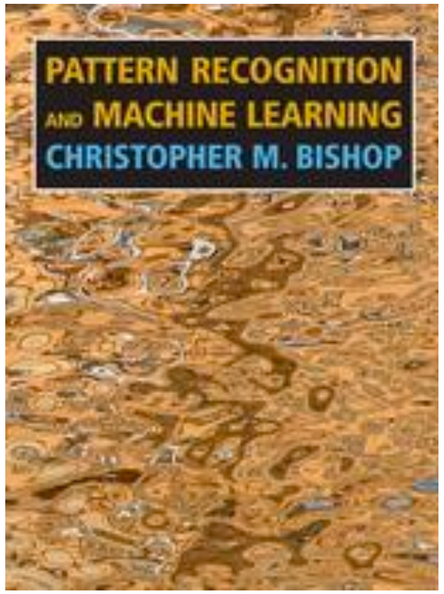
EM algorithm

Yuanning Li

BME 2111
Neural Signal Processing and Data Analysis
2023 Fall

Topics we will cover in PRML



Chap. 4: Classification. Naive Bayes.
Neuroscience application: discrete neural decoding

Chap. 8: Graphical models.

Chap. 9: Mixture models. Expectation-maximization.
Neuroscience application: spike sorting

Chap. 12: Principal components analysis. Factor analysis.
Neuroscience applications: spike sorting, dimensionality reduction

Chap. 13: Kalman filter.
Neuroscience application: continuous neural decoding

Mixture of Gaussian

- Let $z \in \{1,...,K\}$ be a random variable $P(z = k) = \pi_k$, where

$$k = 1,...,K \, (0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1)$$

$$P(\mathbf{x} \mid z = k) = \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$$

- Thus,

$$P(\mathbf{x}) = \sum_{z} P(\mathbf{x} \mid z) P(z) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k) \pi_k$$

- What is the graphical model for $P(\mathbf{x}, z)$?

Maximum likelihood parameter estimation

- Goal: Fit μ_k, Σ_k, π_k to training data $\{x_1, \ldots, x_N\}$

$$P(\{x\} \mid \theta) = \prod_{n=1}^{N} P(x_n)$$

$$= \prod_{n=1}^{N} \sum_{k=1}^{K} \mathcal{N}(x_n \mid \mu_k, \Sigma_k)\pi_k$$

$$\mathcal{L}(x, \theta) = \log P(\{x\} \mid \theta) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \mathcal{N}(x_n \mid \mu_k, \Sigma_k)\pi_k \right\}$$

Maximum likelihood parameter estimation

- Find μ_k

$$\frac{\partial \mathscr{L}}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{j=1}^{K} \mathcal{N}(x_n \mid \mu_j, \Sigma_j)\pi_j} \pi_k \left(\frac{\partial}{\partial \mu_k} \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right) \quad (1)$$

Note that

$$\begin{aligned} \frac{\partial}{\partial \mu_k} \mathcal{N}(x_n \mid \mu_k, \Sigma_k) &= \frac{\partial}{\partial \mu_k} \left(\frac{1}{(2\pi)^{D/2} \mid \Sigma_k \mid^{1/2}} \exp\{-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\} \right) \\ &= \mathcal{N}(x_n \mid \mu_k, \Sigma_k)\Sigma_k^{-1}(x_n - \mu_k) \end{aligned}$$

Plugging into (1),

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial \mu_k} &= \sum_{n=1}^{N} \frac{\mathcal{N}(x_n \mid \mu_k, \Sigma_K)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(x_n \mid \mu_j, \Sigma_j)\pi_j} \Sigma_k^{-1}(x_n - \mu_k) \\ &= \Sigma_k^{-1} \Sigma_{n=1}^{N} \gamma_{nk}(x_n - \mu_k) \\ &= 0 \end{aligned}$$

$$\boxed{\gamma_{nk} \equiv \frac{\mathcal{N}(x_n \mid \mu_k, \Sigma_K)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(x_n \mid \mu_j, \Sigma_j)\pi_j}}$$

Maximum likelihood parameter estimation

Letting $N_k = \sum_{n=1}^{N} \gamma_{nk}$,

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n \qquad (2)$$

Let's try to understand this result intuitively

$$\begin{aligned} \gamma_{nk} &= P(z_n = k \mid x_n) \\ &= \frac{P(x_n \mid z_n = k) P(z_n = k)}{P(x_n)} \\ &= \frac{\mathcal{N}(x_n \mid \mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(x_n \mid \mu_j, \Sigma_j)\pi_j} \end{aligned}$$

γ_{nk} is the "responsibility" that cluster k takes for explaining the observation x_n,

where as π_k is the prior probability that $z_n = k$, γ_{nk} is the posterior probability that $z_n = k$ once we have observed x_n

Thus, in (2), μ_k is obtained by taking a weighted mean, where the weights are given by the responsibilities ($0 \leq \gamma_{nk} \leq 1$)

This is reminiscent of K-means, but weights can now range from 0 to 1 (soft assignments)

Maximum likelihood parameter estimation

- Find Σ_k

Setting $\frac{\partial \mathscr{L}}{\partial \Sigma_k} = 0$,

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k)(x_n - \mu_k)^T \quad (3)$$

(3) has an intuitive interpretation like in (2)

Maximum likelihood parameter estimation

- Find π_k

Here we need to perform a constrained optimization since $\sum_{k=1}^{K} \pi_k = 1$

Instead of maximizing $\mathscr{L}$, we will maximize

$$\mathscr{L}' = \mathscr{L} + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$$

$$\frac{\partial \mathscr{L}'}{\partial \pi_k} = \frac{\partial \mathscr{L}}{\partial \pi_k} + \lambda$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{j=1}^{N} \mathcal{N}(x_n \mid \mu_j, \Sigma_j)\pi_j} \mathcal{N}(x_n \mid \mu_k, \Sigma_k) + \lambda$$

$$= 0$$

Maximum likelihood parameter estimation

Multiplying both sides by π_k,

$$\sum_{n=1}^{N} \gamma_{nk} + \lambda \pi_k = 0, \quad \Rightarrow \quad \boxed{\pi_k = -\frac{N_k}{\lambda}} \quad (4)$$

Enforcing the constraint $\sum_{k=1}^{K} \pi_k = 1,$

$$-\frac{\sum_{k=1}^{K} N_k}{\lambda} = 1, \quad \Rightarrow \quad \boxed{\pi_k = \frac{N_k}{N}} \quad (5)$$

This is the effective number of data points assigned to class k, divided by the total number of data points.

Maximum likelihood parameter estimation

Very important note:

(2), (3) and (5) do not constitute a closed-form solution because the responsibilities γ_{nk} (which appear in (2), (3) and (5)) depend on μ_k, Σ_k and π_k.

This suggests an iterative scheme, which turns out to be an instance of the Expectation-Maximization (EM) algorithm.

The EM algorithm is a powerful and general method for finding the maximum likelihood parameters for models with latent variables.

In the mixture of Gaussians, z is a latent variable because it is not observed (only x is observed).

EM algorithm for Mixture of Gaussians

1) Initialize μ_k, Σ_k, π_k

2) E-step: Evaluate responsibilities given current parameter values

$$\gamma_{nk} = \frac{\mathcal{N}(x_n \,|\, \mu_k, \Sigma_K)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(x_n \,|\, \mu_j, \Sigma_j)\pi_j}$$

3) M-step: Re-estimate parameters using current responsibilities

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k^{\text{new}})(x_n - \mu_k^{\text{new}})^T$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

Where $N_k = \sum_{n=1}^{N} \gamma_{nk}$

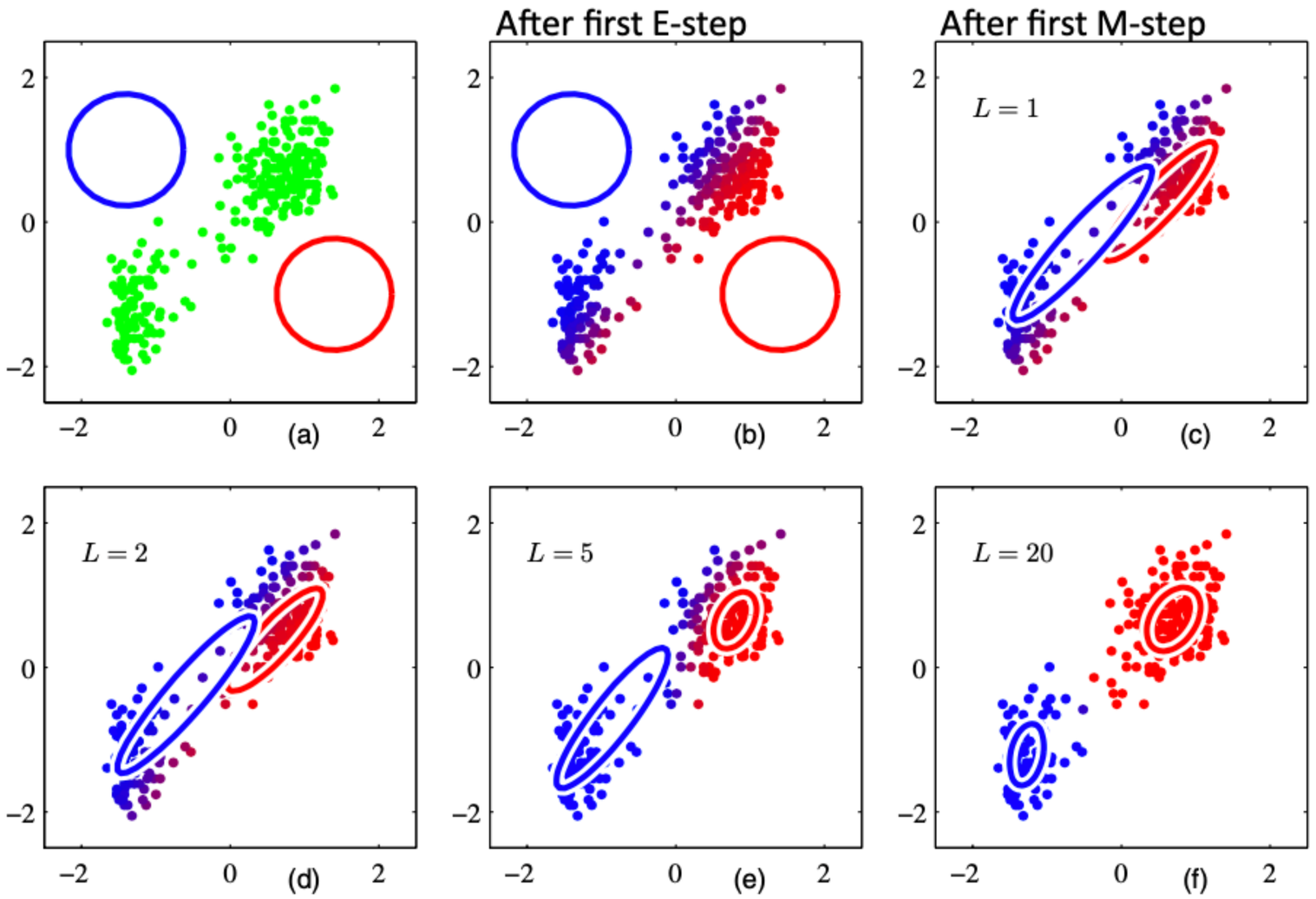
4) Evaluate log likelihood of data

$$\log P(\{x\} \,|\, \theta) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \mathcal{N}(x_n \,|\, \mu_k, \Sigma_k)\pi_k \right\}$$

And check for convergence of either the parameters or the log likelihood. If convergence criterion not satisfied, return to step 2).

We will soon show that each iteration of the EM algorithm is guaranteed to increase the log likelihood $\log P(\{x\} \,|\, \theta)$

Example of EM for mixture of Gaussians



Relating EM for Gaussian Mixtures to Classification

- During the training phase, the goal in both cases is to estimate the model parameters from the training data.
- Key differences:
  - In Classification, the class labels are known;
  - In Gaussian mixtures, the class labels are not known.

Relating EM for Gaussian Mixtures to Classification

- In Classification, we were able to estimate the model parameters in closed form:
  - $\pi_k = \frac{N_k}{N}$, where $N_k = \sum_{n \in C_k} \mathbf{1}$ (1)
  - $\mu_k = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{x}_n$ (2)
  - $\Sigma_k = \frac{1}{N_k} \sum_{n \in C_k} \left(\mathbf{x}_n - \mu_k\right) \left(\mathbf{x}_n - \mu_k\right)^T$ (3)

Relating EM for Gaussian Mixtures to Classification

- In Gaussian mixtures, it is unknown whether $n \in C_k$ (i.e. we don't have class labels), which leads to a "chicken and egg" problem.
  - If the class labels were known, it would be easy to compute model parameters using (1) - (3)
  - If the model parameters were known, it would be easy to compute the posterior likelihood on class labels $P(C_k \mid \mathbf{x}_n)$, similar to classification

Relating EM for Gaussian Mixtures to Classification

- Using the EM algorithm, we can “bootstrap” our way out of the “chicken and egg” problem.
  - Initialize model parameters
  - Repeat until convergence:
    - E-Step: Estimate class labels given current model parameters
    - M-Step: Estimate model parameters given current class labels (represented as a distribution over class labels)

Relating EM for Gaussian Mixtures to Classification

- In Gaussian Mixtures, the training phase uses the EM algorithm, in which
  - The E-step is reminiscent of test phase in classification
  - The M-step is reminiscent of training phase in classification
- In Gaussian Mixtures, the test phase typically involves running a single E-step on the test data (no iteration needed).

The EM algorithm in general

- Motivation:
  - You have a model that you would like to fit to your training data.
  - If the model has latent variables (very likely in neural signal processing), the EM algorithm provides a recipe for fitting the model.
- Because of its generality, the EM algorithm is among the most important and widely used tools in machine learning.
- We will be using it repeatedly in the rest of this course.

Decomposition of data likelihood

- Let X denote all observed variables

  Z denote all latent variables

  θ denote all model parameters

- Goal: Maximize $p(X|\theta)$ w.r.t. θ
- Let $q(Z)$ be some distribution over latent variables.
- For any choice of $q(Z)$, we can decompose $\log p(X|\theta)$ into a sum of two terms:

$$\log p(X|\theta) = \mathcal{L}(q,\theta) + KL(q||p) \qquad (4)$$

Where

$$\mathcal{L}(q,\theta) = \sum_Z q(Z)\log\frac{p(X,Z|\theta)}{q(Z)} \qquad (5)$$

$$KL(q||p) = -\sum_Z q(Z)\log\frac{p(Z|X,\theta)}{q(Z)} \qquad (6)$$

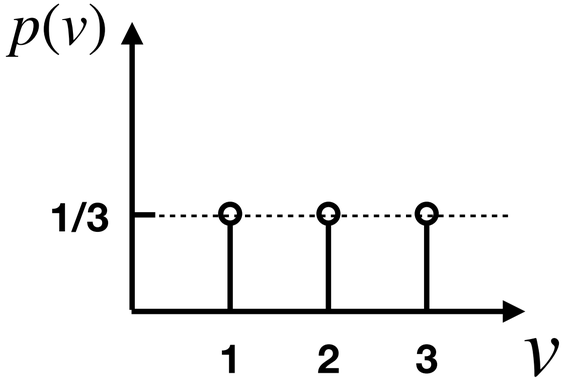
Decomposition of data likelihood

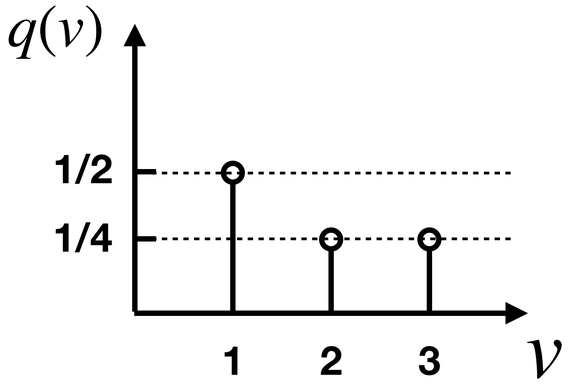
- Notes:
- $\mathcal{L}(q, \theta)$ is a scalar that depends only on q and θ, because X is observed and Z is summed out.
- $KL(q||p)$ is the Kullback-Leibler divergence between $q(Z)$ and $p(Z|X, \theta)$.
  - The KL divergence is a scalar that measures the distance between probability distributions. In general, KL divergence is written as:

$$KL(q||p) = -\sum_{v} q(v) \log \frac{p(v)}{q(v)}$$

KL divergence example

- Let's do a simple example to illustrate KL divergence





Facts:

$KL(q\|p) \geq 0$ **for any** p **and** q

$KL(q\|p) = 0$**, if and only if** $p = q$

- $KL(q\|p) = -\left(q(1)\log\frac{p(1)}{q(1)} + q(2)\log\frac{p(2)}{q(2)} + q(2)\log\frac{p(3)}{q(3)}\right) = 0.0589$
- $KL(p\|q) = -\left(p(1)\log\frac{q(1)}{p(1)} + p(2)\log\frac{q(2)}{p(2)} + p(2)\log\frac{q(3)}{p(3)}\right) = 0.0566$
- $KL(p\|q) = KL(q\|p) = -\sum_{v} p(v)\log\frac{p(v)}{p(v)} = 0$

Decomposition of data likelihood

$$\log p(X|\theta) = \mathcal{L}(q,\theta) + KL(q\|p)$$

- In (6), $KL(q\|p) \geq 0$, with equality if and only if $q(Z) = p(Z|X,\theta)$
- Thus, $\log p(X|\theta) \geq \mathcal{L}(q,\theta)$

  $\Rightarrow$ $\mathcal{L}(q,\theta)$ is a lower bound for $\log p(x|\theta)$
- The lower bound is tight (i.e. $\log p(X|\theta) = \mathcal{L}(q,\theta)$) when $q(Z)$ is chosen to be $p(Z|X,\theta)$

Decomposition of data likelihood

- Let's verify the decomposition.

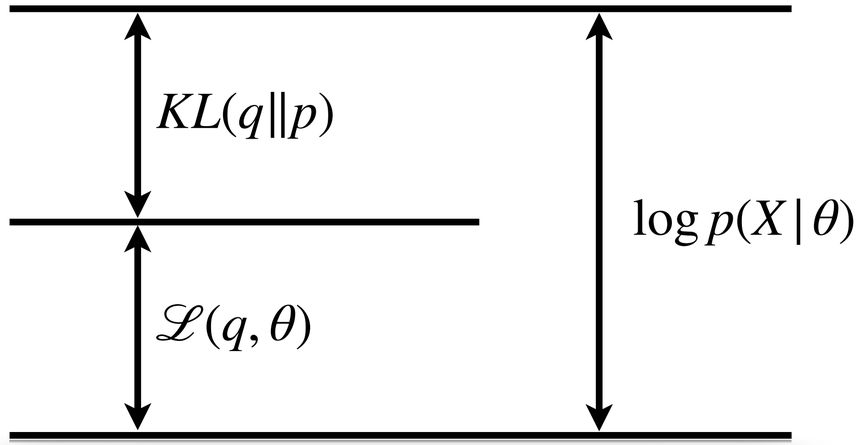
$$\mathcal{L}(q,\theta) = \sum_Z q(Z)\log\frac{p(Z|X,\theta)p(X|\theta)}{q(Z)}$$

$$= \sum_Z q(Z)\log\frac{p(Z|X,\theta)}{q(Z)} + \sum_Z q(Z)\log p(X|\theta)$$

$$= -KL(q\|p) + \log p(X|\theta)$$

$$\Rightarrow \quad \log p(X|\theta) = \mathcal{L}(q,\theta) + KL(q\|p)$$

Picture to have in mind:



The EM algorithm

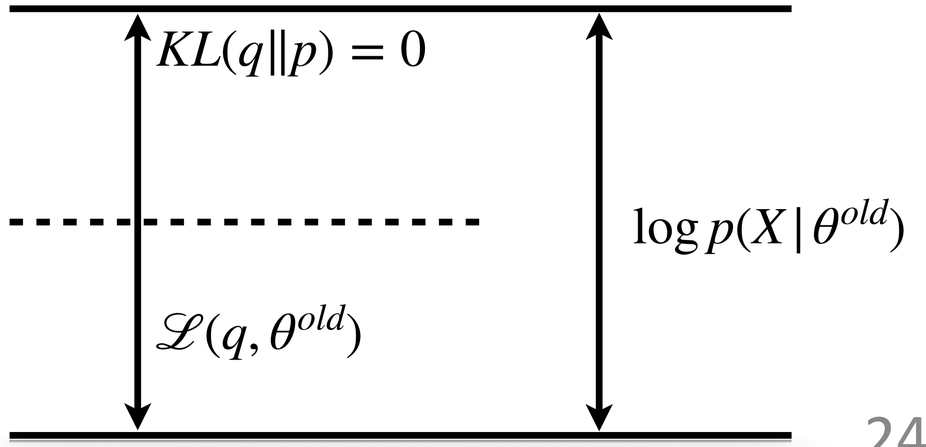
- Goal: maximize $\log p(X|\theta)$ w.r.t. θ
- Approach: Iteratively maximize the lower bound $\mathscr{L}(q,\theta)$ w.r.t. q and θ

  E-step: maximize $\mathscr{L}(q,\theta)$ w.r.t. q while keeping θ fixed.

  M-step: maximize $\mathscr{L}(q,\theta)$ w.r.t. θ while keeping q fixed.

E-step:

- With fixed θ, $\mathscr{L}(q,\theta)$ is maximized when $KL(q\|p) = 0$, corresponding to $q(Z) = p(Z|X,\theta)$



The EM algorithm

M-step:

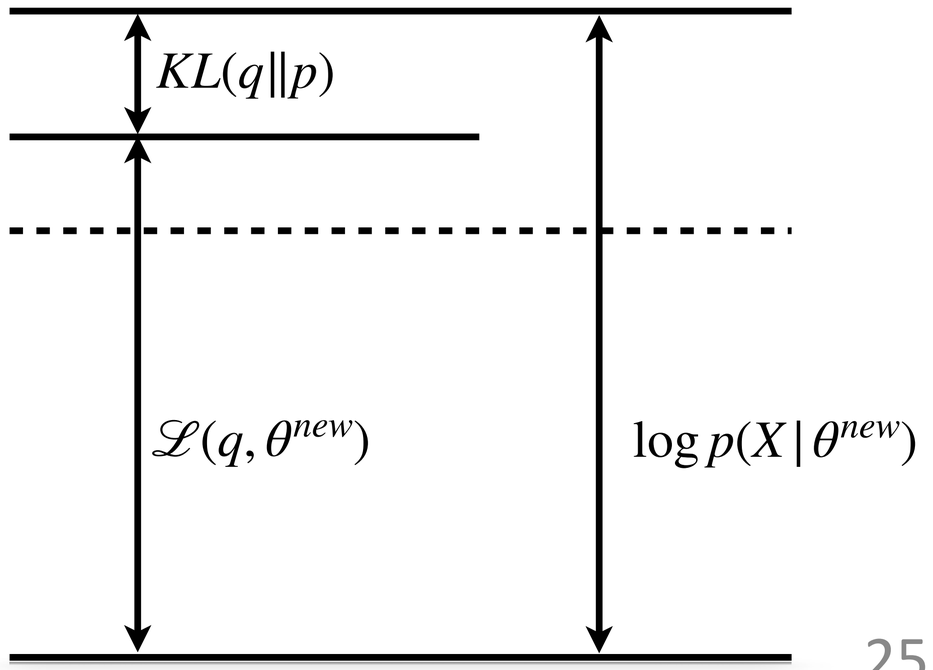
- $\mathscr{L}(q,\theta) = \sum_Z q(Z)\log p(X,Z\,|\,\theta) - \sum_Z q(Z)\log q(Z)$

No θ dependence

This is $\mathbb{E}_q\left[\log p(X,Z\,|\,\theta)\right]$, the expected log joint distribution

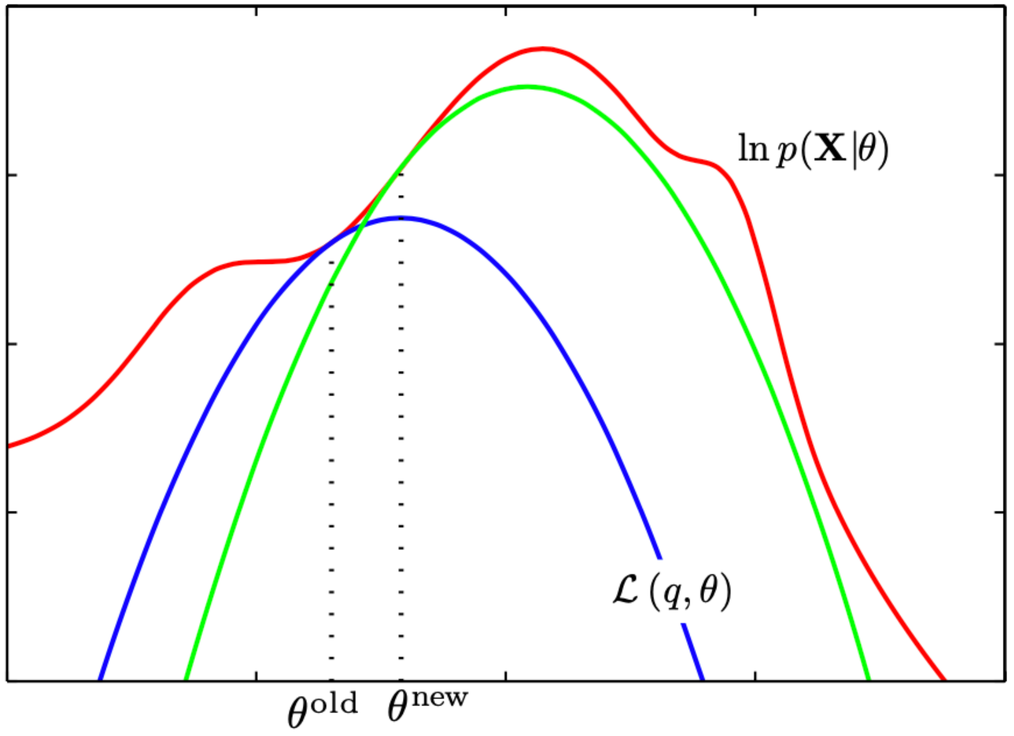
- With fixed q, maximizing $\mathscr{L}(q,\theta)$ is equivalent maximizing $\sum_Z q(Z)\log p(X,Z\,|\,\theta)$

- Because $\mathscr{L}(q,\theta)$ is lower bound on $\log p(X\,|\,\theta)$, $\log p(X\,|\,\theta)$ must increase at least as much as $\mathscr{L}(q,\theta)$ does.
- Now, there's a nonzero $KL(q\|p)$.
- Based on the pictures on the previous pages, we see that $\log p(X\,|\,\theta)$ is guaranteed to be non-decreasing from one EM iteration to the next.
- Thus the EM algorithm is guaranteed to converge to a local optimum.



The EM algorithm in parameter space

- The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values.
- E-step: make lower bound tight
- M-step: find peak of lower bound



Summary of the general EM algorithm

1) Initialize parameters θ^{old}, compute $\log p(X|\theta^{old})$
2) E-step:

   Evaluate $p(Z|X,\theta^{old})$
3) M-step:

   Find $\theta^{new} = \underset{\theta}{\text{argmax}} \left(\sum_{Z} p(Z|X,\theta^{old}) \log p(X,Z|\theta) \right)$
4) Compute $\log p(X|\theta^{new})$, compare to $\log p(X|\theta^{old})$
5) If not converged, assign

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

   and goto 2)