Mixture models and the EM algorithm

Jiali Lin

Virginia Tech

January 15, 2017

Outline

Latent variable models

K-means Clustering

Mixture models

The EM algorithm

Latent Variable Models (LVMs)

- **Graphial model:** model dependence between two variables by adding an edge between them.
- **Latent variable:** assume that the observed variables are correlated because they arise from a hidden common "cause".
- **Pros 1**: LVMs have fewer parameters than models that directly represent correlation in the visible space.
- **Pros 2**: good for compression of $\boldsymbol{x}$.
- **Cons**: harder to fit than models with no latent variables.

Outline

Latent variable models

K-means Clustering

Mixture models

The EM algorithm

K-Means Objective: Compression

- Observed feature vectors: $\boldsymbol{x}_i \in \mathbb{R}^d, \quad i = 1, \ldots, N.$
- Hidden cluster labels: $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, \ldots, N.$
- Hidden cluster centers: $\boldsymbol{\mu}_k \in \mathbb{R}^d, \quad k = 1, \ldots, K.$

$$J(\boldsymbol{z}, \boldsymbol{\mu} | \boldsymbol{x}, K) = \sum_{k=1}^{K} \sum_{i | z_i = k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 = \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{\mu}_{z_i}\|^2$$

$$J(\boldsymbol{z}, \boldsymbol{\mu} | \boldsymbol{x}, K) = \sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2, \quad z_{ik} = \mathbb{I}(z_i = k)$$

- K-Means alternates between:
  - $\boldsymbol{z}^t = \text{argmin}_{\boldsymbol{z}} \, J(\boldsymbol{z}, \boldsymbol{\mu}^{t-1} | \boldsymbol{x}, K)$
  - $\boldsymbol{\mu}^t = \text{argmin}_{\boldsymbol{\mu}} \, J(\boldsymbol{z}^t, \boldsymbol{\mu} | \boldsymbol{x}, K)$

K-Means Algorithm

Objective function: $J(\boldsymbol{z}, \boldsymbol{\mu}|\boldsymbol{x}, K) = \sum_{k=1}^{K} \sum_{i=1}^{N} z_{ik} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2$.

---

given Choose random cluster centers. $\boldsymbol{\mu}^{(0)}$.

repeat

1. Assignment Step: $\boldsymbol{z}^t = \text{argmin}_{\boldsymbol{z}}\, J(\boldsymbol{z}, \boldsymbol{\mu}^{t-1}|\boldsymbol{x}, K)$, $\quad z_i^{(t)} = \text{argmin}_k \|\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t-1)}\|^2$.
2. Mean Update Step: $\boldsymbol{\mu}^t = \text{argmin}_{\boldsymbol{\mu}}\, J(\boldsymbol{z}^t, \boldsymbol{\mu}|\boldsymbol{x}, K)$, $\quad \boldsymbol{\mu}_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^{N} z_{ik} \boldsymbol{x}_i$.

return z_{ik}.

Step 1: assign data to closest cluster centers, breaking ties arbitrarily.
Step 2: means of data assigned to each cluster center (least squares).

Illustration of K-Means

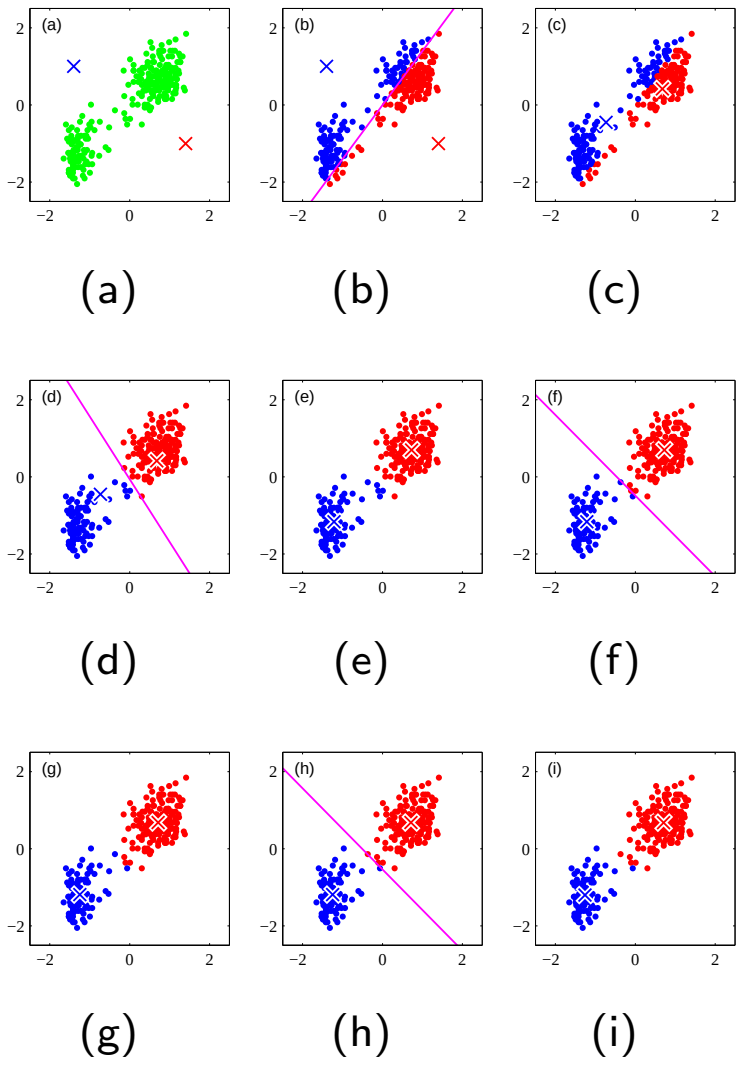


Figure: Illustration of the K-means algorithm using the re-scaled Old Faithful data set. Figure generated by `KmeansDemoFaithful`.

K-Means Implementation & Properties

Initialization: Choose random cluster centers $\boldsymbol{\mu}_{(0)}$

- Should be distinct (breaking symmetry) and in "region" of data.
- Common heuristic: randomly pick K data points.
- K-Means++: randomly pick K widely separated data points.

Theoretical Guarantees:

- Converges after finitely many iterations $z^{(t+1)} = z^{(t)}$.
- Worst-case convergence time poor (super-polynomial in N).
- Different initializations may produce very different solutions.
- Converged objective may be arbitrarily worse than optimum, but smart initializations (K-Means++) do allow some guarantees.
- In practice, can usually still find "useful" local optima.
- Optimal reconstruction error always decreases with $K, 0$ if $K = N$.

Outline

Latent variable models

K-means Clustering

Mixture models

The EM algorithm

Gaussian Mixture Models

- Observed feature vectors: $\boldsymbol{x}_i \in \mathbb{R}^d, \quad i = 1, \ldots, N.$
- Hidden cluster labels: $z_i \in \{1, 2, \ldots, K\}, \quad i = 1, \ldots, N.$
- Hidden cluster centers: $\boldsymbol{\mu}_k \in \mathbb{R}^d, \quad k = 1, \ldots, K.$
- Hidden mixture covariances: $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}, \quad k = 1, \ldots, K.$
- Hidden mixture probabilities: $\pi_k, \quad \sum_{i=1}^{K} \pi_k = 1$
- Gaussian mixture generative model:

$$p(z_i) = \mathsf{Cat}(z_i|\boldsymbol{\pi})$$
$$p(\boldsymbol{x}_i|z_i) = N(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$$
$$p(\boldsymbol{x}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \Sigma) = \sum_{z_i=1}^{K} \pi_{z_i} N(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$$

Unsupervised Learning

- Learning:

$$\underset{\boldsymbol{\pi},\boldsymbol{\theta}}{\operatorname{argmax}} \ln p(\boldsymbol{\pi}) + \ln p(\boldsymbol{\theta}) + \sum_{i=1}^{N} \left[\sum_{z_i} p(z_i|\boldsymbol{\pi}) p(\boldsymbol{x}_i|z_i, \boldsymbol{\theta}) \right]$$

- No notion of training and test data: labels are never observed.
- As before, maximize posterior probability of model parameters.
- For hidden variables associated with each observation, we marginalize over possible values rather than estimating.
  - Fully accounts for uncertainty in these variables.
  - There is one hidden variable per observation, so cannot perfectly estimate even with infinite data.
- Must use generative model (discriminative degenerates).
- Learning is harder
  - In fully observed iid settings, the log likelihood decomposes into a sum of local terms.

$$\ell(\boldsymbol{\theta}) = \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \ln p(\boldsymbol{z}|\boldsymbol{\theta}_{\boldsymbol{z}}) + \ln p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}_{\boldsymbol{x}})$$

  - With latent variables, all the parameters become coupled together via marginalization

$$\ell(\boldsymbol{\theta}) = \ln \sum p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \ln \sum p(\boldsymbol{z}|\boldsymbol{\theta}_{\boldsymbol{z}}) p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}_{\boldsymbol{x}})$$

Singularities: ML for Gaussian Mixtures

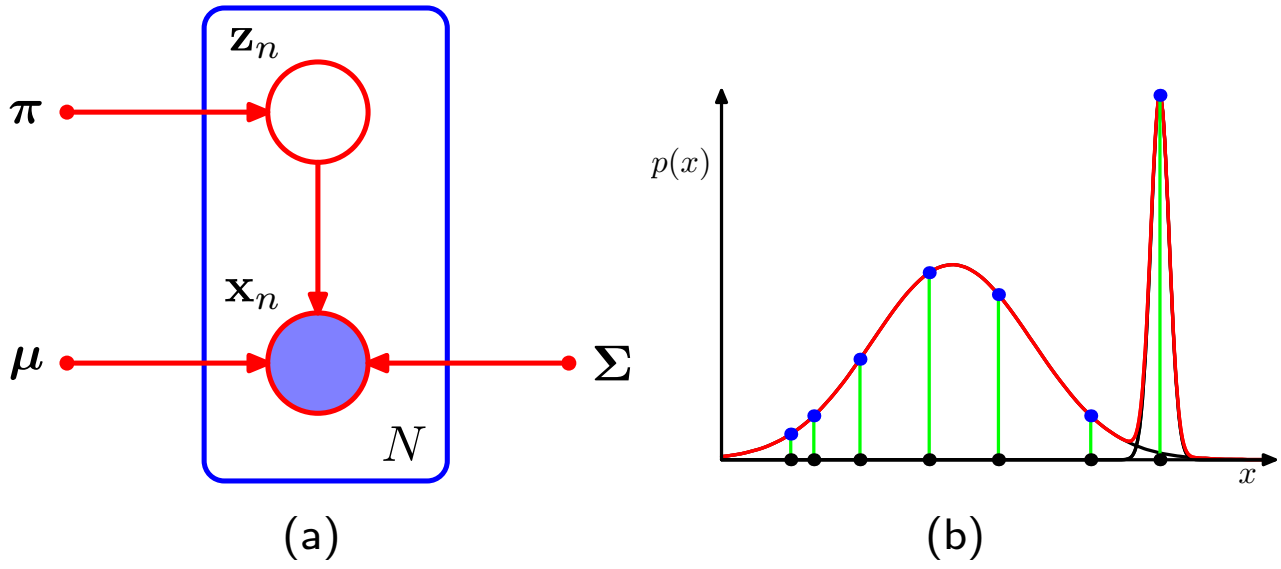


Figure: (a) Graphical representation of a Gaussian mixture model. (b) Illustration of how singularities in the likelihood function arise with mixtures of Gaussians.

Unsupervised Learning Algorithms

- Initialization: Randomly select starting parameters.
- Estimation: Given parameters, infer likely hidden data.
  - Similar to testing phase of supervised learning.
- Learning: Given hidden & observed data, find likely parameters.
  - Similar to training phase of supervised learning.
- Iteration: Alternate estimation & learning until convergence.

Outline

Latent variable models

K-means Clustering

Mixture models

The EM algorithm

Expectation Maximization (EM)

Goal: maximize the likelihood function $p(\boldsymbol{X}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

- **Input:** $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}), p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta})$.
- Choose an initial setting for the parameters $\boldsymbol{\theta}^{\text{old}}$.
- **E step.** Evaluate $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}})$.
- **M step.** Evaluate $\boldsymbol{\theta}^{\text{new}}$ given by

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

  where $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$
- Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

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

  and return to step 2.

Example: EM for Gaussian Mixtures

- Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.
- **E step.** Evaluate the responsibilities (the expected value of the sufficient statistics of the hidden variables) using the current parameter values

$$\gamma(z_{ik}) = p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\pi}, \boldsymbol{\theta}) = \frac{\pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j N(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- **M step.** Re-estimate the parameters using the current responsibilities (i.e. expected value of the hidden variables)

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{ik}) \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{ik})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

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

where $N_k = \sum_{n=1}^{N} \gamma(z_{ik})$.

Example: EM for Gaussian Mixtures(cont'd)

- Evaluate the log likelihood

$$\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

  and check for convergence of the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

Illustration of EM Algorithm for GGM

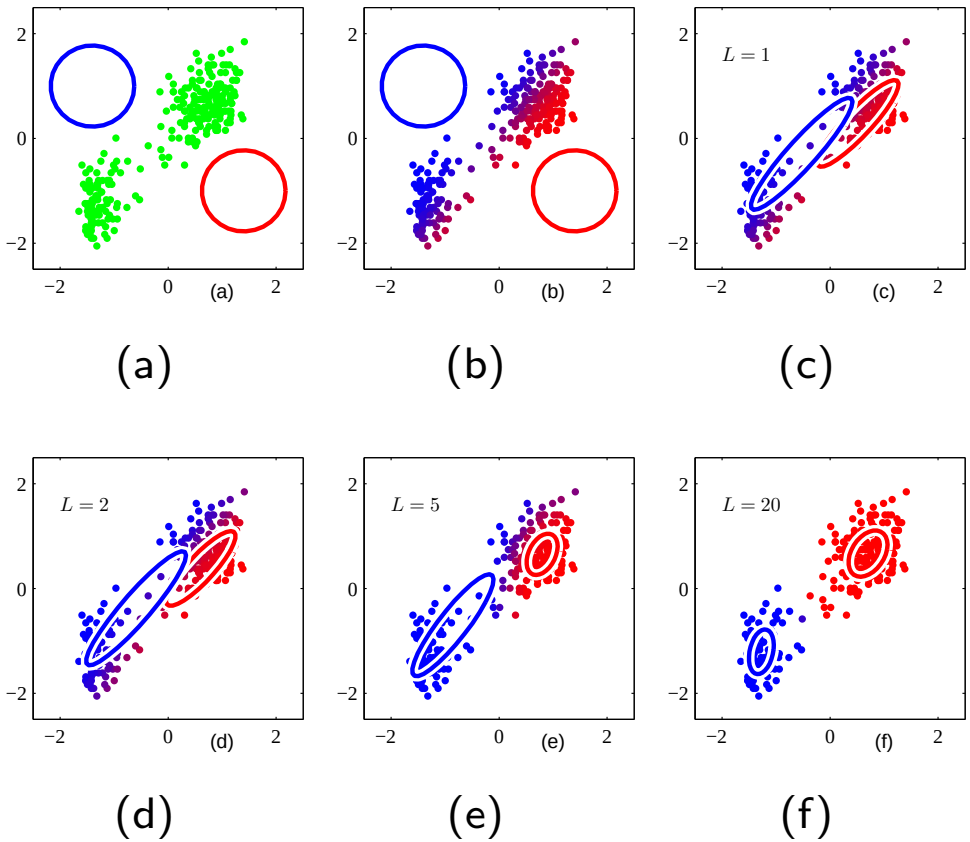


Figure: Illustration of the EM algorithm using the Old Faithful set. Figure generated by `MixGaussDemoFaithful`.

EM as Lower Bound Maximization

$$\begin{aligned}
\ln p(\boldsymbol{x}|\boldsymbol{\theta}) &= \ln(\sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})) = \ln(\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}))}{q(\boldsymbol{z})} \\
&\geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln(\frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{q(\boldsymbol{z})}) \quad \text{(Jensen's Inequality)} \\
&\geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln q(\boldsymbol{z}) = L(q, \boldsymbol{\theta})
\end{aligned}$$

- Initialization: Randomly select starting parameters $\boldsymbol{\theta}_{(0)}$.
- Inference: Given parameters, infer likely hidden data.

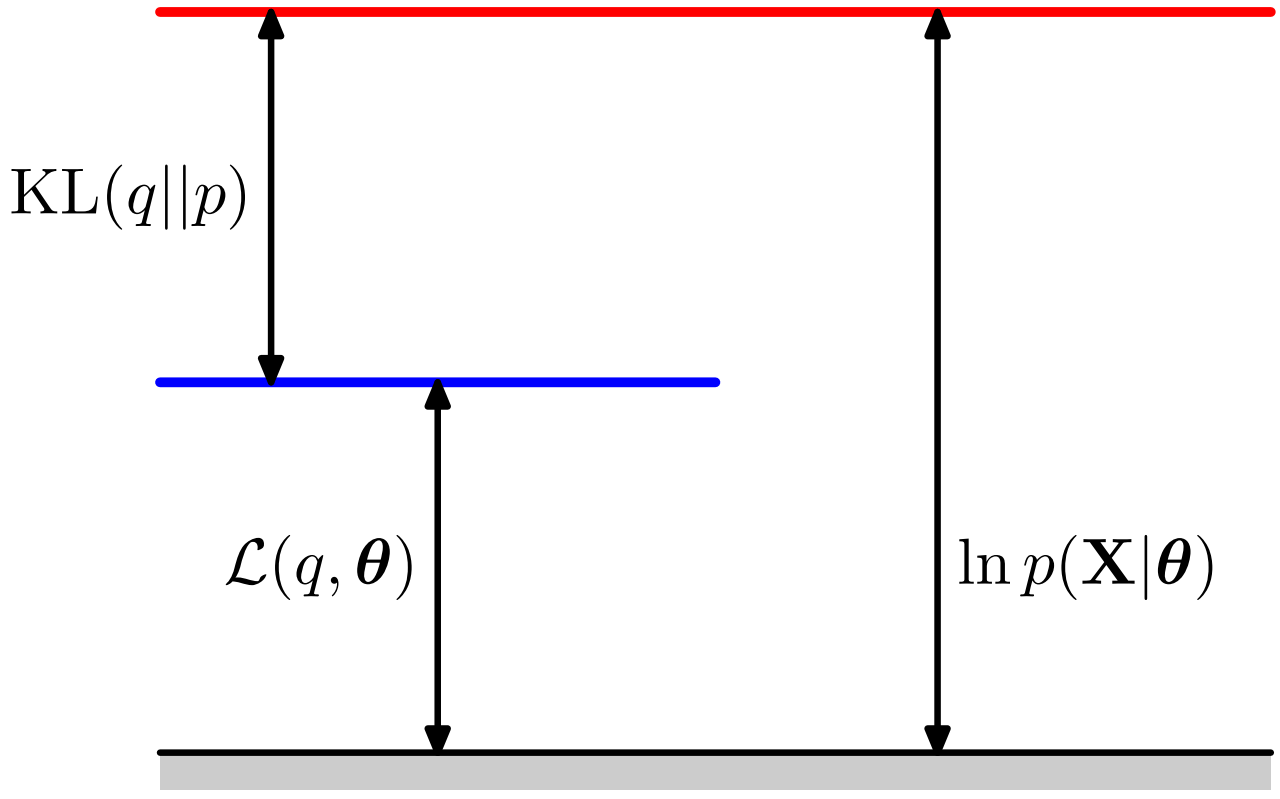
$$q^{(t)} = \underset{q}{\operatorname{argmax}}\, L(q, \boldsymbol{\theta}^{(t-1)})$$

- Learning: Given hidden & observed data, find likely parameters.

$$\boldsymbol{\theta}^{(t)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(q^{(t)}, \boldsymbol{\theta})$$

- Iteration: Alternate estimation & learning until convergence.

Lower Bounds on Marginal Likelihood



EM: Expectation Step

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}) \geq \sum_z q(z) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln q(\boldsymbol{z}) = L(q, \boldsymbol{\theta})$$

$$q^{(t)} = \operatorname*{argmax}_q L(q, \boldsymbol{\theta}^{(t-1)})$$

- One can also show this result using variational calculus

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}) - \ln q(\boldsymbol{z}) = L(q, \boldsymbol{\theta}) = \mathsf{KL}(q||p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}))$$

- General solution, for any probabilistic model

$$q^{(t)} = \operatorname*{argmax}_q L(q, \boldsymbol{\theta}^{(t-1)})$$

- For mixture models, data independent given parameters

$$p(z_i|\boldsymbol{\pi}) = \mathsf{Cat}(z_i|\boldsymbol{\pi})$$

$$p(\boldsymbol{x}_i|z_i, \boldsymbol{\theta}) = p(\boldsymbol{x}_i|\boldsymbol{\theta}_{z_i})$$

$$\gamma(z_{ik}) = p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\pi}, \boldsymbol{\theta}) = \frac{\pi_k N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j N(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Illustration of the E step

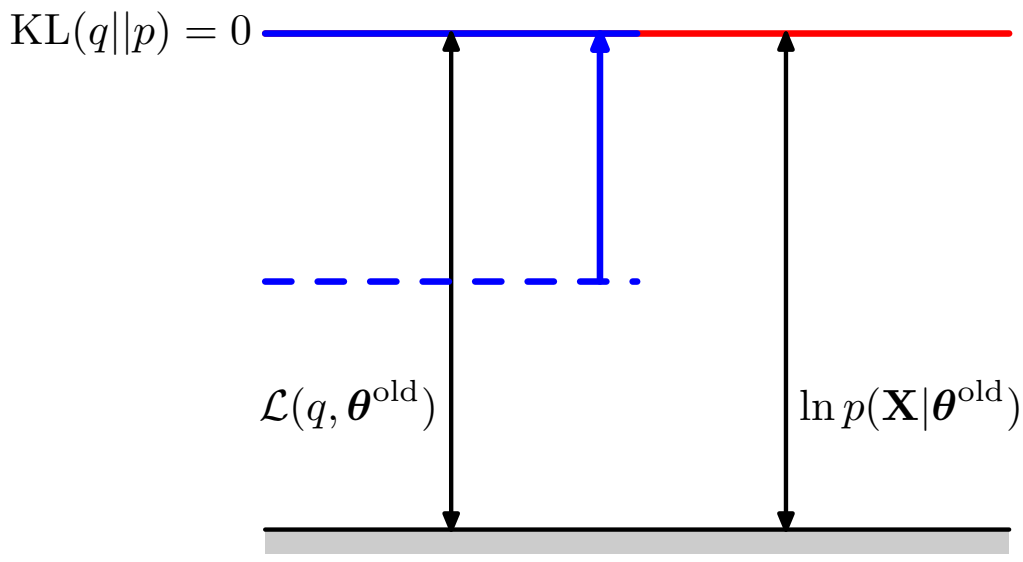


Figure: Illustration of the E step of the EM algorithm. The q distribution is set equal to the posterior distribution for the current parameter values $\boldsymbol{\theta}^{\mathrm{old}}$, causing the lower bound to move up to the same value as the log likelihood function, with the KL divergence vanishing.

EM: Maximization Step

$$\ln p(\boldsymbol{x}|\boldsymbol{\theta}) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln q(\boldsymbol{z}) = L(q, \boldsymbol{\theta})$$

$$\boldsymbol{\theta}^{(t)} = \operatorname*{argmax}_{\boldsymbol{\theta}} L(q^{(t)}, \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$$

- Unlike E-step, no simplified general solution.
- Applying to GMM

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{ik}) \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{ik}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

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

Illustration of the M step

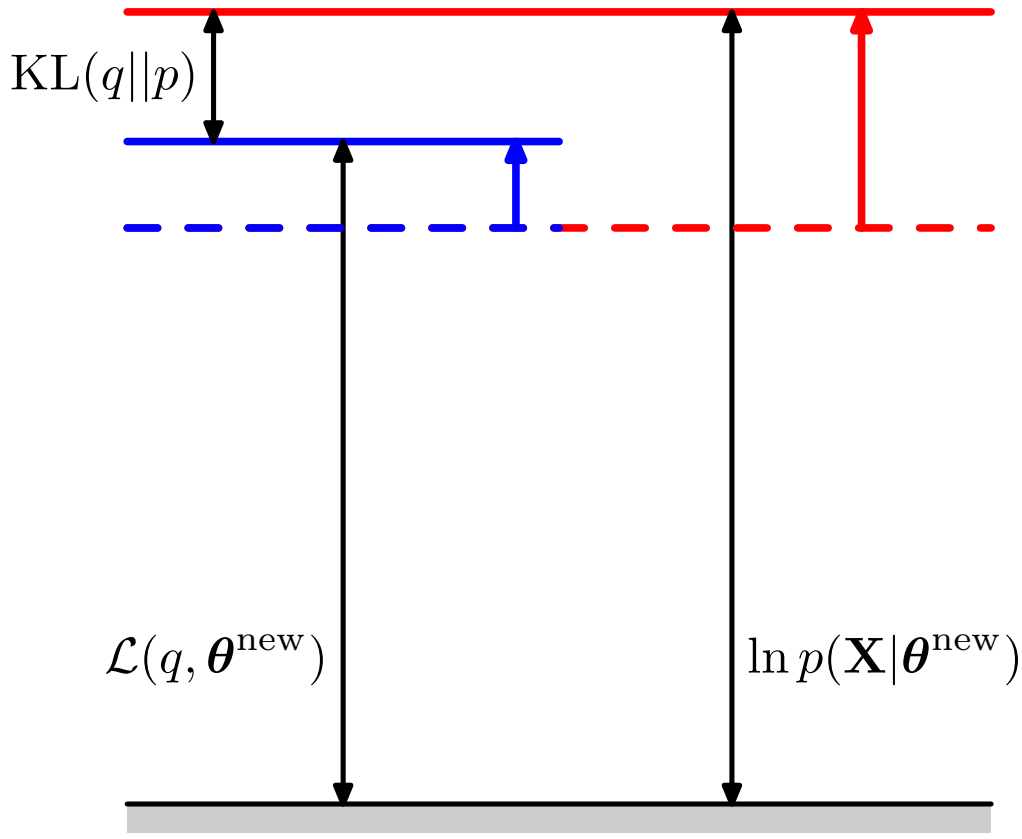


Figure: Illustration of the M step of the EM algorithm. The distribution $q(\boldsymbol{Z})$ is held fixed and the lower bound $L(q, \boldsymbol{\theta})$ is maximized with respect to the parameter vector $\boldsymbol{\theta}$ to give a revised value $\boldsymbol{\theta}^{\text{new}}$. Because the KL divergence is nonnegative, this causes the log likelihood $\ln p(\boldsymbol{X}|\boldsymbol{\theta})$ to increase by at least as much as the lower bound does.

EM: A Sequence of Lower Bounds

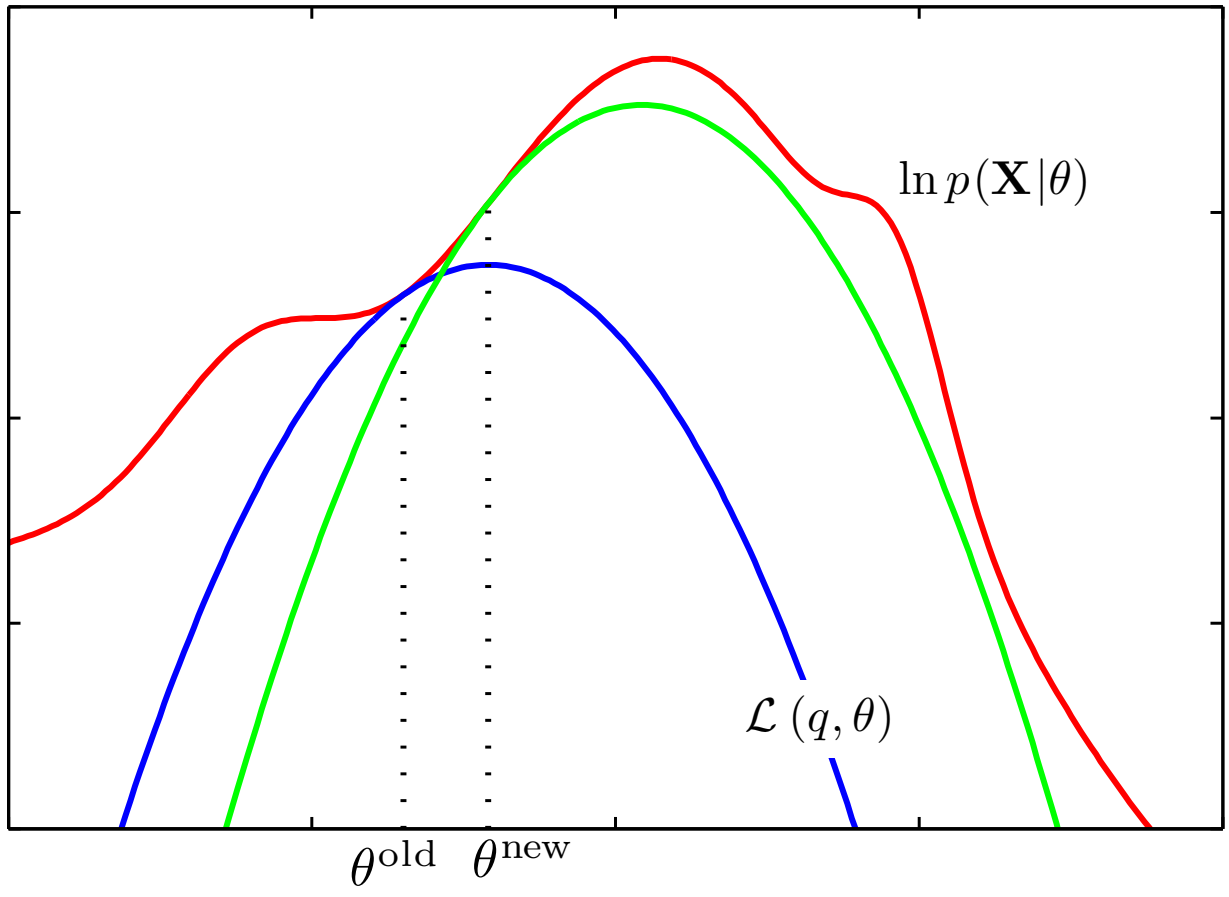


Figure: The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values.