PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 10: MIXTURE MODELS AND EM

Mixture Models

- Define a joint distribution over observed and latent variables
  - The corresponding distribution of the observed variables alone is obtained by marginalization
- Allows relatively complex marginal distributions over observed variables to be expressed in terms of more tractable joint distributions over the expanded space of observed and latent variables
- The introduction of latent variables thereby allows complicated distributions to be formed from simpler components.
- How can mixture distribution be expressed in terms of discrete latent variables?

Mixture Models (2)

- Probability mixture model is a probability distribution that is a convex combination of other probability distributions

$$f_X(x) = \sum_{i=1}^{n} a_i f_{Y_i}(x)$$

$$0 \le a_i \le 1$$

$$a_1 + \cdots + a_n = 1$$

Mixture Models (3)

- Used for:
    - Building more complex distributions
    - Clustering data
- *K*-means algorithm corresponds to a particular non-probabilistic limit of EM applied to mixtures of Gaussians

K-Means Clustering

- $\{x_n\}$ – N observations of a random D-dimensional Euclidian variable $\mathbf{x}$
- Partition the data set into some number K of clusters
- Suppose that the value K is given
- Cluster: group of data points whose inter-point distances are small compared with the distances to points outside of the cluster
- Introduce a set of K D-Dimensional vectors: $\{\mu_k\}$ that define a prototype associated with the k-th cluster
- Think of μ_k as representing the center of the clusters
- Find an assignment of data points to clusters such that the sum of the squares of the distances of each data point to its closest vector μ_k is a minimum

K-Means Clustering (2)

- Use the 1-of-K coding scheme
- Define an objective function called the distortion measure:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

- Goal: find the values for $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize J

Algorithm – Idea

1. Choose some initial values for μ_k
2. Repeat (until convergence)
    3. Step 1. Minimize J with respect to r_{nk}, keeping μ_k fixed – **E**(xpectation) step
    4. Step 2. Minimize J with respect to μ_k, keeping r_{nk} fixed – **M**(aximization) step

- Can be seen as a simple variant of the EM algorithm

E step

- Determination of r_{nk}
- J is a linear combination of r_{nk}
- The terms involving different n are independent
- Optimize for each n separately by choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $||\mathbf{x}_n - \boldsymbol{\mu}_k||$
- Formally:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

- Simply assign the n-th data point to the closest cluster centre

M step

- Determination of μ_k
- J is a quadratic function of μ_k

$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

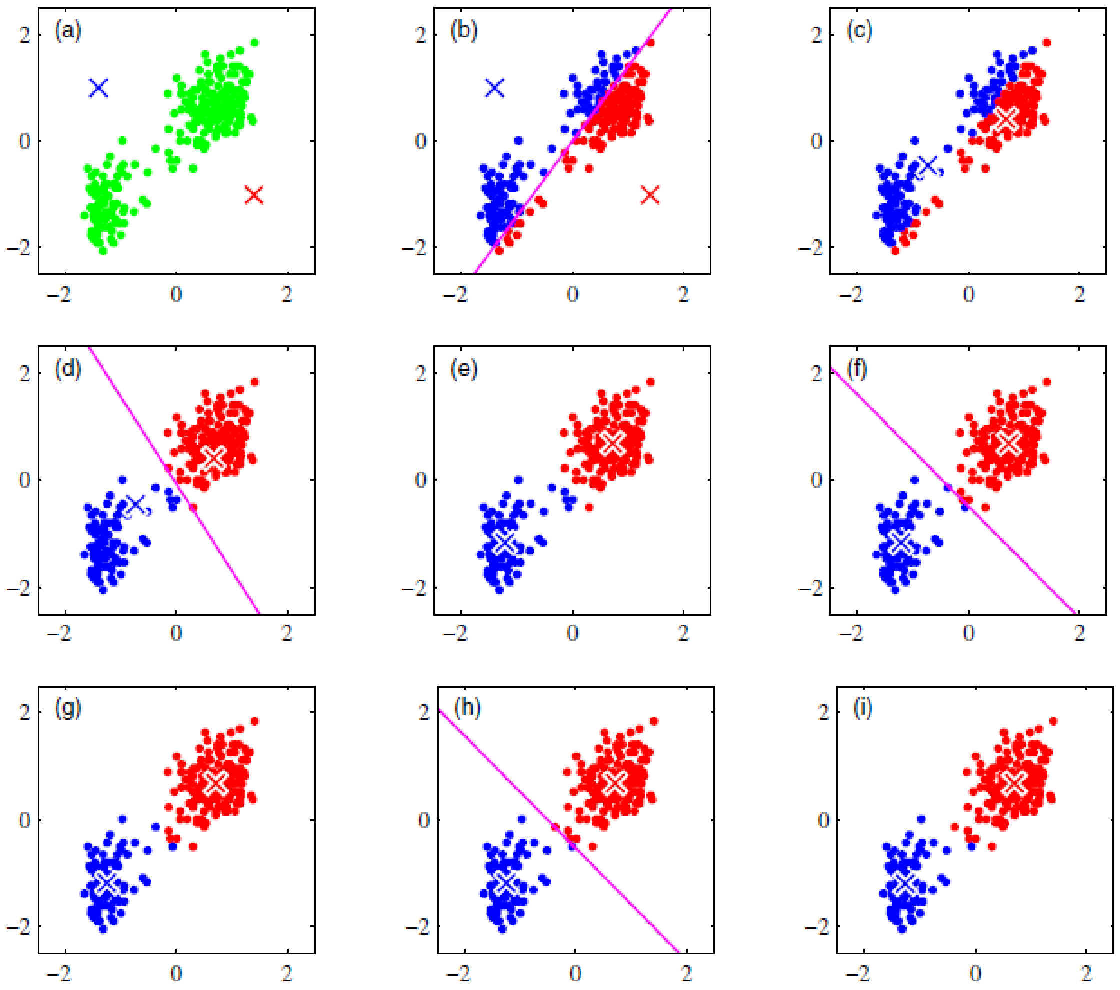
- The solution is: $\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}.$

- Denominator: the number of points in cluster k
- Set $\boldsymbol{\mu}_k$ equal to the mean of all of the data points $\mathbf{x}_n$ assigned to cluster k => K-MEANS ALGORITHM

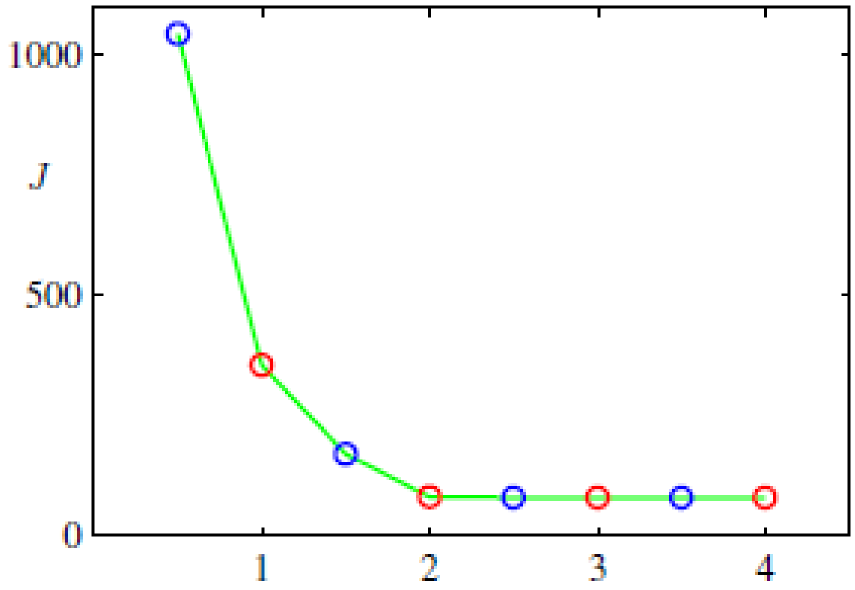
Convergence

- Stop when the assignments do not change in 2 successive steps
- Stop after a maximum number of steps
- Each step reduces the value of J => the convergence of the algorithm is assured
- It may converge to a local rather than global minimum of J

Example



Example



Improvements

- Initialize the initial values of μ_k to random subset of K data points
- The direct implementation of the algorithm is quite slow because at each E step it is needed to compute the distance between each data point and each cluster prototype vector
- Improve this computation
- There is also an on-line algorithm, that uses the following formula for each new data point:

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{old}})$$

- Use soft assignments of the points to clusters

K-medoids

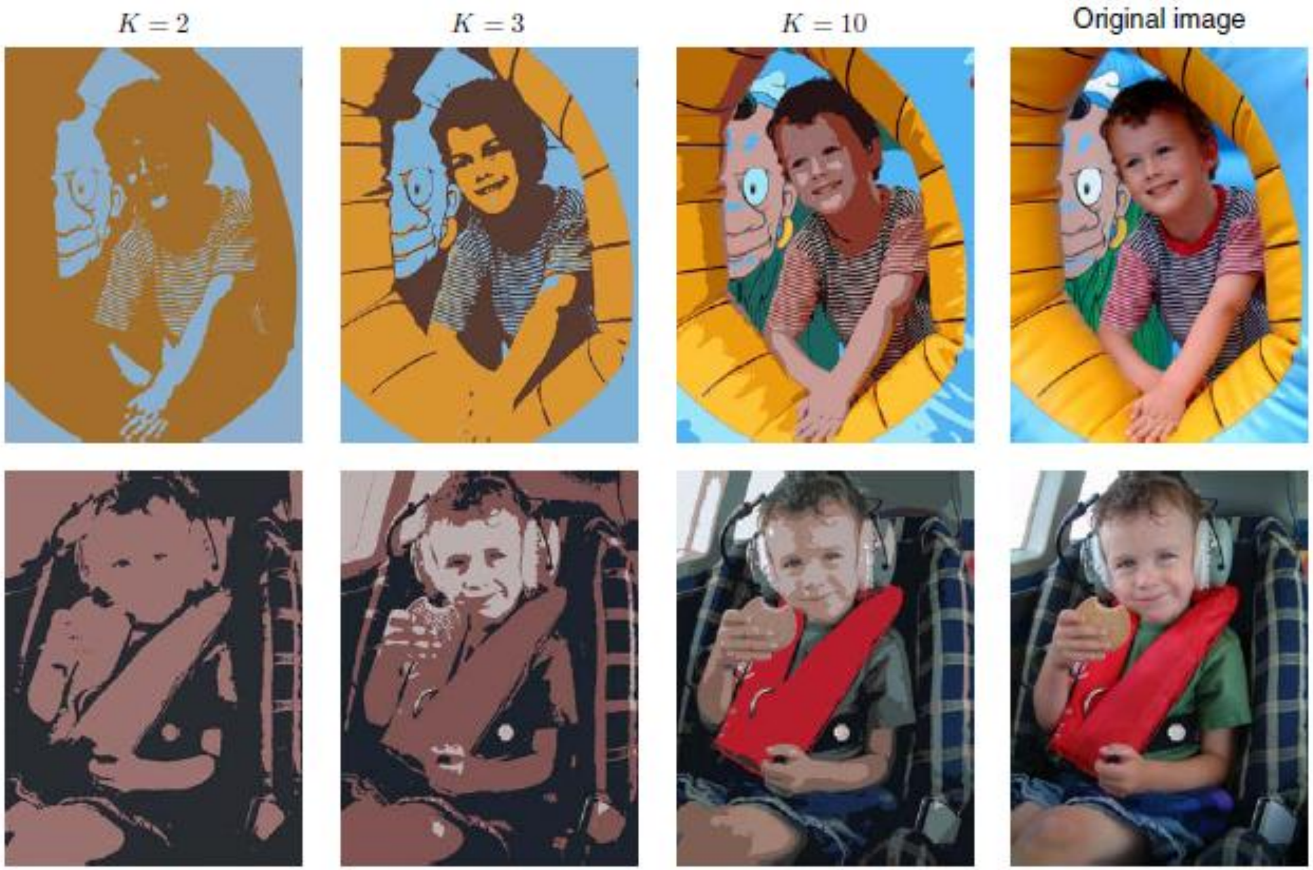
- Uses a more general dissimilarity measure between the data points

$$\tilde{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

- The M step is potentially more complex than for K-means, and so it is common to restrict each cluster prototype to be equal to one of the data vectors assigned to that cluster

Application of K-Means

- Image segmentation and image compression
- Replace the color of each pixel in the original image with the one given by the corresponding cluster's color
- Simplistic approach as it takes no account of the spatial proximity of different pixels
- Similarly, we can apply the K-means algorithm to the problem of lossy data compression


$K = 2$
$K = 3$
$K = 10$
Original image

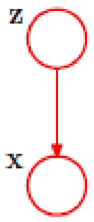
Mixtures of Gaussians

- The Gaussian mixture model: a simple linear superposition of Gaussian components
  - providing a richer class of density models than the single Gaussian
- Turn to a formulation of Gaussian mixtures in terms of discrete latent variables
  - Provides deeper insight into this important distribution
  - Serves to motivate the expectation-maximization algorithm

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

Mixtures of Gaussians (2)

- Let’s introduce a K-dimensional binary random variable **z** having a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0
  - K possible states
- Joint distribution p(**x**, **z**) in terms of a marginal distribution p(**z**) and a conditional distribution p(**x**|**z**)



$$0 \leqslant \pi_k \leqslant 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

- The marginal distribution over **z** is specified in terms of the mixing coefficients π_k, such that $p(z_k = 1) = \pi_k$

Mixtures of Gaussians (3)

- Then, the marginal distribution of **x**:

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

$$\Downarrow$$

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

Mixtures of Gaussians (4)

- Thus the marginal distribution of **x** is a Gaussian mixture
- Consider several observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$
- We have represented the marginal distribution in the form *p*(**x**) = Sum_**z**(*p*(**x**, **z**))
- => for every observed data point $\mathbf{x}_n$ there is a corresponding latent variable $\mathbf{z}_n$
- We have therefore found an equivalent formulation of the Gaussian mixture involving an explicit latent variable

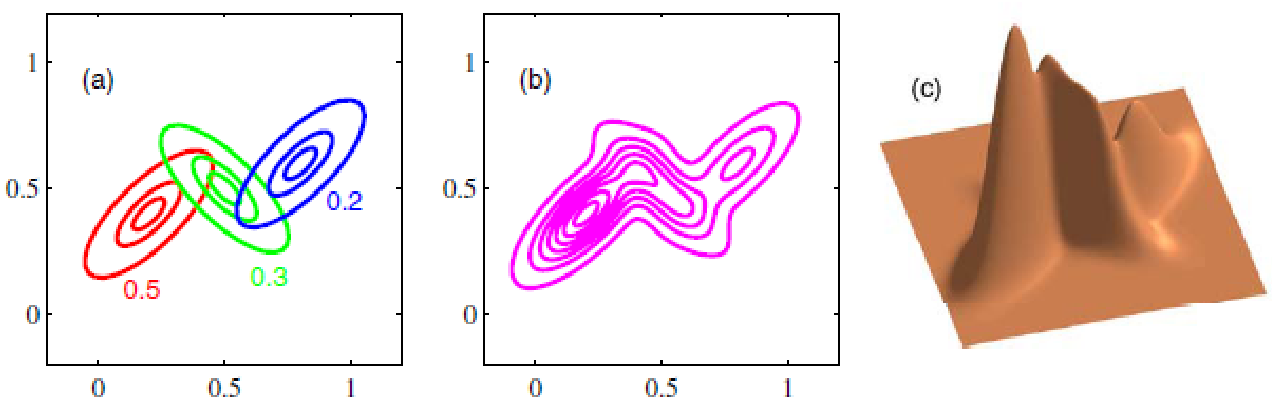
- Advantage: work with p(x, z) instead of p(x)

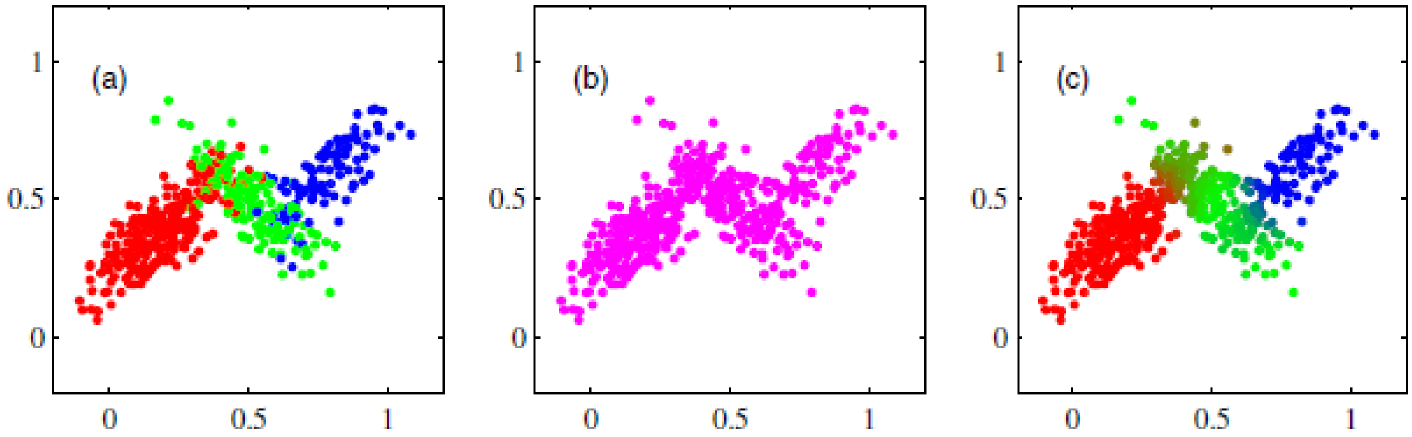
Mixtures of Gaussians (5)

- Use Bayes theorem to compute γ(z_k) – the posterior probability once **x** is observed
- Can also be viewed as the responsibility that component k takes for ‘explaining’ the observation **x**
- π_k is the prior probability of z_k=1

$$\begin{aligned}\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.\end{aligned}$$

Example

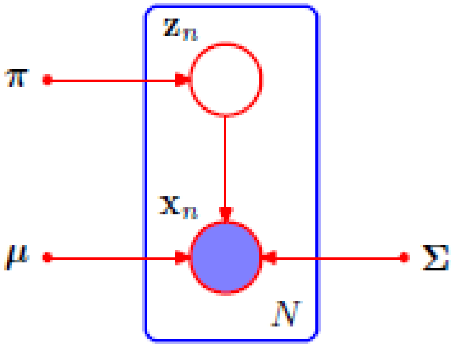




Maximum Likelihood

- Suppose we have a data set of observations
    - $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
    - Want to model it using a mixture of Gaussians
- Represent it as an N x D matrix **X** with rows x_n^T
- The corresponding latent variables will be denoted by an N × K matrix **Z** with rows z_n^T
- If we assume that the data points are drawn independently from the distribution, then we can express the Gaussian mixture model for this i.i.d. data set

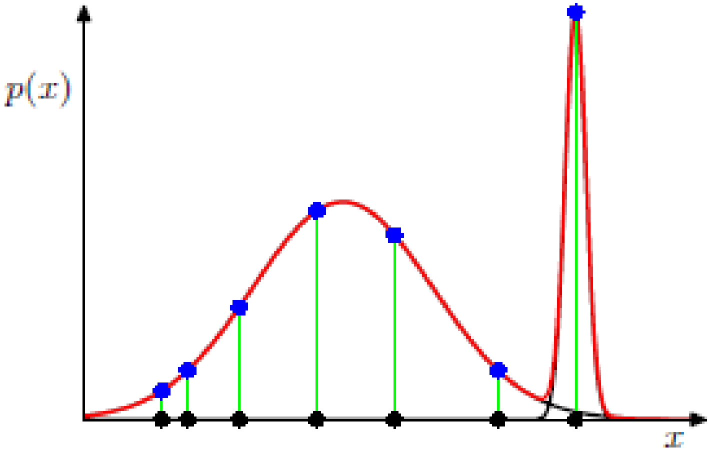
Maximum Likelihood (2)



- The log of the likelihood function:

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

Maximum Likelihood (3)



- We want to maximize the ML
- But, there is a significant problem associated with the maximum likelihood framework applied to Gaussian mixture models, due to the presence of singularities

Maximum Likelihood (4)

- Consider the simple mixture model on the previous slide
- Suppose that one of the components has its mean, μ_j, equal with one of the data points, x_n
- The Gaussians also have a simple covariance
- Then, x_n will contribute to the likelihood with the value:

$$\mathcal{N}(\mathbf{x}_n|\mathbf{x}_n, \sigma_j^2\mathbf{I}) = \frac{1}{(2\pi)^{1/2}}\frac{1}{\sigma_j}.$$

- If $\sigma_j \rightarrow 0$, then this term goes to infinity => log likelihood function will also go to infinity
- Thus the maximization of the log likelihood function is not a well posed problem because such singularities will always be present and will occur whenever one of the Gaussian components 'collapses' onto a specific data point

Maximum Likelihood (5)

- This problem did not arise in the case of a single Gaussian distribution
  - If a single Gaussian collapses onto a data point, it will contribute multiplicative factors to the likelihood function arising from the other data points and these factors will go to zero exponentially fast, giving an overall likelihood that goes to zero rather than infinity.
- However, once we have (at least) two components in the mixture:
  - one of the components can have a finite variance and therefore assign finite probability to all of the data points
  - the other component can shrink onto one specific data point and thereby contribute an ever increasing additive value to the log likelihood
- This difficulty does not occur for a Bayesian approach

Maximum Likelihood (6)

- In applying maximum likelihood to Gaussian mixture models we must take steps to avoid finding such pathological solutions and instead seek local maxima of the likelihood function that are well behaved
- We can hope to avoid the singularities by using suitable heuristics:
  - Detecting when a Gaussian component is collapsing and resetting its mean to a randomly chosen value while also resetting its covariance to some large value, and then continuing with the optimization

Maximum Likelihood (7)

- Maximizing the log likelihood function for a Gaussian mixture model is a more complex problem than for the case of a single Gaussian
- The difficulty arises from the presence of the summation over k that appears inside the logarithm
- The logarithm function no longer acts directly on the Gaussian. If we set the derivatives of the log likelihood to zero, we will no longer obtain a closed form solution, as we shall see shortly
- Solutions:
    - Gradient based optimization techniques
    - EM Algorithm

EM for Gaussian Mixtures

- The expectation-maximization (EM) algorithm is a powerful method for finding maximum likelihood solutions for models with latent variables
- However, EM has a much broader applicability
- First, let’s motivate the EM algorithm in the context of a Gaussian mixture model

EM for Gaussian Mixtures (2)

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Derivative of log likelihood with respect to μ_k

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n-\boldsymbol{\mu}_k)$$

- Multiplying by Σ_k^{-1}

$$\boxed{\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n}$$

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

EM for GM- Interpretation

- We can interpret N_k as the effective number of points assigned to cluster k
- The form of this solution:
  - The mean $\boldsymbol{\mu}_k$ for the k-th Gaussian component is obtained by taking a weighted mean of all of the points in the data set
  - The weighting factor for data point $\mathbf{x}_n$ is given by the posterior probability $\gamma(z_{nk})$ that component k was responsible for generating $\mathbf{x}_n$

EM for Gaussian Mixtures (3)

- Similarly, the derivative of log likelihood with respect to Σ_k

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

- Has the same form as the corresponding result for a single Gaussian fitted to the data set, but again:
  - Each data point is weighted by the corresponding posterior probability
  - The denominator is given by the effective number of points associated with the corresponding component

EM for Gaussian Mixtures (4)

- Want to find the mixing coefficients π_k
- => Maximize the log likelihood with respect to π_k
- But, we have an additional requirement
  - The mixing coefficients must sum to one
- Introduce a Lagrange multiplier and maximize:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

Multiply with π_k
Sum over k

$$\lambda = -N$$

$$\pi_k = \frac{N_k}{N}$$

- The mixing coefficient for the k-th component is given by the average responsibility which that component takes for explaining the data points

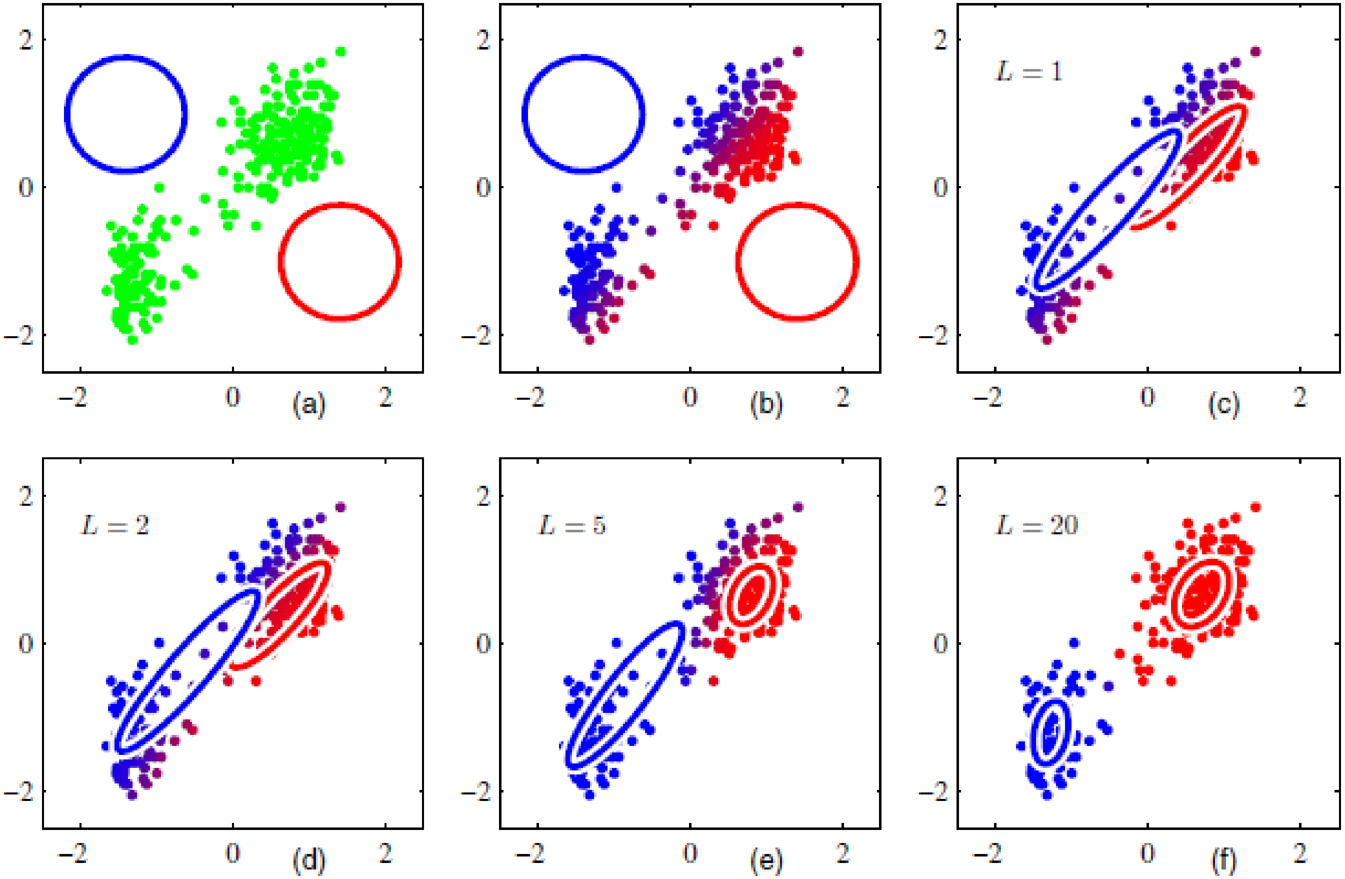
EM for Gaussian Mixtures (5)

- These results do not constitute a closed-form solution for the parameters of the mixture model because the responsibilities $\gamma(z_{nk})$ depend on those parameters in a complex way
- A simple iterative scheme for finding a solution to the maximum likelihood problem, which as we shall see turns out to be an instance of the EM algorithm
- First choose some initial values for the means, covariances, and mixing coefficients
- Then, we alternate a sequence of E steps and M steps

EM for Gaussian Mixtures (6)

- E(xpectation) step: use the current values for the parameters to evaluate the posterior probabilities, or responsibilities
- M(aximization) step: use these responsibilities to re-estimate the means, covariances, and mixing coefficients using the previous formulas
  - First find the new means
  - Then find the new covariances
- Each update to the parameters resulting from an E step followed by an M step is guaranteed to increase the log likelihood function
- The algorithm has converged when the change in the log likelihood function, or alternatively in the parameters, falls below some threshold

Example



EM for Gaussian Mixtures (7)

- The EM algorithm takes many more iterations to reach (approximate) convergence compared with the *K*-means algorithm
- Each cycle requires significantly more computation
- Therefore, first run the *K*-means algorithm in order to find a suitable initialization for a Gaussian mixture model that is subsequently adapted using EM
  - The covariance matrices can conveniently be initialized to the sample covariances of the clusters found by the *K*-means algorithm
  - The mixing coefficients can be set to the fractions of data points assigned to the respective clusters.
- Techniques must be employed to avoid singularities of the likelihood function in which a Gaussian component collapses onto a particular data point
- There will generally be multiple local maxima of the log likelihood function, and that EM is not guaranteed to find the largest of these maxima

EM for Gaussian Mixtures - Algorithm

Input: Gaussian mixture model, data set

Goal: Maximize the likelihood function

with respect to the parameters

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood
2. **E step**. Evaluate the responsibilities using the current parameters:

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

EM for Gaussian Mixtures - Algorithm

3. **M step.** Re-estimate the parameters using the current responsibilities

$$\begin{aligned}\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\mathbf{x}_n \\ \boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n-\boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \\ \pi_k^{\text{new}} &= \frac{N_k}{N}\end{aligned}$$

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

4. Evaluate the log likelihood and check for convergence of either the parameters or the log likelihood

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

5. If the convergence criterion is not satisfied return to step 2.

Alternative View of EM

- What is the role of the latent variables in the EM algorithm?
- The goal of the EM algorithm is to find maximum likelihood solutions for models having latent variables
- **X** – observed data
- **Z** – latent variables
- θ – set of all the parameters of the model
- The log likelihood is given by:

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

- The presence of the sum prevents the logarithm from acting directly on the joint distribution, resulting in complicated expressions for the maximum likelihood solution

Alternative View of EM (2)

- Suppose that, for each observation in **X**, we know the corresponding value of the latent variable **Z**
- {**X**, **Z**} is called the *complete* data set
- The actual observed data **X** is *incomplete*
- The likelihood function for the complete data set simply takes the form $\ln p(\mathbf{X},\mathbf{Z}|\theta)$
- We shall suppose that maximization of this complete-data log likelihood function is straightforward

Alternative View of EM (3)

- In practice, we are not given the complete data set {$\mathbf{X}$,$\mathbf{Z}$}, but only the incomplete data $\mathbf{X}$
- The values of the latent variables in $\mathbf{Z}$ is given only by the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$
- Because we cannot use the complete-data log likelihood, we consider instead its expected value under the posterior distribution of the latent variable, which corresponds (as we shall see) to the E step of the EM algorithm
- In the subsequent M step, we maximize this expectation
- If the current estimate for the parameters is denoted $\boldsymbol{\theta}^{old}$, then a pair of successive E and M steps gives rise to a revised estimate $\boldsymbol{\theta}^{new}$
- The algorithm is initialized by choosing some starting value for the parameters $\boldsymbol{\theta}_0$
- The use of the expectation may seem somewhat arbitrary. We shall see the motivation for this choice later

Alternative View of EM (4)

E step

1. Use $\boldsymbol{\theta}^{\text{old}}$ to find the posterior distribution of the latent variables $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$
2. Use this posterior distribution to find the expectation of the complete-data log likelihood evaluated for some general parameter value θ

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

Alternative View of EM (5)

M step

1. Determine the revised parameter estimate θ^{new} by maximizing the expectation computed in the previous step

$$\theta^{\text{new}} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{\text{old}}).$$

- Notice! In the definition of the expectation, the logarithm acts directly on the joint distribution => the corresponding M-step maximization will, by supposition, be tractable

Alternative View of EM (6)

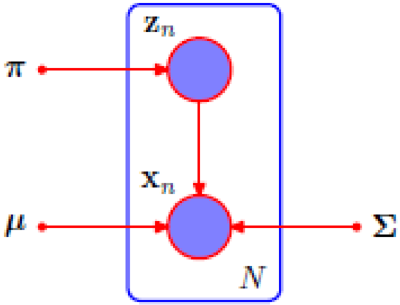
- Convergence: either the log likelihood or the parameter values
- Repeat E-M steps until convergence
- Each cycle of EM will increase the incomplete-data log likelihood (unless it is already at a local maximum)

Remarks

- The EM algorithm can also be used to find MAP solutions for models in which a prior p($\boldsymbol{\theta}$) is defined over the parameters
- The E step remains the same as in the maximum likelihood case
- In the M step, the quantity to be maximized is given by Q($\boldsymbol{\theta}$, $\boldsymbol{\theta}^{old}$) + lnp($\boldsymbol{\theta}$)
- Suitable choices for the prior will remove singularities

Revisiting Gaussian Mixtures

- The graphical model for the complete data set {X, Z}



- Maximize its likelihood

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

$$\Downarrow$$

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

Revisiting Gaussian Mixtures (2)

- Comparison with the log likelihood function for the incomplete data shows that the summation over k and the logarithm have been interchanged
- This leads to a much simpler solution to the maximum likelihood problem
- Thus the maximization with respect to a mean or a covariance is exactly as for a single Gaussian, except that it involves only the subset of data points that are ‘assigned’ to that component

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad\qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2$$

Revisiting Gaussian Mixtures (3)

- Using the same reasoning as in the previous case, the mixing coefficients are equal to the fractions of data points assigned to the corresponding components

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} z_{nk}$$

- The complete-data log likelihood function can be maximized trivially in closed form
- In practice, we do not have values for the latent variables => we consider the expectation of the complete-data log likelihood, with respect to the posterior distribution of the latent variables

Revisiting Gaussian Mixtures (4)

- This posterior distribution takes the form and hence factorizes over n, so that under the posterior distribution the {**z**$_n$} are independent

$$p(\mathbf{Z}|\mathbf{X},\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi}) \propto \prod_{n=1}^{N}\prod_{k=1}^{K}\left[\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]^{z_{nk}}$$

- The expected value of z$_{nk}$ under this distribution is just the responsibility of component k for data point x$_n$

$$\begin{aligned}\mathbb{E}[z_{nk}] &= \frac{\displaystyle\sum_{z_{nk}} z_{nk}\left[\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right]^{z_{nk}}}{\displaystyle\sum_{z_{nj}}\left[\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)\right]^{z_{nj}}} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\displaystyle\sum_{j=1}^{K}\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} = \gamma(z_{nk})\end{aligned}$$

Revisiting Gaussian Mixtures (5)

- The expected value of the complete-data log likelihood function:

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\pi})] = \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left\{\ln\pi_k + \ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\right\}.$$

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

- Proceed as follows:
  - Choose some initial values for the parameters: μ^{old}, Σ^{old} and π^{old}, and use these to evaluate the responsibilities (the E step)
  - Keep the responsibilities fixed and maximize the above formula with respect to μ_k, Σ_k and π_k (the M step) => μ^{new}, Σ^{new} and π^{new}
  - This is precisely the EM algorithm for Gaussian mixtures as derived earlier (the same formulas, etc.)

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

Relation to K-Means

- Comparison of the *K*-means algorithm with the EM algorithm for Gaussian mixtures shows that there is a close similarity
- The *K*-means algorithm performs a hard assignment of data points to clusters
- The EM algorithm makes a soft assignment based on the posterior probabilities
- We can derive the *K*-means algorithm as a particular limit of EM for Gaussian mixtures

Relation to K-Means (2)

- Gaussian mixture model where the components have the covariance matrices of the form εI, ε – variance parameter shared by all the components

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x}-\boldsymbol{\mu}_k\|^2\right\}$$

- The responsabilities are (if ε is treated like a constant)

$$\gamma(z_{nk}) = \frac{\pi_k \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\}}{\sum_j \pi_j \exp\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\}}.$$

- Consider what happens when $\varepsilon \to 0$
  - In the denominator, the term for which $\|x_n - \mu_j\|$ is smallest will go to zero most slowly => $\gamma(z_{nk})$ for the data point x_n all go to zero except for term j, for which the responsibility $\gamma(z_{nj})$ will go to unity
  - Note that this holds independently of the values of the π_k so long as none of the π_k is zero

Relation to K-Means (3)

- We obtain a hard assignment just as for K-Means: $\gamma(z_{nk}) \longrightarrow r_{nk}$
- The EM re-estimation equation for the μ_k then reduces to the K-means result
- The re-estimation formula for the mixing coefficients simply re-sets the value of π_k to be equal to the fraction of data points assigned to cluster k, although these parameters no longer play an active role in the algorithm
- Moreover, if $\varepsilon \longrightarrow 0$

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] \rightarrow -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \text{const.}$$

- Maximizing the expected complete-data log likelihood is equivalent to minimizing the distortion measure J for the K-means algorithm
- The K-means algorithm does not estimate the covariances of the clusters but only the cluster means (there exists an elliptical K-means algorithm)

The EM Algorithm in General

- The EM algorithm is a general technique for finding maximum likelihood solutions for probabilistic models having latent variables
- **X** – observed variables
- **Z** – hidden variables
- The joint distribution p(X,Z|θ) is governed by a set of parameters θ
- We want to maximize the likelihood function given by

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

The EM Algorithm in General (2)

- Supposition: direct optimization of p(X|θ) is difficult, but that optimization of the complete-data likelihood function p(X,Z|θ) is significantly easier
- Next, we introduce a distribution q(Z) defined over the latent variables
- For any choice of q(Z), the following decomposition holds

$$\boxed{\ln p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q\|p)}$$

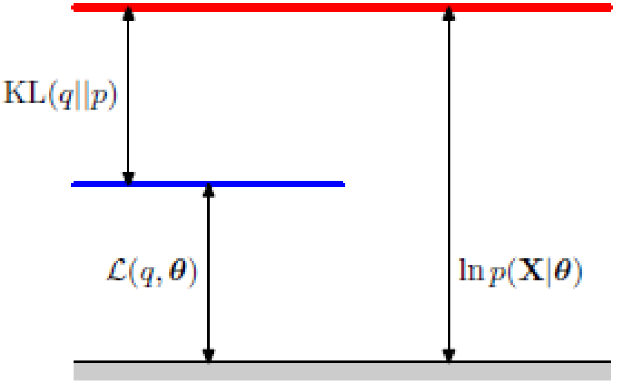
$$\begin{aligned} \mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \\ \mathrm{KL}(q\|p) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \right\}. \end{aligned}$$

The EM Algorithm in General (3)

- To verify the relation, first use the product rule

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

- KL(q||p) is the Kullback-Leibler divergence between q(Z) and the posterior distribution p(Z|X, θ)
- The Kullback-Leibler divergence satisfies KL(q||p) >= 0, with equality if and only if q(Z) = p(Z|X, θ) => L(q, θ) is a lower bound on ln p(X|θ)

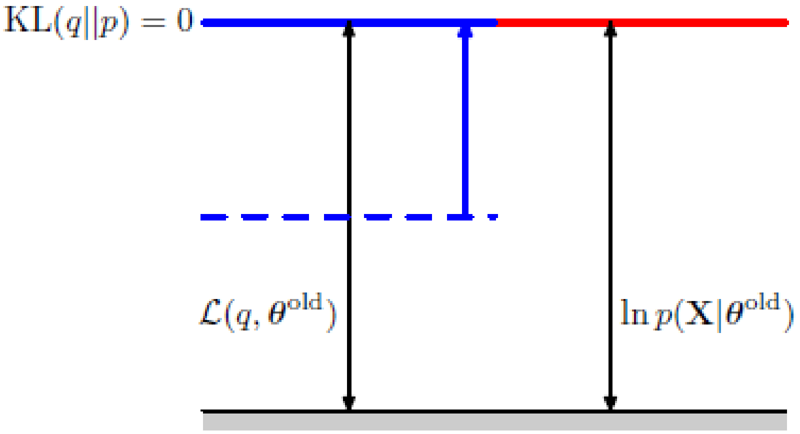

$\mathrm{KL}(q||p)$
$\mathcal{L}(q, \boldsymbol{\theta})$
$\ln p(\mathbf{X}|\boldsymbol{\theta})$

The EM Algorithm in General (4)

- EM algorithm is a two-stage iterative optimization technique for finding maximum likelihood solution
- Suppose that the current value of the parameter vector is θ^{old}
- In the E step, the lower bound $L(q, \theta^{old})$ is maximized with respect to $q(Z)$ while holding θ^{old} fixed
- In the subsequent M step, the distribution $q(Z)$ is held fixed and the lower bound $L(q, \theta)$ is maximized with respect to θ to give some new value θ^{new}
- This will cause the lower bound L to increase (unless it is already at a maximum), which will necessarily cause the corresponding log likelihood function to increase

The EM Algorithm in General (5)

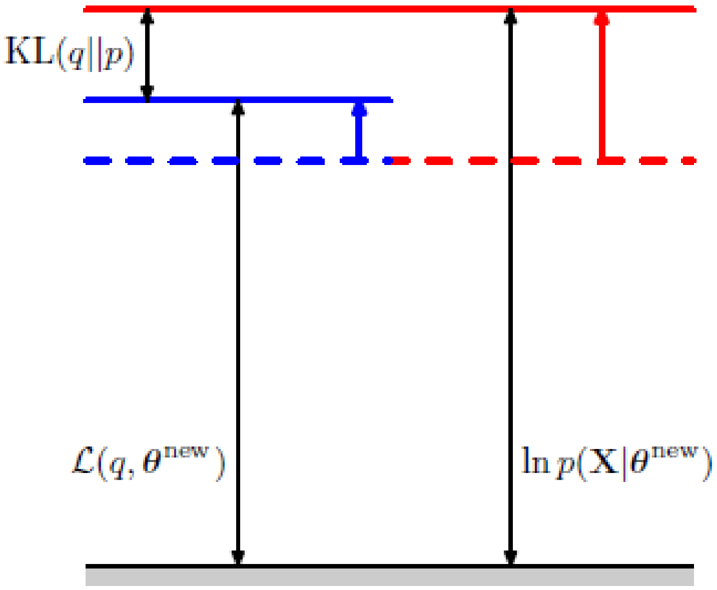
- E step


$\mathrm{KL}(q||p) = 0$
$\mathcal{L}(q, \boldsymbol{\theta}^{\mathrm{old}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\mathrm{old}})$

$$\begin{aligned}\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \ln p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\mathrm{old}}) \\ &= \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{old}}) + \mathrm{const}\end{aligned}$$

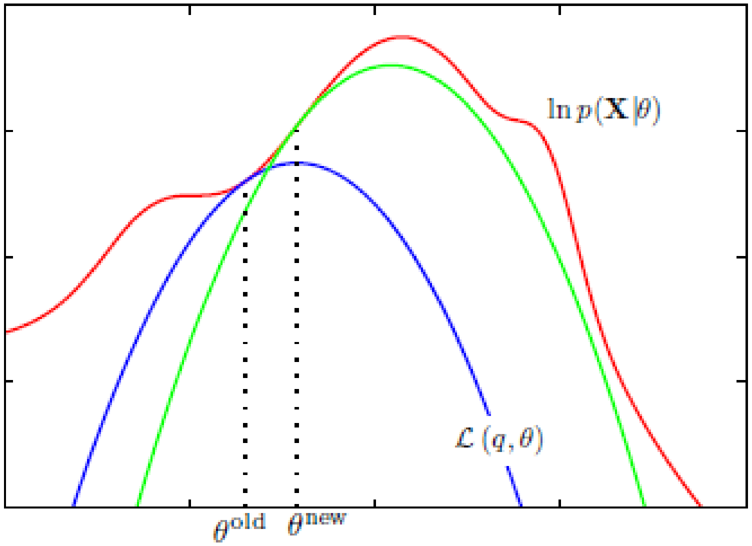
The EM Algorithm in General (6)

- M step



$\mathrm{KL}(q||p)$
$\mathcal{L}(q, \boldsymbol{\theta}^{\text{new}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\text{new}})$

The EM Algorithm in General (7)


$\ln p(\mathbf{X}|\theta)$
$\mathcal{L}(q,\theta)$
θ^{old}
θ^{new}