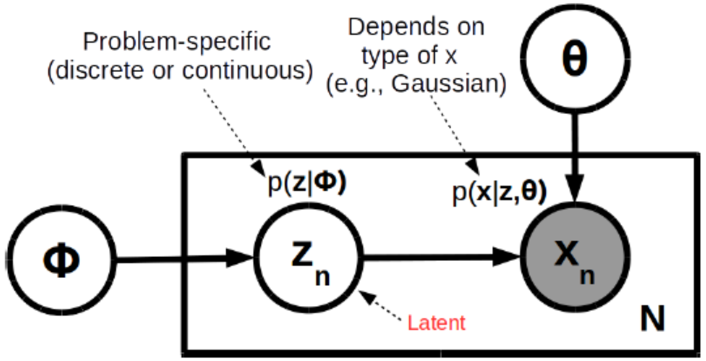
Latent Variable Models

Latent Variable Models



- Variables that cannot be observed (both in training and testing)

Advantages:

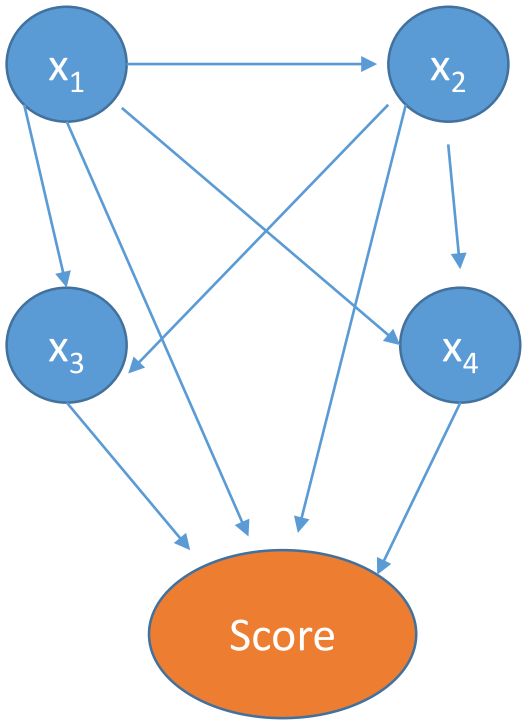
- Augment model to simplify inference (logistic regression)
- Latent features/properties of data (clusters, topics, representation)

Example of Latent Variable

- Team Selection for a Sports Meet

Height (x_1) (m)	Weight (x_2) (kg)	Daily exercise (x_3) (kCal)	Hours of sleep (x_4) (hrs)	Performance Score
1.64	85	2300	8	60
1.83	80	2700	7	90
1.52	70	2200	6	70

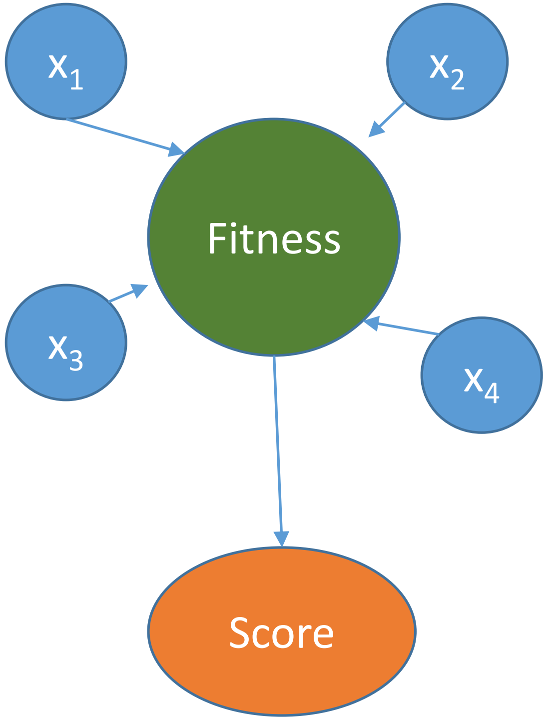
Probabilistic Inference



$p(\text{score}|x_1, x_2, x_3, x_4)$

Large number possible combinations of the variables.

Probabilistic Inference



Fitness – latent variable

$p(\text{score}|\text{fitness})p(\text{fitness}|x_1)p(\text{fitness}|x_2)p(\text{fitness}|x_3)p(\text{fitness}|x_4)$

Reduction in number of model parameters

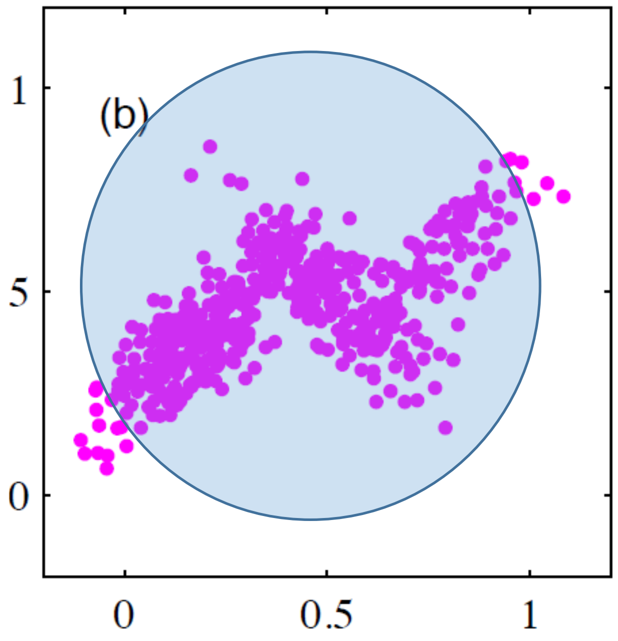
Parameters vs Latent Variables

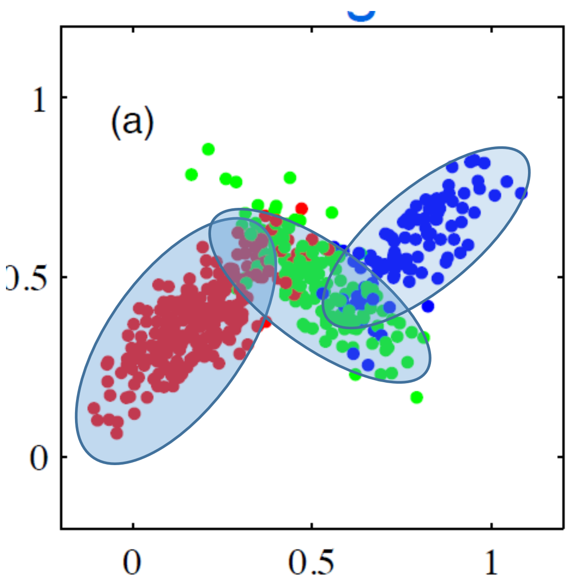
- Parameters are global, Latent Variables are observation specific/local

- Computationally difficult to do posterior inference for all the variables

- Hybrid inference
  - Estimate Posterior for latent/local variable
  - Point estimate (e.g., MLE) for parameters/global variables

Example: Gaussian Mixture Model (GMM)





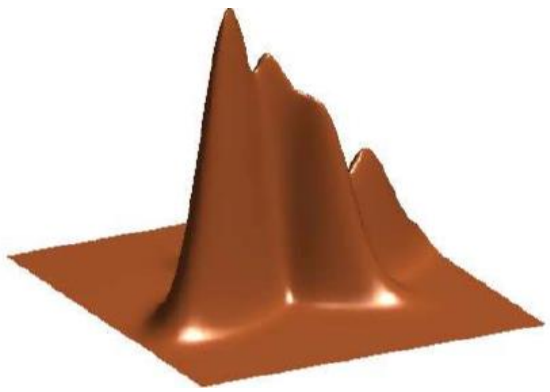
Mixture of Gaussian

- A Gaussian mixture model represents a **distribution** as

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

with π_k the mixing coefficients, where:

$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$



- GMMs are **universal approximators of densities**

Latent Variable View of GMM

- We could introduce a hidden (latent) variable z which would represent which Gaussian generated our observation $\mathbf{x}$, with some probability
- Let $z \sim \mathrm{Categorical}(\boldsymbol{\pi})$ (where $\pi_k \geq 0, \quad \sum_k \pi_k = 1$)
- Then:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}, z = k)$$

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

Parameter Estimation of GMM

- Maximum likelihood maximizes

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

w.r.t $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

- How would you optimize this?
- Can we have a closed form update?
- Don't forget to satisfy the constraints on π_k

Parameter Estimation in GMM

- A Gaussian mixture distribution:

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

- We had: $z \sim \text{Categorical}(\boldsymbol{\pi})$ $\quad$ (where $\pi_k \geq 0, \quad \sum_k \pi_k = 1$)
- Joint distribution: $\qquad p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$
- Log-likelihood:

$$\begin{aligned}
\ell(\boldsymbol{\pi}, \mu, \Sigma) = \ln p(\mathbf{X}|\pi, \mu, \Sigma) &= \sum_{n=1}^{N} \ln p(\mathbf{x}^{(n)}|\pi, \mu, \Sigma) \\
&= \sum_{n=1}^{N} \ln \sum_{z^{(n)}=1}^{K} p(\mathbf{x}^{(n)}|\, z^{(n)}; \mu, \Sigma) p(z^{(n)}|\, \boldsymbol{\pi})
\end{aligned}$$

- Note: We have a hidden variable $z^{(n)}$ for every observation
- General problem: sum inside the log

Learning Parameters

- **If we knew** $z^{(n)}$ for every $x^{(n)}$, the maximum likelihood problem is easy:

$$\ell(\boldsymbol{\pi}, \mu, \Sigma) = \sum_{n=1}^{N} \ln p(x^{(n)}, z^{(n)} | \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln p(\mathbf{x}^{(n)} | z^{(n)}; \mu, \Sigma) + \ln p(z^{(n)} | \boldsymbol{\pi})$$

$$\begin{aligned}
\mu_k &= \frac{\sum_{n=1}^{N} 1_{[z^{(n)}=k]} \, \mathbf{x}^{(n)}}{\sum_{n=1}^{N} 1_{[z^{(n)}=k]}} \\
\Sigma_k &= \frac{\sum_{n=1}^{N} 1_{[z^{(n)}=k]} \, (\mathbf{x}^{(n)} - \mu_k)(\mathbf{x}^{(n)} - \mu_k)^T}{\sum_{n=1}^{N} 1_{[z^{(n)}=k]}} \\
\pi_k &= \frac{1}{N} \sum_{n=1}^{N} 1_{[z^{(n)}=k]}
\end{aligned}$$

Learning Parameters

- Similarly if we knew the parameters π, μ, Σ
  - Estimating the latent variable is easy
- Chicken and Egg Problem!

Expectation Maximization Algorithm

- Optimization uses the Expectation Maximization algorithm, which alternates between two steps:
    1. E-step: Compute the posterior probability that each Gaussian generates each datapoint (as this is unknown to us)
    2. M-step: Assuming that the data really was generated this way, change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for.

EM Algorithm

- Elegant and powerful method for finding maximum likelihood solutions for models with latent variables

1. E-step:
   - In order to adjust the parameters, we must first solve the inference problem: Which Gaussian generated each datapoint?
   - We cannot be sure, so it's a distribution over all possibilities.

$$\gamma_k^{(n)} = p(z^{(n)} = k|\mathbf{x}^{(n)}; \pi, \mu, \Sigma)$$

2. M-step:
   - Each Gaussian gets a certain amount of posterior probability for each datapoint.
   - At the optimum we shall satisfy

$$\frac{\partial \ln p(\mathbf{X}|\pi, \mu, \Sigma)}{\partial \Theta} = 0$$

   - We can derive closed form updates for all parameters

E Step

- Conditional probability (using Bayes rule) of $\mathbf{z}$ given $\mathbf{x}$

$$\begin{aligned}
\gamma_k = p(z = k|\mathbf{x}) &= \frac{p(z = k)p(\mathbf{x}|z = k)}{p(\mathbf{x})} \\
&= \frac{p(z = k)p(\mathbf{x}|z = k)}{\sum_{j=1}^{K} p(z = j)p(\mathbf{x}|z = j)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}
\end{aligned}$$

M Step

- Log-likelihood:

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

- Set derivatives to 0:

$$\frac{\partial \ln p(\mathbf{X}|\pi, \mu, \Sigma)}{\partial \mu_k} = 0 = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)} \Sigma_k(\mathbf{x}^{(n)} - \mu_k)$$

- We used:

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

and:

$$\frac{\partial(\mathbf{x}^T A \mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}^T (A + A^T)$$

M Step

$$\frac{\partial \ln p(\mathbf{X}|\pi,\mu,\Sigma)}{\partial \mu_k} = 0 = \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j,\Sigma_j)}}_{\gamma_k^{(n)}} \Sigma_k(\mathbf{x}^{(n)} - \mu_k)$$

- This gives

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)}$$

with N_k the effective number of points in cluster k

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

M Step

- We can get similarly expression for the variance

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\mathbf{x}^{(n)} - \mu_k)(\mathbf{x}^{(n)} - \mu_k)^T$$

- We can also minimize w.r.t the mixing coefficients

$$\pi_k = \frac{N_k}{N}, \quad \text{with} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)}$$

- The optimal mixing proportion to use (given these posterior probabilities) is just the fraction of the data that the Gaussian gets responsibility for.
- Note that this is not a closed form solution of the parameters, as they depend on the responsibilities $\gamma_k^{(n)}$, which are complex functions of the parameters
- But we have a simple iterative scheme to optimize

Summary of GMM

- Initialize the means μ_k, covariances Σ_k and mixing coefficients π_k
- Iterate until convergence:
  - E-step: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = p(z^{(n)}|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \Sigma_j)}$$

  - M-step: Re-estimate the parameters given current responsibilities

$$\begin{aligned}
\mu_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)} \\
\Sigma_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\mathbf{x}^{(n)} - \mu_k)(\mathbf{x}^{(n)} - \mu_k)^T \\
\pi_k &= \frac{N_k}{N} \quad \text{with} \quad N_k = \sum_{n=1}^{N} \gamma_k^{(n)}
\end{aligned}$$

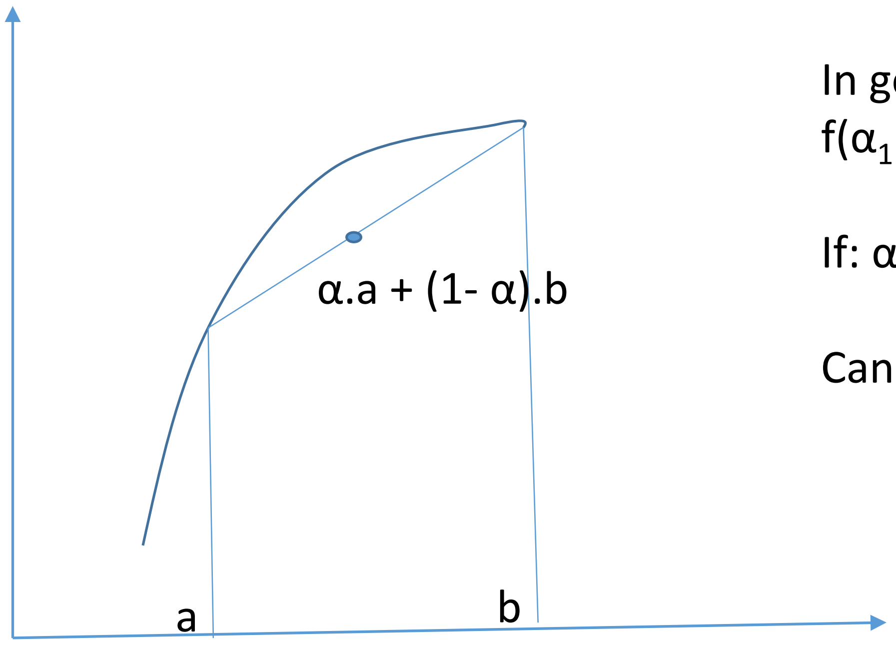
  - Evaluate log likelihood and check for convergence

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

Generalized Expectation Maximization

Generalized EM Algorithm

$f(\alpha.a + (1-\alpha).b) \geq \alpha.f(a) + (1-\alpha).f(b)$



Concave function

In general:

$f(\alpha_1.a_1 + \alpha_2.a_2 + \alpha_3.a_3) \geq \alpha_1.f(a_1) + \alpha_2.f(a_2) + \alpha_3.f(a_3)$

If: $\alpha_1 + \alpha_2 + \alpha_3 = 1$

Can think of α's as probabilities.

Logarithm is an example of concave function

Jensen Inequality

Theorem. Let f be a convex function, and let X be a random variable. Then:

$$\mathrm{E}[f(X)] \geq f(\mathrm{E}X).$$

Reverse holds for concave functions.

If f is concave, -f is convex

Kullback-Leibler Divergence

- Measures similarity between two distributions

$$D_{KL}(p||q) = \int_x p(x) log \frac{p(x)}{q(x)} dx$$

- The value is greater than or equal to zero.
- The value is zero when two distributions are identical.

MLE in LVM

- Suppose we want to estimate parameters Θ via MLE. If we knew both $\boldsymbol{x}_n$ and $\boldsymbol{z}_n$ then we could do

$$\Theta_{MLE} \quad = \quad \arg\max_{\Theta} \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) = \arg\max_{\Theta} \sum_{n=1}^{N} [\log p(\boldsymbol{z}_n|\phi) + \log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)]$$

- Simple to solve (usually closed form) if $p(\boldsymbol{z}_n|\phi)$ and $p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)$ are "simple" (e.g., exp-fam. dist.)
- However, in LVMs where $\boldsymbol{z}_n$ is "hidden", the MLE problem will be the following

$$\Theta_{MLE} \quad = \quad \arg\max_{\Theta} \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta) = \arg\max_{\Theta} \log p(\mathbf{X}|\Theta)$$

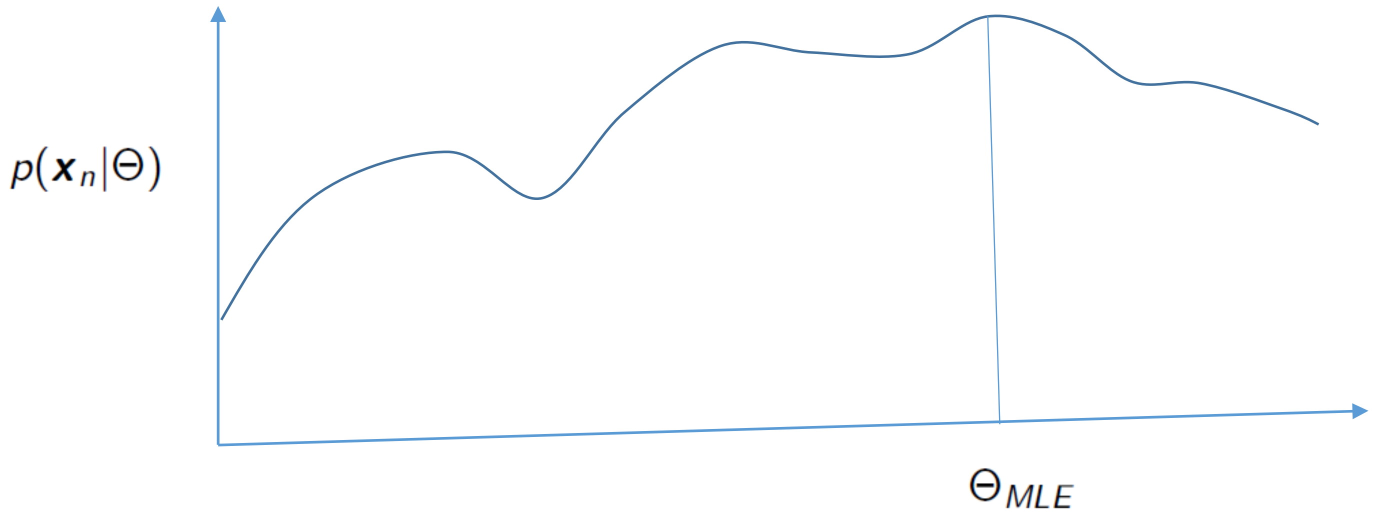
- The form of $p(\boldsymbol{x}_n|\Theta)$ may not be simple since we need to sum over unknown $\boldsymbol{z}_n$'s possible values

$$p(\boldsymbol{x}_n|\Theta) = \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$$

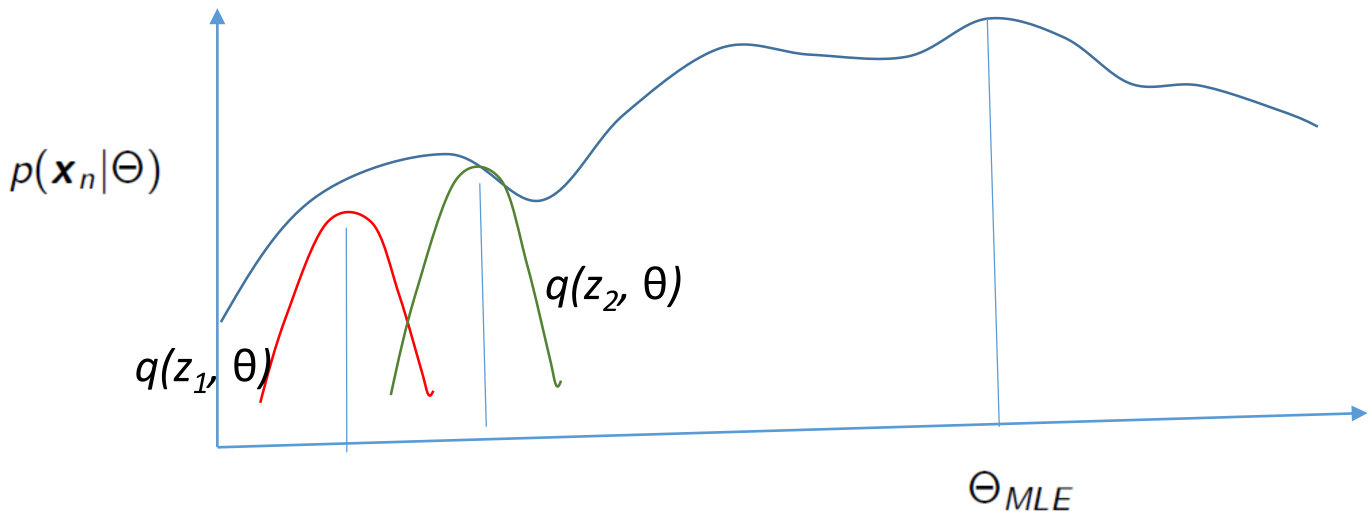
or if $\boldsymbol{z}_n$ is continuous: $p(\boldsymbol{x}_n|\Theta) = \int p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta) d\boldsymbol{z}_n$

MLE in LVM: Optimization Problem

- The summation/integral may lead to complex expressions for the likelihood



Optimizing a Lower Bound



$p(\boldsymbol{x}_n|\Theta) \geq q(z, \theta)$

q – variational distribution, changes with z

Depends on both latent variable and parameter
Easy to maximise

Two Step Iterative Optimization (for MLE)

- Step 1: Obtain the variational distribution lower bound with lowest gap

- Step 2: Obtain the maximum point for that variational distribution as candidate solution for θ_{MLE}

- Repeat

Lower Bound on the Likelihood

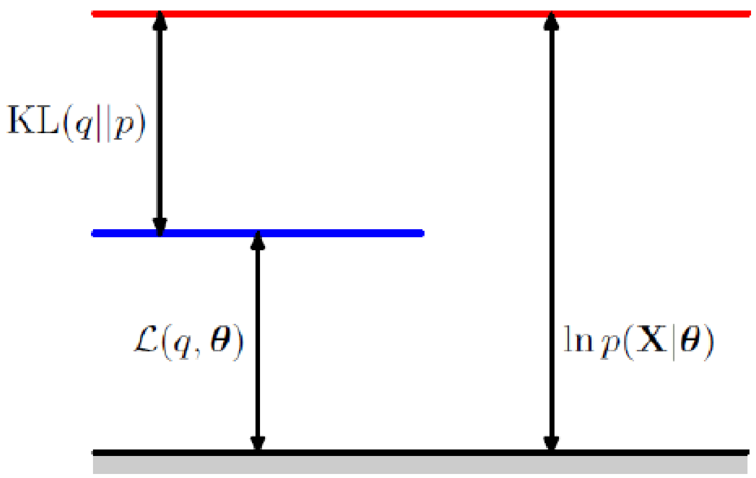
- Define $p_z = p(\mathbf{Z}|\mathbf{X}, \Theta)$ and let $q(\mathbf{Z})$ be some distribution over $\mathbf{Z}$
- Assume discrete $\mathbf{Z}$, the identity below holds for any choice of the distribution $q(\mathbf{Z})$

$$\boxed{\log p(\mathbf{X}|\Theta) = \mathcal{L}(q, \Theta) + \mathrm{KL}(q||p_z)}$$

$$\mathcal{L}(q, \Theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\Theta)}{q(\mathbf{Z})} \right\}$$

$$\mathrm{KL}(q||p_z) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \Theta)}{q(\mathbf{Z})} \right\}$$

(Exercise: Verify the above identity)



- Since $\mathrm{KL}(q||p_z) \geq 0$, $\mathcal{L}(q, \Theta)$ is a lower-bound on $\log p(\mathbf{X}|\Theta)$

$$\log p(\mathbf{X}|\Theta) \geq \mathcal{L}(q, \Theta)$$

- Maximizing $\mathcal{L}(q, \Theta)$ will also improve $\log p(\mathbf{X}|\Theta)$

Maximising $\mathcal{L}$

- First recall the identity we had: $\log p(\mathbf{X}|\Theta) = \mathcal{L}(q,\Theta) + \mathrm{KL}(q||p_z)$ with

$$\mathcal{L}(q,\Theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left\{\frac{p(\mathbf{X},\mathbf{Z}|\Theta)}{q(\mathbf{Z})}\right\} \quad \text{and} \quad \mathrm{KL}(q||p_z) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left\{\frac{p(\mathbf{Z}|\mathbf{X},\Theta)}{q(\mathbf{Z})}\right\}$$

- Maximize $\mathcal{L}$ w.r.t. q with Θ fixed at Θ^{old}: Since $\log p(\mathbf{X}|\Theta)$ will be a constant in this case,

$$\hat{q} = \arg\max_{q} \mathcal{L}(q,\Theta^{old}) = \arg\min_{q} \mathrm{KL}(q||p_z) = p_z = p(\mathbf{Z}|\mathbf{X},\Theta^{old})$$

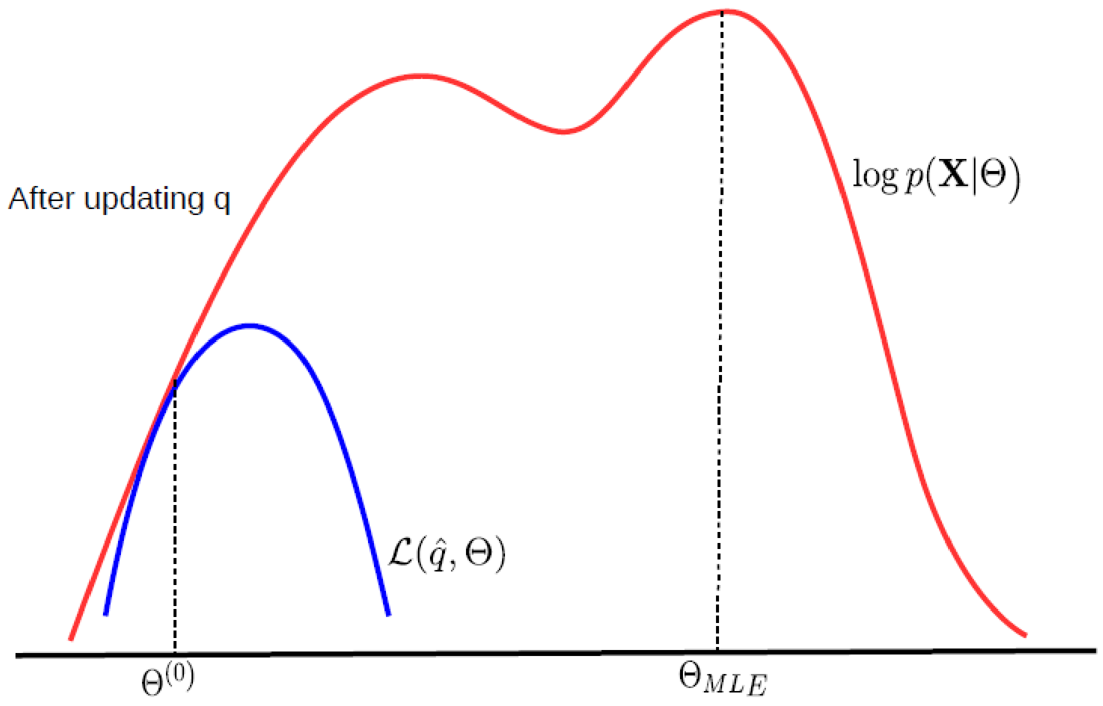
- Maximize $\mathcal{L}$ w.r.t. Θ with q fixed at $\hat{q} = p(\mathbf{Z}|\mathbf{X},\Theta^{old})$

$$\Theta^{new} = \arg\max_{\Theta} \mathcal{L}(\hat{q},\Theta) = \arg\max_{\Theta} \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\Theta^{old}) \log \frac{p(\mathbf{X},\mathbf{Z}|\Theta)}{p(\mathbf{Z}|\mathbf{X},\Theta^{old})} = \arg\max_{\Theta} \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\Theta^{old}) \log p(\mathbf{X},\mathbf{Z}|\Theta)$$

.. therefore, $\boxed{\Theta^{new} = \arg\max_{\theta} \mathcal{Q}(\Theta,\Theta^{old})}$ where $\mathcal{Q}(\Theta,\Theta^{old}) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\Theta^{old})}[\log p(\mathbf{X},\mathbf{Z}|\Theta)]$

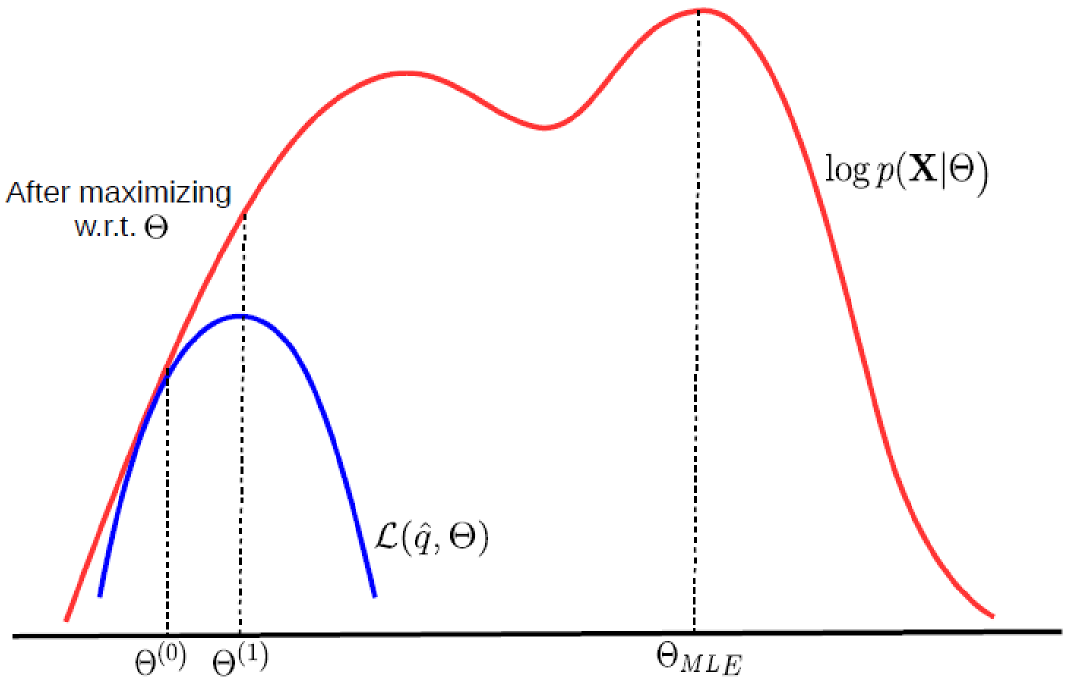
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



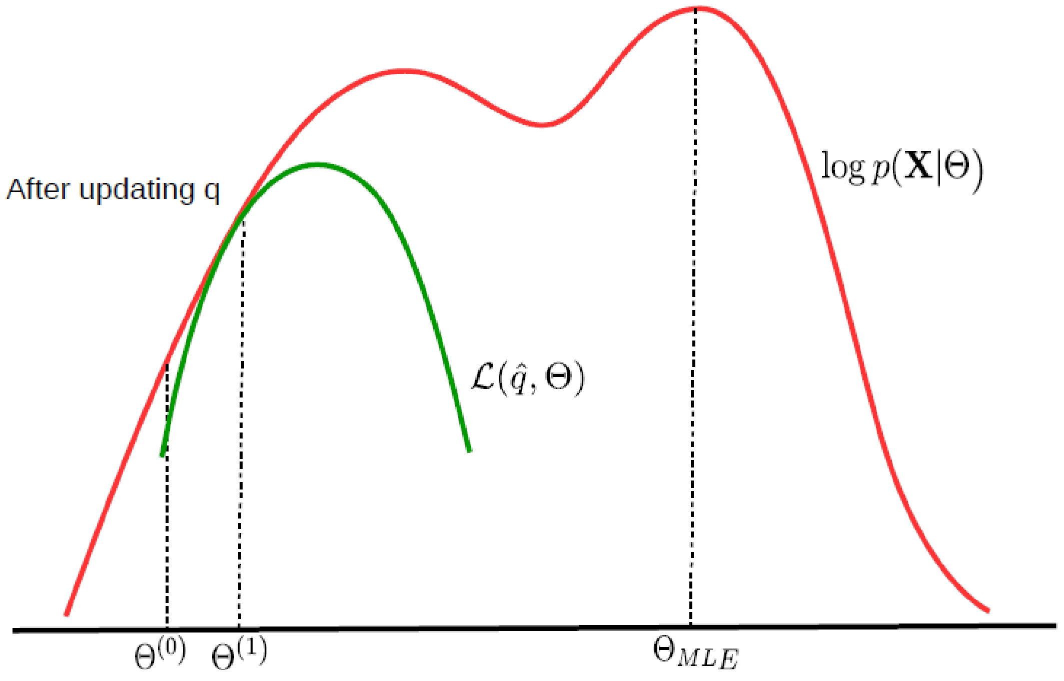
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



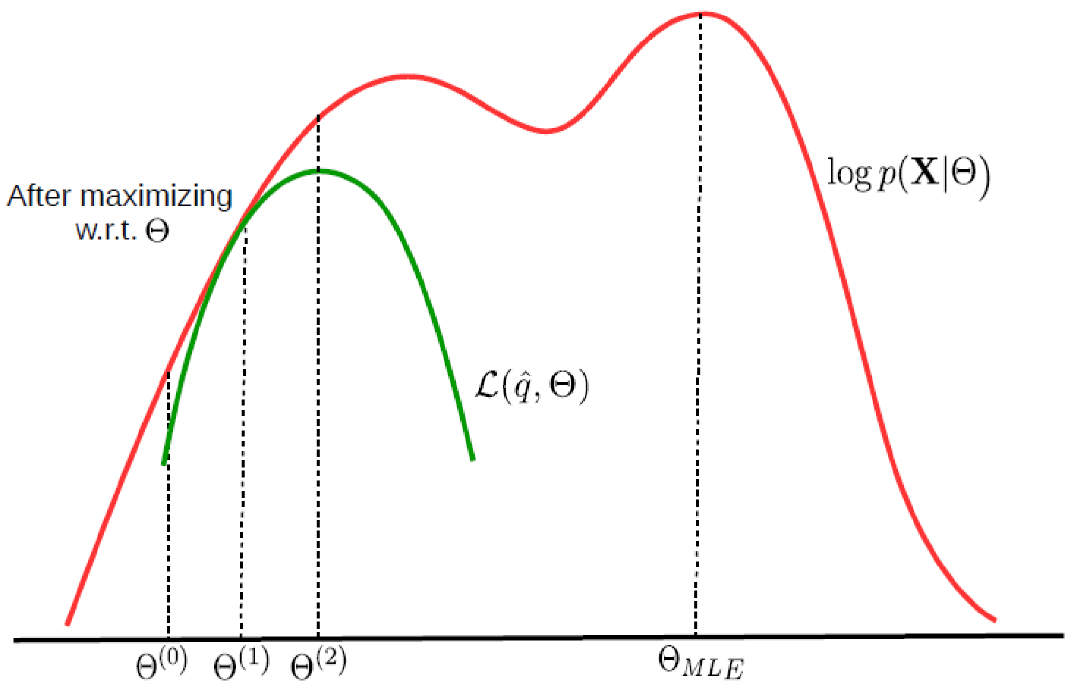
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



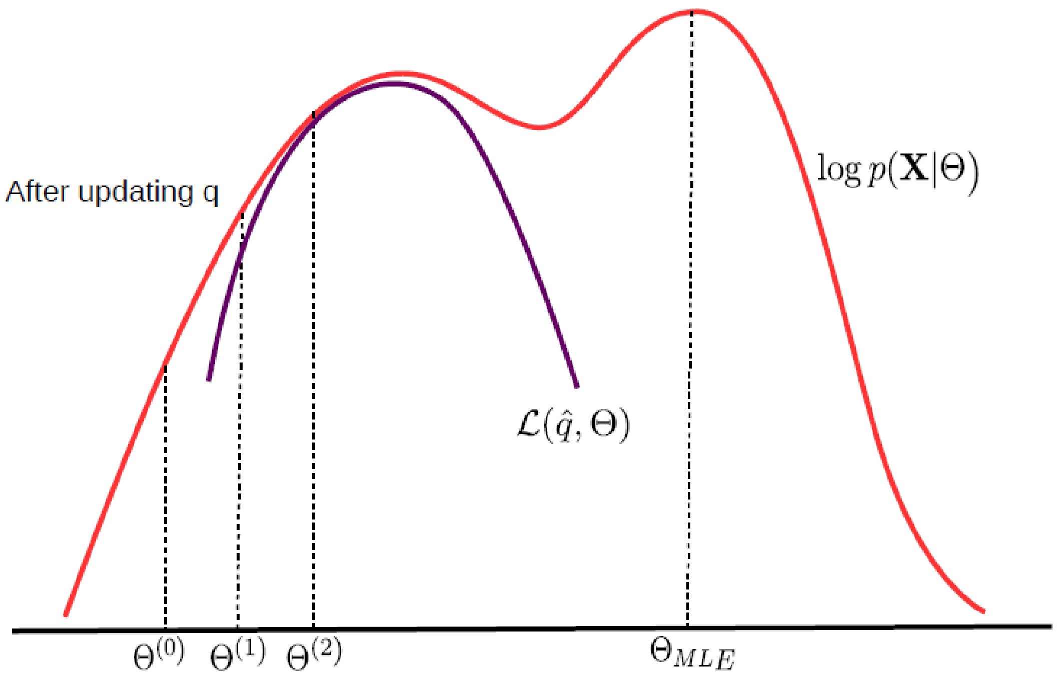
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



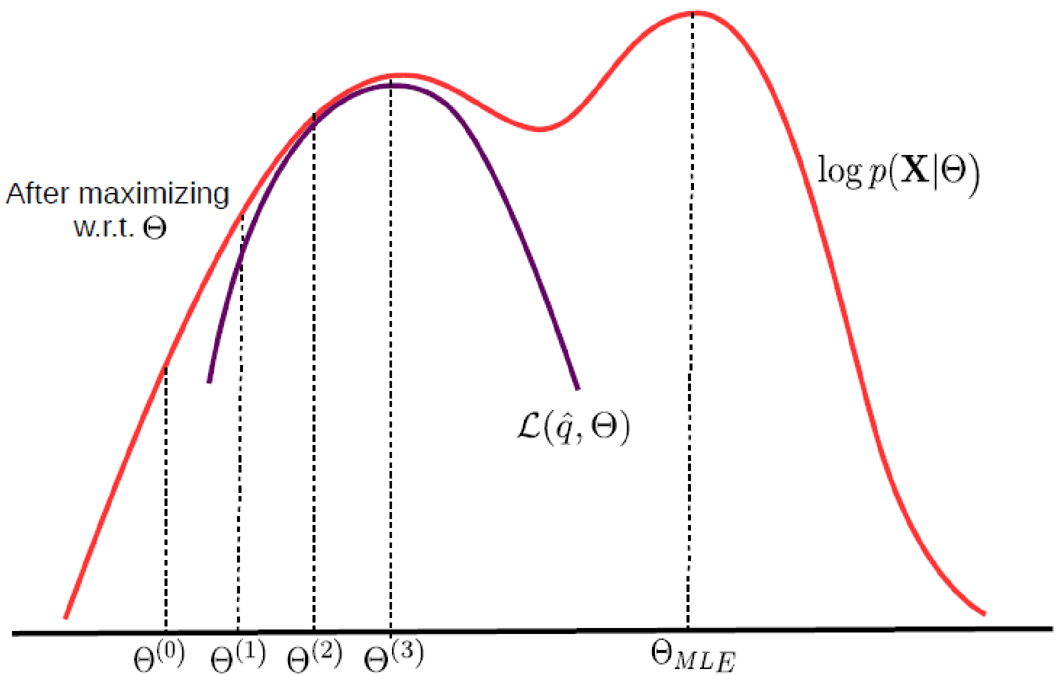
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



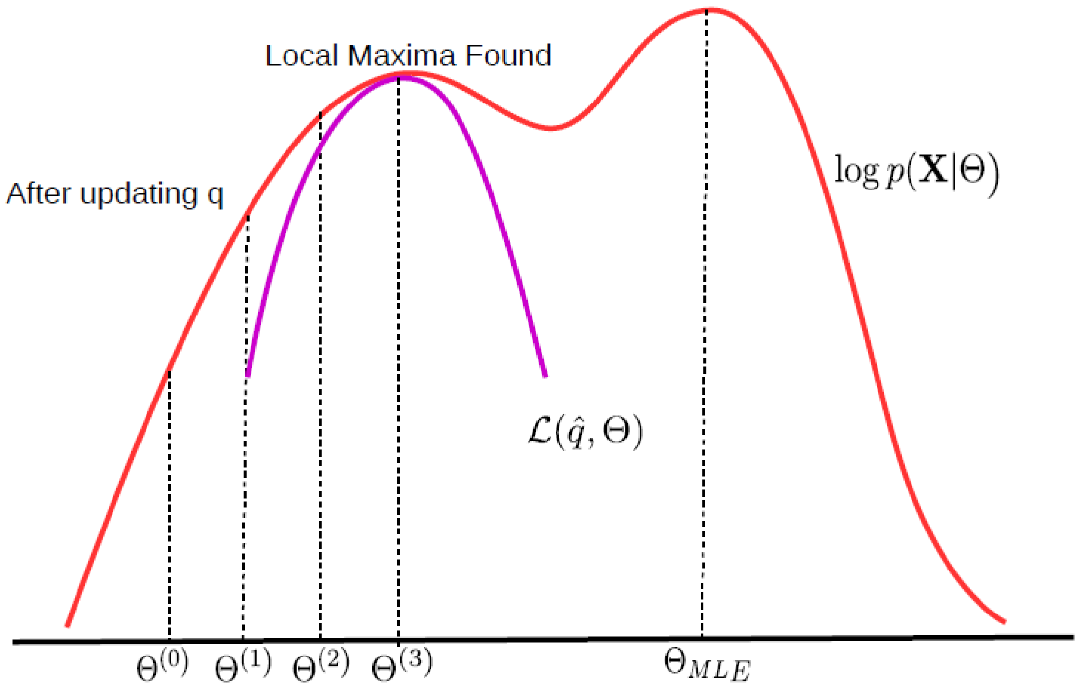
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



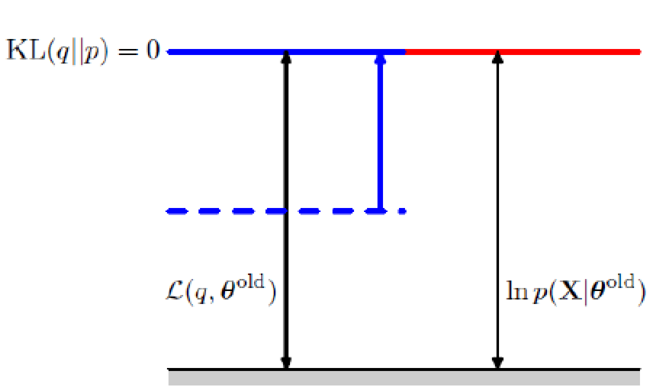
Visualization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at Θ^{old}
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. Θ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)

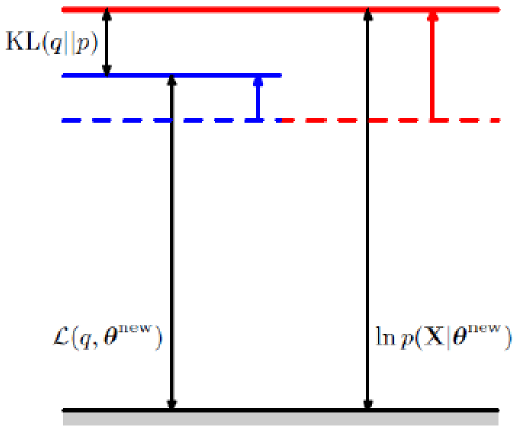


Monotonicity

- The two-step alternating optimzation scheme we saw can never decrease $p(\mathbf{X}|\Theta)$ (good thing)
- To see this consider both steps: (1) Optimize q given $\Theta = \Theta^{old}$; (2) Optimize Θ given this q



(Step 1)



(Step 2)

- Step 1 keeps Θ fixed, so $p(\mathbf{X}|\Theta)$ obviously can't decrease (stays unchanged in this step)
- Step 2 maximizes the lower bound $\mathcal{L}(q, \Theta)$ w.r.t Θ. Thus $p(\mathbf{X}|\Theta)$ can't decrease!

The EM Algorithm

Initialize the parameters: Θ^{old}. Then alternate between these steps:

- **E (Expectation) step:**
  - Compute the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$ over latent variables $\mathbf{Z}$ using Θ^{old}
  - Compute the expected complete data log-likelihood w.r.t. *this* posterior distribution

$$
\begin{aligned}
\mathcal{Q}(\Theta, \Theta^{old}) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\Theta^{old})}[\log p(\mathbf{X}, \mathbf{Z}|\Theta)] = \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\Theta^{old})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)] \\
&= \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\Theta^{old})}[\log p(\boldsymbol{x}_n|\boldsymbol{z}_n, \Theta) + \log p(\boldsymbol{z}_n|\Theta)]
\end{aligned}
$$

- **M (Maximization) step:**
  - Maximize the expected complete data log-likelihood w.r.t. Θ

$$
\Theta^{new} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{old})
$$

Continue till log-likelihood does not converge!

Pseudocode

The EM Algorithm

- Initialize Θ as $\Theta^{(0)}$, set $t = 1$
- Step 1: Compute conditional posterior of latent vars given current params $\Theta^{(t-1)}$

$$p(\boldsymbol{z}_n^{(t)}|\boldsymbol{x}_n, \Theta^{(t-1)}) = \frac{p(\boldsymbol{z}_n^{(t)}|\Theta^{(t-1)})p(\boldsymbol{x}_n|\boldsymbol{z}_n^{(t)}, \Theta^{(t-1)})}{p(\boldsymbol{x}_n|\Theta^{(t-1)})} \propto \text{prior} \times \text{likelihood}$$

- Step 2: Now maximize the expected complete data log-likelihood w.r.t. Θ

$$\Theta^{(t)} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{(t-1)}) = \arg\max_{\Theta} \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n^{(t)}|\boldsymbol{x}_n, \Theta^{(t-1)})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n^{(t)}|\Theta)]$$

- If not yet converged, set $t = t + 1$ and go to Step 1.

Applications of EM

- Mixture of (multivariate) Gaussians/Bernoullis,multinoullis, Mixture of experts models
- Problems with missing labels/features (treat these as latent variables)
- Note that EM not only gives estimates of the parameters Θ but also infers latent variables $\mathbf{Z}$
- Hyperparameter estimation in probabilistic models (an alternative to MLE-II)
  - We've already seen MLE-II where we did MLE on marginal likelihood, e.g., for linear regression
  $$p(\boldsymbol{y}|\mathbf{X}, \lambda, \beta) = \int p(\boldsymbol{y}|\mathbf{X}, \boldsymbol{w}, \beta)p(\boldsymbol{w}|\lambda)d\boldsymbol{w}$$
  - As an alternative, can treat $\boldsymbol{w}$ as a latent variable and β, λ as parameters and use EM to learn these