UVA CS 4501: Machine Learning

Lecture 24-Extra : EM (Extra)

Dr. Yanjun Qi

University of Virginia
Department of Computer Science
Where are we?

major sections of this course

- Regression (supervised)
- Classification (supervised)
  - Feature selection
- Unsupervised models
  - Dimension Reduction (PCA)
  - Clustering (K-means, GMM/EM, Hierarchical)
- Learning theory
- Graphical models
  - (BN and HMM slides shared)
Today Outline

• Principles for Model Inference
  – Maximum Likelihood Estimation
  – Bayesian Estimation

• Strategies for Model Inference
  – EM Algorithm – simplify difficult MLE
    • Algorithm
    • Application
    • Theory
  – MCMC – samples rather than maximizing
Model Inference through Maximum Likelihood Estimation (MLE)

Assumption: the data is coming from a known probability distribution

The probability distribution has some parameters that are unknown to you

Example: data is distributed as Gaussian $y_i = \mathcal{N}(\mu, \sigma^2)$, so the unknown parameters here are $\theta = (\mu, \sigma^2)$

MLE is a tool that estimates the unknown parameters of the probability distribution from data
MLE: e.g. Single Gaussian Model (when $p=1$)

- Need to adjust the parameters ($\Rightarrow$ model inference)
- So that the resulting distribution fits the observed data well
Maximum Likelihood revisited

\[ y_i = N(\mu, \sigma^2) \]

\[ Y = \{y_1, y_2, \ldots, y_N\} \]

\[ l(\theta) = \log(L(\theta; Y)) = \log \prod_{i=1}^{N} p(y_i) \]

Choose \( \theta \) that maximizes \( l(\theta) \)

\[ \frac{\partial l}{\partial \theta} = 0 \]
MLE: e.g. Single Gaussian Model

- Assume observation data \( y_i \) are independent

- Form the Likelihood:

\[
L(\theta; Y) = \prod_{i=1}^{N} p(y_i) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right);
\]

\( Y = \{y_1, y_2, \ldots, y_N\} \)

- Form the Log-likelihood:

\[
l(\theta) = \log\left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right)\right) = -\sum_{i=1}^{N} \frac{(y_i - \mu)^2}{2\sigma^2} - N \log(\sqrt{2\pi\sigma})
\]

4/30/18
MLE: e.g. Single Gaussian Model

- To find out the unknown parameter values, maximize the log-likelihood with respect to the unknown parameters:

\[ \frac{\partial l}{\partial \theta} = 0 \]

Choose \( \theta \) that maximizes \( l(\theta) \) . . .

\[ \frac{\partial l}{\partial \mu} = 0 \Rightarrow \mu = \frac{\sum_{i=1}^{N} y_i}{N} ; \quad \frac{\partial l}{\partial \sigma^2} = 0 \Rightarrow \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mu)^2 \]
MLE: A Challenging Mixture Example

\[ Y_1 \sim N(\mu_1, \sigma_1^2); \quad Y_2 \sim N(\mu_2, \sigma_2^2) \]
\[ Y = (1 - \Delta)Y_1 + \Delta Y_2; \quad \Delta \in \{0,1\} \]

\( g_Y(y) = (1 - \pi)\Phi_{\theta_1}(y) + \pi \Phi_{\theta_2}(y) \)

\( \theta_1 = (\mu_1, \sigma_1); \quad \theta_2 = (\mu_2, \sigma_2) \)

\( \pi \) is the probability with which the observation is chosen from density model 2

\( (1 - \pi) \) is the probability with which the observation is chosen from density 1
MLE: Gaussian Mixture Example

\[ g_Y(y) = (1 - \pi) \Phi_{\theta_1}(y) + \pi \Phi_{\theta_2}(y) \]

Maximum likelihood fitting for parameters: \( \hat{\theta} = (\pi, \mu_1, \mu_2, \sigma_1, \sigma_2) \)

\[ l(\theta) = \sum_{i=1}^{N} \log[(1 - \pi) \Phi_{\theta_1}(y_i) + \pi \Phi_{\theta_2}(y_i)] \]

\[ \frac{\partial l}{\partial \theta} = 0 \]

Numerically (and of course analytically, too)
Challenging to solve!!
Bayesian Methods & Maximum Likelihood

- Bayesian
  \[ \text{Pr}(\text{model}|\text{data}) \text{ i.e. posterior} \]
  \[ = \text{Pr}(\text{data}|\text{model}) \text{ Pr}(\text{model}) \]
  \[ = \text{Likelihood} \ast \text{prior} \]

- Assume prior is uniform, equal to MLE
  \[ \text{argmax}_{\text{model}} \text{ Pr}(\text{data} | \text{model}) \text{ Pr}(\text{model}) \]
  \[ = \text{argmax}_{\text{model}} \text{ Pr}(\text{data} | \text{model}) \]
Today Outline

• Principles for Model Inference
  – Maximum Likelihood Estimation
  – Bayesian Estimation

• Strategies for Model Inference
  – EM Algorithm – simplify difficult MLE
    • Algorithm
    • Application
    • Theory
  – MCMC – samples rather than maximizing
Here is the problem

\[ M_1, \sigma_1 \quad M_2, \sigma_2 \]

Source Unknown

\[ 1 - \pi \quad \pi \]
All we have is

From which we need to infer the likelihood function which generate the observations
Expectation Maximization: add latent variable $\Delta \rightarrow$ latent data $\Delta_i$

$EM$ **augments the data space**—**assumes with latent data**

$\Delta_i \in \{0, 1\}$ (latent data)

if($\Delta_i = 0$)

$y_i$ was generated from first component

if($\Delta_i = 1$)

$y_i$ was generated from second component

**Complete data:** $t_i = (y_i, \Delta_i)$

$p(t_i|\theta) = p(y_i, \Delta_i|\theta) = p(y_i|\Delta_i, \theta)Pr(\Delta_i)$

$p(t_i|\theta) = [\Phi_{\theta_1}(y_i)(1 - \pi)]^{(1-\Delta_i)}[\Phi_{\theta_2}(y_i)\pi]^{\Delta_i}$
Computing \textbf{log-likelihood} based on complete data

\[ p(t_i|\theta) = [\Phi_{\theta_1}(y_i)(1-\pi)]^{(1-\Delta_i)}[\pi \Phi_{\theta_2}(y_i)\pi]^{\Delta_i} \]

\[ l_0(\theta; T) = \sum_{i=1}^{N} (1-\Delta_i)\log[(1-\pi)\Phi_{\theta_1}(y_i)] + \Delta_i\log[\pi \Phi_{\theta_2}(y_i)] \]

\[ = \sum_{i=1}^{N} (1 - \Delta_i)\log\Phi_{\theta_1}(y_i) + \Delta_i\log\Phi_{\theta_2}(y_i) \]

\[ + \sum_{i=1}^{N} [(1 - \Delta_i)\log(1 - \pi) + \Delta_i\log\pi]\]  \hspace{1cm} (8.40)

Maximizing this form of log-likelihood is now \textit{tractable}

Note that we \textbf{cannot} analytically maximize the previous log-likelihood with only observed \( Y = \{y_1, y_2, \ldots, y_n\} \)
EM: The Complete Data Likelihood

By simple differentiations we have:

\[
\frac{\partial l_0}{\partial \mu_1} = 0 \Rightarrow \mu_1 = \frac{\sum_{i=1}^{N} (1-\Delta_i)y_i}{\sum_{i=1}^{N} (1-\Delta_i)}; \\
\frac{\partial l_0}{\partial \sigma_1^2} = 0 \Rightarrow \sigma_1^2 = \frac{\sum_{i=1}^{N} (1-\Delta_i)(y_i - \mu_1)^2}{\sum_{i=1}^{N} (1-\Delta_i)}; 
\]

So, maximization of the complete data likelihood is much easier!

How do we get the latent variables?
EM: The Complete Data Likelihood

By simple differentiations we have:

\[
\frac{\partial l_0}{\partial \mu_2} = 0 \Rightarrow \mu_2 = \frac{\sum_{i=1}^{N} \Delta_i y_i}{\sum_{i=1}^{N} \Delta_i} ;
\]

\[
\frac{\partial l_0}{\partial \sigma_2^2} = 0 \Rightarrow \sigma_2^2 = \frac{\sum_{i=1}^{N} \Delta_i (y_i - \mu_2)^2}{\sum_{i=1}^{N} \Delta_i} ;
\]

\[
\frac{\partial l_0}{\partial \pi} = 0 \Rightarrow \pi = \frac{\sum_{i=1}^{N} \Delta_i}{N} ;
\]

So, maximization of the complete data likelihood is much easier!

How do we get the latent variables?
Obtaining Latent Variables

The latent variables are computed as expected values given the data and parameters:

\[ \gamma_i(\theta) = E(\Delta_i \mid \theta, y_i) = \Pr(\Delta_i = 1 \mid \theta, y_i) \]

Apply Bayes’ rule:

\[
\gamma_i(\theta) = \Pr(\Delta_i = 1 \mid \theta, y_i) = \frac{\Pr(y_i \mid \Delta_i = 1, \theta) \Pr(\Delta_i = 1 \mid \theta)}{\Pr(y_i \mid \Delta_i = 1, \theta) \Pr(\Delta_i = 1 \mid \theta) + \Pr(y_i \mid \Delta_i = 0, \theta) \Pr(\Delta_i = 0 \mid \theta)} = \frac{\Phi_{\theta_2}(y_i)\pi}{\Phi_{\theta_1}(y_i)(1-\pi) + \Phi_{\theta_2}(y_i)\pi}
\]

\[(y_{\hat{i}}, \theta^{(t)}) \rightarrow E(\Delta_i)^{(t)}\]
Dilemma Situation

- We need to know latent variable / data to maximize the complete log-likelihood to get the parameters.

- We need to know the parameters to calculate the expected values of latent variable / data.

- ➔ Solve through iterations.
So we iterate \( \Rightarrow \) EM for Gaussian Mixtures...

1. Initialize parameters \( \hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi} \)

2. Expectation Step:

\[
\gamma_i(\theta) = E(\Delta_i|\theta,Y) = Pr(\Delta_i = 1|\theta,Y)
\]

By Bayes' theorem:

\[
Pr(\Delta_i = 1|\theta, y_i) = \frac{p(y_i|\Delta_i = 1, \theta).Pr(\Delta_i = 1|\theta)}{p(y_i|\theta)}
\]

\[
= \frac{\Phi_{\hat{\theta}_2}(y_i).\hat{\pi}}{(1-\hat{\pi})\Phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\Phi_{\hat{\theta}_2}(y_i)}
\]

\[
E[l_0(\theta; T|Y, \hat{\theta}(j))] = \sum_{i=1}^{N} [(1 - \hat{\gamma}_i)log\Phi_{\hat{\theta}_1}(y_i) + \hat{\gamma}_i log\Phi_{\hat{\theta}_2}(y_i)]
+ \sum_{i=1}^{N} [(1 - \hat{\gamma}_i)log(1 - \pi) + \hat{\gamma}_i log\pi]
\]
3. Maximization Step:

$$Q(\theta', \hat{\theta}(j)) = E[l_0(\theta'; T|Y, \hat{\theta}(j))]$$

$$= \sum_{i=1}^{N} [(1 - \hat{\gamma}_i) \log \Phi_{\theta_1}(y_i) + \hat{\gamma}_i \log \Phi_{\theta_2}(y_i)]$$

$$+ \sum_{i=1}^{N} [(1 - \hat{\gamma}_i) \log (1 - \pi) + \hat{\gamma}_i \log \pi]$$

Find $\theta'$ that maximizes $Q(\theta', \hat{\theta}(j))$ …

Set \[ \frac{\partial Q}{\partial \mu_1}, \frac{\partial Q}{\partial \mu_2}, \frac{\partial Q}{\partial \sigma_1}, \frac{\partial Q}{\partial \sigma_2}, \frac{\partial Q}{\partial \pi} = 0 \]

to get $\hat{\theta}(j+1)$

4. Use this $\hat{\theta}^{j+1}$ to compute the expected values $\hat{\gamma}_i$ and repeat…until convergence
EM for Two-component Gaussian Mixture

- Initialize $\mu_1, \sigma_1, \mu_2, \sigma_2, \pi$
- Iterate until convergence
  - **Expectation** of latent variables
    $$\gamma_i(\theta) = \frac{\Phi_{\theta_2}(y_i)\pi}{\Phi_{\theta_1}(y_i)(1-\pi) + \Phi_{\theta_2}(y_i)\pi} = \frac{1}{1 + \frac{1-\pi}{\pi} \frac{\sigma_2}{\sigma_1} \exp\left(-\frac{(y_i - \mu_1)^2}{2\sigma_1^2} + \frac{(y_i - \mu_2)^2}{2\sigma_2^2}\right)}$$
  - **Maximization** for finding parameters

$$\mu_1 = \frac{\sum_{i=1}^{N} (1-\gamma_i)y_i}{\sum_{i=1}^{N} (1-\gamma_i)}; \quad \mu_2 = \frac{\sum_{i=1}^{N} \gamma_i y_i}{\sum_{i=1}^{N} \gamma_i}; \quad \sigma_1^2 = \frac{\sum_{i=1}^{N} (1-\gamma_i)(y_i - \mu_1)^2}{\sum_{i=1}^{N} (1-\gamma_i)}; \quad \sigma_2^2 = \frac{\sum_{i=1}^{N} \gamma_i(y_i - \mu_2)^2}{\sum_{i=1}^{N} \gamma_i}; \quad \pi = \frac{\sum_{i=1}^{N} \gamma_i}{N};$$
EM in....simple words

• Given observed data, you need to come up with a generative model
• You choose a model that comprises of some hidden variables $\Delta_i$ (this is your belief!)
• Problem: To estimate the parameters of model
  – Assume some initial values parameters
  – Replace values of hidden variable with their expectation (given the old parameters)
  – Recompute new values of parameters (given $\Delta_i$)
  – Check for convergence using log-likelihood

$\Delta_i$
EM – Example (cont’d)

**Figure 8.6:** EM algorithm: observed data log-likelihood as a function of the iteration number.

**Selected iterations of the EM algorithm**
*For mixture example*

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.485</td>
</tr>
<tr>
<td>5</td>
<td>0.493</td>
</tr>
<tr>
<td>10</td>
<td>0.523</td>
</tr>
<tr>
<td>15</td>
<td>0.544</td>
</tr>
<tr>
<td>20</td>
<td>0.546</td>
</tr>
</tbody>
</table>
EM Summary

- An iterative approach for MLE
- Good idea when you have missing or latent data
- Has a nice property of convergence
- Can get stuck in local minima (try different starting points)
- Generally hard to calculate expectation over all possible values of hidden variables
- Still not much known about the rate of convergence
Today Outline

• Principles for Model Inference
  – Maximum Likelihood Estimation
  – Bayesian Estimation

• Strategies for Model Inference
  – EM Algorithm – simplify difficult MLE
    • Algorithm
    • Application
    • Theory
  – MCMC – samples rather than maximizing
Applications of EM

– Mixture models
– HMMs
– Latent variable models
– Missing data problems
– ...
Applications of EM (1)

- Fitting mixture models
Applications of EM (2)

- Probabilistic Latent Semantic Analysis (pLSA)
  - Technique from text for topic modeling

\[
P(w, d) \quad P(w | z) \quad P(z | d)
\]

Diagram:
- $P(w, d)$, $P(w | z)$, and $P(z | d)$ as probability distributions.
- $W$ and $Z$ as latent variables.
- $D$ as document collection.
Applications of EM (3)

- Learning parts and structure models
Applications of EM (4)

• Automatic segmentation of layers in video

http://www.psi.toronto.edu/images/figures/cutouts_vid.gif
Expectation Maximization (EM)

- Old idea (late 50’s) but formalized by Dempster, Laird and Rubin in 1977

single-variable + two-cluster case

\[ \pi = \Pr (\Delta = 1) \]

**Joint Prob. Model:**

1. \[ \Pr \left( y_i, \Delta_i | \theta \right) = \Pr \left( y_i | \Delta_i, \theta \right) \Pr (\Delta_i) \]

   \[ \begin{cases} \Pr (\Delta_i = 1) & \text{if } \Delta_i = 1 \\ \Pr (\Delta_i = 0) & \text{if } \Delta_i = 0 \end{cases} \]

   \[ \begin{bmatrix} N \left( y_i | \mu_1, \sigma_1 \right) (1 - \pi) \\ N \left( y_i | \mu_2, \sigma_2 \right) \pi \end{bmatrix} \Delta_i \]

2. \[ \left( \text{Marginal} \right) \text{Prob.} \]

   \[ \Pr (y_i | \theta) = \sum_{\Delta_i} \Pr (y_i | \Delta_i, \theta) \Pr (\Delta_i) \]

   \[ = N \left( y_i | \mu_1, \sigma_1 \right) (1 - \pi) + N \left( y_i | \mu_2, \sigma_2 \right) \pi \]

3. \[ \left( \text{Conditional} \right) \]

   \[ \Pr (y_i | \Delta_i, \theta) = \begin{cases} N \left( y_i | \mu_1, \sigma_1 \right) & \text{if } \Delta_i = 1 \\ N \left( y_i | \mu_2, \sigma_2 \right) & \text{if } \Delta_i = 0 \end{cases} \]

**Estep**

\[ \Pr (\Delta_i = 1 | y_i, \theta) = \frac{\Pr (y_i | \Delta_i = 1) \Pr (\Delta_i = 1 | \theta)}{\Pr (y_i | \theta)} \]
multi-variable
+
multi-cluster case

multi-variate \Rightarrow \text{Given } (x_1, x_2, \ldots, x_n)

multi-cluster \Rightarrow \text{complete } (z_1, z_2, \ldots, z_n)

with

each vector \( \vec{z}_i = (0, 0, 0, \ldots, 1, 0, 0, 0) \) \( \text{with position } i \)

\[ \Rightarrow \text{parameters } \theta \text{ includes } \]

\[ \left\{ \pi_j, \Sigma_j \right\}, j = 1, 2, \ldots, K \]

s.t. \[ \sum_{j=1}^{K} \pi_j = 1 \]


\[ p(x_i, z_i | \theta) = \prod_{j=1}^{K} \pi_j \cdot N(x_i | \mu_j, \Sigma_j) \]

\[ p(x_i, z_i = 1 | \theta) = \pi_j \cdot N(x_i | \mu_j, \Sigma_j) \]

2. Marginal

\[ p(x_i | \theta) = \sum_{j=1}^{K} \pi_j \cdot N(x_i | \mu_j, \Sigma_j) \]

3. Conditional

\[ p(z_i = 1 | x_i, \mu_j, \Sigma_j) = \frac{\prod_{j=1}^{K} \pi_j \cdot N(x_i | \mu_j, \Sigma_j)}{\sum_{k=1}^{K} \prod_{j=1}^{K} \pi_j \cdot N(x_i | \mu_k, \Sigma_k)} \]
Today Outline

• Principles for Model Inference
  – Maximum Likelihood Estimation
  – Bayesian Estimation

• Strategies for Model Inference
  – EM Algorithm – simplify difficult MLE
    • Algorithm
    • Application
  • Theory
    – MCMC – samples rather than maximizing
Why is Learning Harder?

- In fully observed iid settings, the complete log likelihood decomposes into a sum of local terms.
  \[
  \ell_c(\theta; D) = \log p(x, z \mid \theta) = \log p(z \mid \theta_z) + \log p(x \mid z, \theta_x)
  \]

- When with latent variables, all the parameters become coupled together via marginalization.
  \[
  / (\theta; D) = \log p(x \mid \theta) = \log \sum_z p(z \mid \theta_z)p(x \mid z, \theta_x)
  \]
Gradient Learning for mixture models

- We can learn mixture densities using gradient descent on the observed log likelihood. The gradients are quite interesting:

\[
\ell (\theta) = \log p(x \mid \theta) = \log \sum_k \pi_k p_k(x \mid \theta_k)
\]

\[
\frac{\partial \ell}{\partial \theta} = \frac{1}{p(x \mid \theta)} \sum_k \pi_k \frac{\partial p_k(x \mid \theta_k)}{\partial \theta}
\]

\[
= \sum_k \frac{\pi_k}{p(x \mid \theta)} p_k(x \mid \theta_k) \frac{\partial \log p_k(x \mid \theta_k)}{\partial \theta}
\]

\[
= \sum_k \pi_k \frac{p_k(x \mid \theta_k)}{p(x \mid \theta)} \frac{\partial \log p_k(x \mid \theta_k)}{\partial \theta_k} = \sum_k r_k \frac{\partial \ell_k}{\partial \theta_k}
\]

- In other words, the gradient is the responsibility weighted sum of the individual log likelihood gradients.
- Can pass this to a conjugate gradient routine.
Parameter Constraints

- Often we have constraints on the parameters, e.g. $\sum_k k$ being symmetric positive definite.
- We can use constrained optimization, or we can re-parameterize in terms of unconstrained values.
  - For normalized weights, softmax to e.g. $\sum_{j=1}^{K} \pi_j = 1$
  - For covariance matrices, use the Cholesky decomposition:
    $$\Sigma^{-1} = A^T A$$
    where $A$ is upper diagonal with positive diagonal:
    $$A_{ii} = \exp(\lambda_i) > 0 \quad A_{ij} = \eta_{ij} \quad (j > i) \quad A_{ij} = 0 \quad (j < i)$$
  - Use chain rule to compute
    $$\frac{\partial e}{\partial \pi} \cdot \frac{\partial e}{\partial A}.$$
Identifiability

• A mixture model induces a multi-modal likelihood.
• Hence gradient ascent can only find a local maximum.
• Mixture models are unidentifiable, since we can always switch the hidden labels without affecting the likelihood.
• Hence we should be careful in trying to interpret the “meaning” of latent variables.
Expectation-Maximization (EM) Algorithm

• EM is an Iterative algorithm with two linked steps:
  – E-step: fill-in hidden values using inference: $p(z|x, \theta^t)$.
  – M-step: update parameters (t+1) rounds using standard MLE/MAP method applied to completed data

• We will prove that this procedure monotonically improves (or leaves it unchanged). Thus it always converges to a local optimum of the likelihood.
Theory underlying EM

• What are we doing?

• Recall that according to MLE, we intend to learn the model parameter that would have maximize the likelihood of the data.

• But we do not observe $z$, so computing

$$\ell_c(\theta; D) = \log \sum_z p(x, z | \theta) = \log \sum_z p(z | \theta_z) p(x | z, \theta_x)$$

is difficult!

• What shall we do?
(1) Incomplete Log Likelihoods

- Incomplete log likelihood

With $z$ unobserved, our objective becomes the log of a marginal probability:

- This objective won't decouple

$$l(\theta; x) = \log p(x | \theta) = \log \sum_{z} p(x, z | \theta)$$
(2) Complete Log Likelihoods

• Complete log likelihood

Let $X$ denote the observable variable(s), and $Z$ denote the latent variable(s). If $Z$ could be observed, then

$$l_c(\theta; x, z) = \log p(x, z | \theta) = \log p(z | \theta_z)p(x | z, \theta_x)$$

– Usually, optimizing $l_c()$ given both $Z$ and $X$ is straightforward (c.f. MLE for fully observed models).
– Recalled that in this case the objective for, e.g., MLE, decomposes into a sum of factors, the parameter for each factor can be estimated separately.
– But given that $Z$ is not observed, $l_c()$ is a random quantity, cannot be maximized directly.
Three types of log-likelihood over multiple observed samples ($x_1, x_2, \ldots, x_N$)

- **Observed data**
  \[ x = (x_1, x_2, \ldots, x_N) \]

- **Latent variables**
  \[ z = (z_1, z_2, \ldots, z_N) \]

- **Iteration index**
  \[ t \]

Log-likelihood [Incomplete log-likelihood (ILL)]
\[
 l(\theta; x) = \log p(x|\theta) = \log \prod_x p(x|\theta) = \sum_x \log \sum_z p(x, z|\theta)
\]

Complete log-likelihood (CLL)
\[
 l_c(\theta; x, z) \triangleq \sum_x \log p(x, z | \theta)
\]

Expected complete log-likelihood (ECLL)
\[
 E_q[f(z)] = \sum_z q(z) f(z)
\]

- **Log-likelihood [Incomplete log-likelihood (ILL)]**
  \[
  E_q[f(z)] = \sum_z q(z) f(z)
  \]

- **Complete log-likelihood (CLL)**
  \[
  E_q[f(z)] = \sum_z q(z) f(z)
  \]

- **Expected complete log-likelihood (ECLL)**
  \[
  E_q[f(z)] = \sum_z q(z) f(z)
  \]
(3) Expected Complete Log Likelihood

- For any distribution $q(z)$, define expected complete log likelihood (ECLL):
  - CLL is random variable $\Rightarrow$ ECLL is a deterministic function of $\theta$
  - Linear in $\text{CLL()}$ —— inherit its factorizability
  - Does maximizing this surrogate yield a maximizer of the likelihood?

\[
ECLL = \left\langle l_c(\theta; x, z) \right\rangle_q = \sum_z q(z | x, \theta) \log p(x, z | \theta)
\]
Jensen’s inequality

Concave function \( f(x) \)

\[ x_t = (1-t)a + tb \]

\[ \Rightarrow f(x_t) \geq (1-t)f(a) + tf(b) \]

\[ \Rightarrow f(\sum_{j=1}^{M} \lambda_j x_j) \geq \sum_{j=1}^{M} \lambda_j f(x_j) \]

where \( \sum \lambda_j = 1 \)

\[ f(E[X]) \geq E[f(X)] \]
Jensen’s inequality

- Jensen’s inequality

\[ I LL = l(\theta; x) = \log p(x | \theta) \]
\[ = \log \sum_z p(x, z | \theta) \]
\[ = \log \sum_z q(z | x) \left( \frac{p(x, z | \theta)}{q(z | x)} \right) \]
\[ \geq \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)} \]
\[ = \sum_z q(z | x) \log p(x, z | \theta) - \sum_z q(z | x) \log q(z | x) \]
\[ = ECLL + H_q \]

\[ ECLL = \left< l_c(\theta; x, z) \right>_q = \sum_q q(z | x, \theta) \log p(x, z | \theta) \]

\[ f = \log (\cdot) \]

\[ E_q[f(E_{\theta} f(\frac{p(x, z | \theta)}{q(z | x)}))] \]

\[ E_q[f(\cdot)] = \sum_q q(z) f(\cdot) \]

\[ \Rightarrow \sum_q q(z) f(\cdot) \]

\[ \Rightarrow e(\theta; x) \geq \left< e_c(\theta; x, z) \right>_q + H_q \]

\[ I LL \geq ECLL + H_q \]
Lower Bounds and Free Energy

• For fixed data \( x \), define a functional called the free energy:

\[
F(q, \theta) = \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)} \leq \ell(\theta; x)
\]

• The EM algorithm is coordinate-ascent on \( F \):
  
  - E-step: \( q^{t+1} = \arg \max_q F(q, \theta^t) \)
  
  - M-step: \( \theta^{t+1} = \arg \max_\theta F(q^{t+1}, \theta^t) \)
How EM optimize ILL?
**E-step: maximization of w.r.t. $q$**

- **Claim:**
  \[
  q^{t+1} = \arg\max_{q} F(q, \theta^t) = p(z \mid x, \theta^t)
  \]
  - This is the posterior distribution over the latent variables given the data and the parameters. Often we need this at test time anyway (e.g. to perform clustering).

- **Proof (easy):** this setting attains the bound of ILL
  \[
  F(p(z \mid x, \theta^t), \theta^t) = \sum_z p(z \mid x, \theta^t) \log \frac{p(x, z \mid \theta^t)}{p(z \mid x, \theta^t)}
  \]
  \[
  = \sum_z p(z \mid x, \theta^t) \log p(x \mid \theta^t)
  \]
  \[
  = \log p(x \mid \theta^t) = \ell(\theta^t; x)
  \]

- **Can also show this result using variational calculus or the fact that**
  \[
  \ell(\theta; x) - F(q, \theta) = KL(q \parallel p(z \mid x, \theta))
  \]
E-step: Alternative derivation

\[ \ell(\theta; x) \geq F(q, \theta) \]

\[ \ell(\theta; x) - F(q, \theta) = \text{KL}(q \parallel p(z \mid x, \theta)) \]

\[ = l(\theta; x) - \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \]

\[ = \sum_z q(z \mid x) \log p(x \mid \theta) - \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \]

\[ = \sum_{q} q(\beta \mid x) \log \frac{q(\beta \mid x)}{p(\beta \mid x, \theta)} \]

\[ = D_{\text{KL}}(q(z \mid x) \parallel p(z \mid x, \theta)) \]

\[ \Rightarrow D_{\text{KL}} = 0 \quad \text{iff} \quad q = p \quad \text{almost everywhere} \]
M-step: maximization w.r.t. $\theta$

- Note that the free energy breaks into two terms:

\[
F(q, \theta) = \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)} = \sum_z q(z | x) \log p(x, z | \theta) - \sum_z q(z | x) \log q(z | x)
\]

\[
= \langle \ell_c(\theta; x, z) \rangle_q + H_q
\]

- The first term is the expected complete log likelihood (energy) and the second term, which does not depend on $\theta$, is the entropy.
M-step: maximization w.r.t. $\theta$

- Thus, in the M-step, maximizing with respect to $\theta$ for fixed $q$ we only need to consider the first term:

$$
\theta^{t+1} = \arg \max_{\theta} \langle \ell_c (\theta; x, z) \rangle_{q^{t+1}} = \arg \max_{\theta} \sum_z q(z \mid x) \log p(x, z \mid \theta)
$$

- Under optimal $q^{t+1}$, this is equivalent to solving a standard MLE of fully observed model $p(x, z \mid \theta)$, with the **sufficient statistics** involving $z$ replaced by their expectations w.r.t. $p(z \mid x, \theta)$. 

Summary: EM Algorithm

• A way of maximizing likelihood function for latent variable models. Finds MLE of parameters when the original (hard) problem can be broken up into two (easy) pieces:
  1. Estimate some “missing” or “unobserved” data from observed data and current parameters.
  2. Using this “complete” data, find the maximum likelihood parameter estimates.

• Alternate between filling in the latent variables using the best guess (posterior) and updating the parameters based on this guess:
  - E-step: \( q^{t+1} = \arg \max_q F(q, \theta^t) \)
  - M-step: \( \theta^{t+1} = \arg \max_{\theta} F(q^{t+1}, \theta^t) \)

• In the M-step we optimize a lower bound on the likelihood. In the E-step we close the gap, making bound=likelihood.
How EM optimize ILL?
A Report Card for EM

• Some good things about EM:
  – no learning rate (step-size) parameter
  – automatically enforces parameter constraints
  – very fast for low dimensions
  – each iteration guaranteed to improve likelihood
  – Calls inference and fully observed learning as subroutines.

• Some bad things about EM:
  – can get stuck in local minima
  – can be slower than conjugate gradient (especially near convergence)
  – requires expensive inference step
  – is a maximum likelihood/MAP method
References

• Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides

• The EM Algorithm and Extensions by Geoffrey J. MacLauchlan, Thriyambakam Krishnan