Iterative Algorithm

- Motivates a coordinate ascent-like algorithm:
  1. Infer missing values $z^i$ given estimate of parameters $\hat{\theta}$
  2. Optimize parameters to produce new $\hat{\theta}$ given "filled in" data $z^i$
  3. Repeat

- Example: MoG (derivation soon... + HW)
  1. Infer "responsibilities"
     \[
     r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) = \max_{\pi_k} \pi_k \prod_{j \neq k} \frac{n_j^{-1}}{\pi_j} p(x^i, \phi_{j}^{(t-1)})
     \]
  2. Optimize parameters
     \[
     \max_w \text{w.r.t. } \pi_k:
     \]
     \[
     \max_w \text{w.r.t. } \mu_k, \Sigma_k:
     \]
     \[
     \hat{\mu}_k = \frac{1}{N} \sum_{i=1}^N r_{ik} x^i
     \]
     \[
     \hat{\Sigma}_k = \frac{1}{N} \sum_{i=1}^N r_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T
     \]
More broadly applicable than just to mixture models considered so far

Model: $x$, observable – "incomplete" data
$y$, not (fully) observable – "complete" data
$\theta$, parameters

Interested in maximizing (wrt $\theta$):

$$p(x | \theta) = \sum_{y} p(x, y | \theta) = \sum_{y} p(x | y, \theta) p(y | \theta)$$

Special case:

$x = g(y)$

e.g. $y = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$

Step 1
- Rewrite desired likelihood in terms of complete data terms

$$p(y | \theta) = p(y | x, \theta) p(x | \theta)$$

$$\Rightarrow \log p(x | \theta) = \log \theta p(y | x, \theta) - \log p(y | \theta)$$

Step 2
- Assume estimate of parameters $\hat{\theta}$
- Take expectation with respect to $p(y | x, \hat{\theta})$

$$L_x(\theta) = \mathbb{E}[\log p(y | x, \hat{\theta})] + \mathbb{E}[-\log p(y | x, \hat{\theta}) | x, \hat{\theta}]$$

$$= \mathbb{V}(\theta, \hat{\theta})$$
Expectation Maximization (EM) – Derivation

**Step 3**
- Consider log likelihood of data at any $\theta$ relative to log likelihood at $\hat{\theta}$

$$L_x(\theta) - L_x(\hat{\theta}) = [U(\theta, \hat{\theta}) - U(\hat{\theta}, \hat{\theta})] + [V(\theta, \hat{\theta}) - V(\hat{\theta}, \hat{\theta})]$$

- **Aside: Gibbs Inequality** $E_p[\log p(x)] \geq E_p[\log q(x)]$

**Proof:**
Use Jensen’s Ineq: $E[ f(x)] \leq f( E[x])$ for any convex function $f$.

Here:

$$E_p[\log q] - E_p[\log p] = E_p[\log \frac{q}{p}]$$

$$\leq \log E_p[\frac{q}{p}] = \log \int_x p(x) \frac{q(x)}{p(x)} dx = 0$$

**Step 4**
- Determine conditions under which log likelihood at $\theta$ exceeds that at $\hat{\theta}$

Using Gibbs inequality:

$$V(\theta, \hat{\theta}) = E[-\log p(y|x, \theta) | x, \hat{\theta}] \geq E[-\log p(y|x, \hat{\theta}) | x, \hat{\theta}]$$

If $U(\theta, \hat{\theta}) = U(\hat{\theta}, \hat{\theta})$,

Then $L_x(\theta) \geq L_x(\hat{\theta})$

choosing $\theta$ s.t. this is true means we’re moving in the right direction (or at least not wrong)
Motivates EM Algorithm

- Initial guess: \( \hat{\theta}^{(0)} \)
- Estimate at iteration \( t \): \( \hat{\theta}^{(t)} \)

**E-Step**
Compute
\[
U(\theta, \hat{\theta}^{(t)}) = E[\log p(y | \theta) | x, \hat{\theta}^{(t)}]
\]

**M-Step**
Compute
\[
\hat{\theta}^{(t+1)} = \arg \max_\theta U(\theta, \hat{\theta}^{(t)})
\]
From above, \( U(\hat{\theta}^{t+1}, \hat{\theta}^{(t)}) \geq U(\hat{\theta}^{t}, \hat{\theta}^{(t)}) \)
\( \Rightarrow L_x(\hat{\theta}^{(t+1)}) \geq L_x(\hat{\theta}^{(t)}) \)

Example – Mixture Models

- **E-Step**
  Compute
  \[
  U(\theta, \hat{\theta}^{(t)}) = E[\log p(y | \theta) | x, \hat{\theta}^{(t)}]
  \]

- **M-Step**
  Compute
  \[
  \hat{\theta}^{(t+1)} = \arg \max_\theta U(\theta, \hat{\theta}^{(t)})
  \]

Consider \( y_i = \{z_i, x_i\} \)
\( i.i.d. \)
\[
p(x_i, z_i | \theta) = \pi_z p(x_i | \phi_{z_i})
\]
\[
E_{q_t}[\log p(y | \theta)] = \sum_i E_{q_t}[\log p(x_i, z_i | \theta)] = \sum_i \left( \prod_k (\pi_k p(x_i | \phi_k)) \log p(z_i | x_i, \phi_k) \right)
\]

\[
\frac{q_i}{\max_{i'} q_{i'}} \prod_k \pi_k = \frac{\sum_{i} r_{ik}}{N}
\]
Coordinate Ascent Behavior

- Bound log likelihood:
  \[ L_x(\theta) = U(\theta, \hat{\theta}^{(t)}) + V(\theta, \hat{\theta}^{(t)}) \]
  \[ \geq U(\theta, \hat{\theta}^{(0)}) + V(\theta^{(0)}, \theta^{(0)}) = LB_x^{(0)}(\theta, \theta^{(0)}) \]
  \[ L_x^{(t)}(\theta) = U(\theta^{(t)}, \hat{\theta}^{(t)}) + V(\theta^{(t)}, \hat{\theta}^{(t)}) = LB_x^{(t)}(\theta, \theta^{(0)}) \]

Comments on EM

- Since Gibbs inequality is satisfied with equality only if \( p=q \), any step that changes \( \theta \) should strictly increase likelihood.
- In practice, can replace the M-Step with increasing U instead of maximizing it (Generalized EM).
- Under certain conditions (e.g., in exponential family), can show that EM converges to a stationary point of \( L_x(\theta) \).
- Often there is a natural choice for \( y \) ... has physical meaning.
- If you want to choose any \( y \), not necessarily \( x=g(y) \), replace \( p(y \mid \theta) \) in U with \( p(y, x \mid \theta) \).
Initialization

- In mixture model case where $y^i = \{z^i, x^i\}$ there are many ways to initialize the EM algorithm.

- Examples:
  - Choose K observations at random to define each cluster. Assign other observations to the nearest "centroid" to form initial parameter estimates.
  - Pick the centers sequentially to provide good coverage of data.
  - Grow mixture model by splitting (and sometimes removing) clusters until K clusters are formed.

- Can be quite important to convergence rates in practice or quality of solution.

What you should know

- K-means for clustering:
  - algorithm
  - converges because it’s coordinate ascent.

- EM for mixture of Gaussians:
  - How to “learn” maximum likelihood parameters (locally max. like.) in the case of unlabeled data.

- Be happy with this kind of probabilistic analysis.

- Remember, E.M. can get stuck in local minima, and empirically it DOES.

- EM is coordinate ascent.
Dimensionality Reduction
PCA

Machine Learning – CSE4546
Carlos Guestrin
University of Washington

November 13, 2013
Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has
- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – hard to visualize more than 3D or 4D
  - discover “intrinsic dimensionality” of data
    - high dimensional data that is truly lower dimensional

Lower dimensional projections

- Rather than picking a subset of the features, we can new features that are combinations of existing features
  \[ z = \begin{pmatrix} 2.5x_1 - 2.9x_2 & \ldots & 3x_5 \end{pmatrix} \]
  - model: \( z = Ax \)
  - learn \( A \) from data
  - minimize reconstruction error
- Let’s see this in the unsupervised setting
  - just \( X \), but no \( Y \)
Linear projection and reconstruction

Project $n$-dimensional data into $k$-dimensional space while preserving information:
- e.g., project space of 10000 words into 3-dimensions
- e.g., project 3-d into 2-d

Choose projection with minimum reconstruction error

Principal component analysis – basic idea
**Linear projections, a review**

- Project a point into a (lower dimensional) space:
  - **Point**: \( x = (x_1, \ldots, x_d) \)
  - **Select a basis** – set of basis vectors – \((u_1, \ldots, u_k)\)
    - we consider orthonormal basis:
      - \( u_i \cdot u_i = 1 \), and \( u_i \cdot u_j = 0 \) for \( i \neq j \)
  - **Select a center** – \( \bar{x} \), defines offset of space
  - **Best coordinates** in lower dimensional space defined by dot-products: \((z_1, \ldots, z_k)\), \( z_i = (x - \bar{x}) \cdot u_i \)
    - minimum squared error

**PCA finds projection that minimizes reconstruction error**

- Given \( N \) data points: \( x^i = (x^i_1, \ldots, x^i_d) \), \( i = 1 \ldots N \)
- Will represent each point as a projection:
  - \( \hat{x}^i = \bar{x} + \sum_{j=1}^{k} z_j \cdot u_j \) where: \( \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x^i \) and \( z_j = (x^i - \bar{x}) \cdot u_j \)

- **PCA**:
  - Given \( k \ll d \), find \((u_1, \ldots, u_k)\)
    - minimizing reconstruction error:
      - \( \text{error}_k = \sum_{i=1}^{N} (x^i - \hat{x}^i)^2 \)
Understanding the reconstruction error

Given \( k \ll d \), find \((u_1, \ldots, u_k)\)

\[
\min \sum_{j=1}^{k} z_j^2 \text{ minimizing reconstruction error:}
\]

\[
\text{error}_k = \sum_{i=1}^{N} (x^i - \bar{x}_i)^2
\]

Note that \( x^i \) can be represented exactly by \( d \)-dimensional projection:

\[
x^i = \bar{x} + \sum_{j=1}^{d} z_j u_j
\]

Rewriting error:

\[
\text{error}_k = \sum_{i=1}^{N} (x^i - \bar{x}_i)^2 = \sum_{i=1}^{N} \left( \sum_{j=1}^{d} z_j^i u_j - (x^i - \bar{x}) \right)^2
\]

\[
= \sum_{i=1}^{N} \left( \sum_{j=1}^{d} z_j^i u_j \right)^2 + \sum_{i=1}^{N} \left( \sum_{j=1}^{d} z_j^i u_j \right) \sum_{j=1}^{d} z_j^i u_j
\]

\[
= \sum_{i=1}^{N} \sum_{j=1}^{d} (z_j^i)^2 \leq \min \text{ error} \equiv \min \text{ projection on to ignored direction}
\]

Reconstruction error and covariance matrix

\[
\text{error}_k = \sum_{i=k+1}^{N} \sum_{j=1}^{d} [u_j \cdot (x^i - \bar{x})]^2
\]

\[
= \sum_{i=k+1}^{N} \sum_{j=1}^{d} \left( \sum_{j=1}^{k+1} z_j^i u_j \right)^2
\]

\[
= \sum_{i=k+1}^{N} \left[ \sum_{j=1}^{k+1} (z_j^i)^2 \right] \sum_{j=1}^{k+1} u_j
\]

\[
\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^i - \bar{x})(x^i - \bar{x})^T
\]

\[
\Sigma = \begin{pmatrix} \sigma_{k\times k} & \sigma_{k	imes d} \\ \sigma_{d\times k} & \sigma_{d\times d} \end{pmatrix}
\]

\[
\sigma_{uv} = \frac{1}{N} \sum_{i=1}^{N} (x^i - \bar{x})(x^i - \bar{x})
\]

\[
\Sigma = \frac{1}{N} \sum_{i=1}^{N} (x^i - \bar{x})(x^i - \bar{x})^T
\]