Machine Learning

10-701/15-781, Fall 2011

Expectation Maximization

Eric Xing

Lecture 9, October 10, 2011

Reading: Chap. 9, C.B book

© Eric Xing @ CMU, 2006-2011
Unobserved Variables

- A variable can be unobserved (latent) because:
  - it is an imaginary quantity meant to provide some simplified and abstractive view of the data generation process
    - e.g., speech recognition models, mixture models …
  - it is a real-world object and/or phenomena, but difficult or impossible to measure
    - e.g., the temperature of a star, causes of a disease, evolutionary ancestors …
  - it is a real-world object and/or phenomena, but sometimes wasn’t measured, because of faulty sensors; or was measure with a noisy channel, etc.
    - e.g., traffic radio, aircraft signal on a radar screen.

- Discrete latent variables can be used to partition/cluster data into sub-groups (mixture models, forthcoming).

- Continuous latent variables (factors) can be used for dimensionality reduction (factor analysis, etc., later lectures).

Uni-modal and multi-modal distributions
Mixture Models

A density model $p(x)$ may be multi-modal.
We may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
Each mode may correspond to a different sub-population (e.g., male and female).
Gaussian Mixture Models (GMMs)

- Consider a mixture of $K$ Gaussian components:
  - $Z$ is a latent class indicator vector:
    \[
    p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_{nk}}
    \]
  - $X$ is a conditional Gaussian variable with a class-specific mean/covariance:
    \[
    p(x_n | z^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x_n - \mu_k)^\top \Sigma_k^{-1} (x_n - \mu_k) \right\}
    \]
  - The likelihood of a sample:
    \[
    p(x_n | \mu, \Sigma) = \sum_k p(z^k = 1 | \pi) p(x_n | z^k = 1, \mu, \Sigma) = \sum_k \pi_k \prod_{z_{nk}} (\pi_k)^{z_{nk}} N(x_n | \mu_k, \Sigma_k)
    \]

- This model can be used for unsupervised clustering.
  - This model (fit by AutoClass) has been used to discover new kinds of stars in astronomical data, etc.
Learning mixture models

Why is Learning Harder?

- In fully observed iid settings, the log likelihood decomposes into a sum of local terms.
  \[ \ell(\theta; D) = \log p(x, z | \theta) = \log p(z | \theta_2) + \log p(x | z, \theta_1) \]

- With latent variables, all the parameters become coupled together via marginalization
  \[ \ell(\theta; D) = \log \sum_z p(x, z | \theta) = \log \sum_z p(z | \theta_2) p(x | z, \theta_1) \]
Gradient Learning for mixture models

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

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

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

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

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

- 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. \(\sum_k \pi_k = 1\), \(\Sigma\) being symmetric positive definite (hence \(\Sigma > 0\)).
- We can use constrained optimization, or we can reparameterize in terms of unconstrained values.
  - For normalized weights, use the softmax transform:
  - For covariance matrices, use the Cholesky decomposition:

\[
\Sigma^{-1} = A^T A
\]

where \(A\) is upper diagonal with positive diagonal:

\[
A_{ii} = \exp(\lambda_i) > 0 \quad A_{ij} = \eta_{ij} \quad (j > i) \quad A_{ij} = 0 \quad (j < i)
\]

the parameters \(\gamma_i, \lambda, \eta_{ij} \in \mathbb{R}\) are unconstrained.
- Use chain rule to compute

\[
\frac{\partial \ell}{\partial \pi} \cdot \frac{\partial \ell}{\partial A}
\]
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.
Toward the EM algorithm

- E.g., A mixture of K Gaussians:
  - \( Z \) is a latent class indicator vector
    \[
    p(z_n) = \text{multi}(z_n, \pi) = \prod_k (\pi_k)^{z_n}
    \]
  - \( X \) is a conditional Gaussian variable with a class-specific mean/covariance
    \[
    p(x_n \mid z_n = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right)
    \]
  - The likelihood of a sample:
    \[
    p(x_n \mid \mu, \Sigma) = \sum_{z_n} p(z_n = 1 \mid x_n) p(x_n \mid z_n = 1, \mu, \Sigma)
    = \sum_k \prod_k \left(\pi_k \right)^{z_n} N(x_n \mid \mu_k, \Sigma_k) = \sum_k \pi_k N(x_n \mid \mu_k, \Sigma_k)
    \]

Toward the EM algorithm

- Recall MLE for completely observed data
  - Data log-likelihood
    \[
    \ell(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)
    \]
    \[
    = \sum_n \log \prod_k \pi_k^{z_n} + \sum_n \log \prod_k N(x_n \mid \mu_k, \Sigma_k)
    = \sum_n z_n \log \pi_k - \sum_n \frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + C
    \]
  - MLE
    \[
    \hat{\mu}_{k, \text{MLE}} = \operatorname{arg\ max}_\mu \ell(\theta; D), \quad \Rightarrow \hat{\mu}_{k, \text{MLE}} = \frac{\sum_n z_n x_n}{\sum_n z_n}
    \]
  - What if we do not know \( z_n \)?
Expectation-Maximization (EM) Algorithm

- EM is an optimization strategy for objective functions that can be interpreted as likelihoods in the presence of missing data.
- It is much simpler than gradient methods:
  - No need to choose step size.
  - Enforces constraints automatically.
  - Calls inference and fully observed learning as subroutines.
- EM is an iterative algorithm with two linked steps:
  - E-step: fill-in hidden values using inference, \( p(z|x, \theta) \).
  - M-step: update parameters \( t+1 \) 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.
K-means

- Start:
  - "Guess" the centroid $\mu_k$ and covariance $\Sigma_k$ of each of the K clusters

- Loop
  - For each point $n=1$ to $N$,
    compute its cluster label:
    $$z_n^{(t)} = \arg\max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1}(x_n - \mu_k^{(t)})$$
  - For each cluster $k=1:K$
    $$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$
    $$\Sigma_k^{(t+1)} = ...$$

Expectation-Maximization

- Start:
  - "Guess" the centroid $\mu_k$ and covariance $\Sigma_k$ of each of the K clusters

- Loop
How is EM derived?

- A mixture of $K$ Gaussians:
  - $Z$ is a latent class indicator vector
    
    $p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_{nk}}$
  
  - $X$ is a conditional Gaussian variable with a class-specific mean/covariance
    
    $p(x_n | z_{nk} = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^k} \exp\left\{ -\frac{1}{2} (x_n - \mu_k)^\top \Sigma_k^{-1} (x_n - \mu_k) \right\}$

  - The likelihood of a sample:
    
    $p(x_n | \mu, \Sigma) = \sum_k p(z_n = k | \pi) p(x_n | z_{nk} = 1, \mu, \Sigma) = \sum_k \prod_k \left( \sigma_k \Sigma_k^{-1/2} N(x_n | \mu_k, \Sigma_k) \right) = \sum_k \pi_k N(x_n | \mu_k, \Sigma_k)$

- The “complete” likelihood
  
  $p(x_n, z_{nk} | \mu, \Sigma) = p(z_{nk} = k | \pi) p(x_n | z_{nk} = k, \mu, \Sigma) = \pi_k N(x_n | \mu_k, \Sigma_k)$

  $p(x_n, z_{nk} | \mu, \Sigma) = \prod_k \left( \sigma_k \Sigma_k^{-1/2} N(x_n | \mu_k, \Sigma_k) \right) = \pi_k N(x_n | \mu_k, \Sigma_k)$

  But this is itself a random variable! Not good as objective function.

- The complete log likelihood:
  
  $\ell(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma)$

  $= \sum_n \log \prod_k \pi_k^{z_{nk}} + \log \prod_k N(x_n | \mu_k, \sigma)^{z_{nk}}$

  $= \sum_n \sum_k z_{nk} \log \pi_k - \sum_n \sum_k z_{nk} \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C$

- The expected complete log likelihood
  
  $\langle \ell(\theta; x, z) \rangle = \sum_n \langle \log p(z_n | \pi) \rangle_{p(z|x)} + \sum_n \langle \log p(x_n | z_n, \mu, \Sigma) \rangle_{p(z|x)}$

  $= \sum_n \sum_k \langle z_{nk} \rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \langle z_{nk}^2 \rangle (x_n - \mu_k)^\top \Sigma_k^{-1} (x_n - \mu_k) + \log|\Sigma_k| + C$
We maximize $\ell_c(\theta)$ iteratively using the following iterative procedure:

- **E-step**: computing the expected value of the sufficient statistics of the hidden variables (i.e., $z$) given current est. of the parameters (i.e., $x$ and $\mu$).

$$
\tau_n^{(t)} = \left\{ z_n^k \right\} = \mathbb{E}(z_n^k | x_n, \mu_n^{(t)}, \Sigma_n^{(t)}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_i \pi_i \mathcal{N}(x_n | \mu_i^{(t)}, \Sigma_i^{(t)})}
$$

- **M-step**: compute the parameters under current results of the expected value of the hidden variables

$$
\pi_k^* = \arg \max \mathbb{E}(\ell_c(\theta)) \Rightarrow \frac{1}{N} \mathbb{E}(\ell_c(\theta)) = 0, \forall k, \text{ s.t. } \sum \pi_k = 1
$$

$$
\mu_k^* = \arg \max \mathbb{E}(\ell_c(\theta)) \Rightarrow \mu_k^{(t+1)} = \frac{\sum_{n=1}^{N} \mathbb{E}(z_n^k) x_n}{\sum_{n=1}^{N} \mathbb{E}(z_n^k)}
$$

$$
\Sigma_k^* = \arg \max \mathbb{E}(\ell_c(\theta)) \Rightarrow \Sigma_k^{(t+1)} = \frac{\sum_{n=1}^{N} \mathbb{E}(z_n^k) (x_n - \mu_k^{(t+1)}) (x_n - \mu_k^{(t+1)})^T}{\sum_{n=1}^{N} \mathbb{E}(z_n^k)}
$$

- Here we are essentially doing inference.

This is isomorphic to MLE except that the variables that are hidden are replaced by their expectations (in general they will by replaced by their corresponding “sufficient statistics.”)
Compare: K-means

- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means "E-step" we do hard assignment:
  \[ z_{n}^{(r)} = \arg \max_{k} (x_{n} - \mu_{k}^{(r)})^T \sum_{n} \delta(z_{n}^{(r)}, k) (x_{n} - \mu_{k}^{(r)}) \]
- In the K-means "M-step" we update the means as the weighted sum of the data, but now the weights are 0 or 1:
  \[ \mu_{k}^{(r+1)} = \frac{\sum_{n} \delta(z_{n}^{(r)}, k) x_{n}}{\sum_{n} \delta(z_{n}^{(r)}, k)} \]

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?
Complete & Incomplete 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
  \[
  \mathcal{L}_c(\theta; x, z) \overset{\text{def}}{=} \log p(x, z | \theta)
  \]
  - Usually, optimizing \( \mathcal{L}_c() \) 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, \( \mathcal{L}_c() \) is a random quantity, cannot be maximized directly.

- Incomplete log likelihood
  With \( z \) unobserved, our objective becomes the log of a marginal probability:
  \[
  \mathcal{L}_i(\theta; x) = \log p(x | \theta) = \log \sum_z p(x, z | \theta)
  \]
  - This objective won’t decouple

Expected Complete Log Likelihood

- For any distribution \( q(z) \), define expected complete log likelihood:
  \[
  \mathbb{E}_q(\mathcal{L}_c(\theta; x, z)) \overset{\text{def}}{=} \sum_z q(z | x, \theta) \log p(x, z | \theta)
  \]
  - A deterministic function of \( \theta \)
  - Linear in \( \mathcal{L}_c() \) --- inherit its factorizability
  - Does maximizing this surrogate yield a maximizer of the likelihood?

- Jensen’s inequality
  \[
  \mathcal{L}(\theta; x) = \log p(x | \theta) = \log \sum_z p(x, z | \theta) = \log \sum_z q(z | x) \frac{p(x, z | \theta)}{q(z | x)} \geq \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)} \Rightarrow \mathcal{L}(\theta; x) \geq \mathbb{E}_q(\mathcal{L}_c(\theta; x, z)) + H_q
  \]
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:** $\theta^{t+1} = \arg \max_\theta F(q^{t+1}, \theta)$

\[
\begin{align*}
\ell_c \text{ w.r.t. } q
\end{align*}
\]

- **Claim:**
  \[ q^{t+1} = \arg \max_q F(q, \theta^t) = p(z | x, \theta^t) \]
  - 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 classification).

- **Proof (easy):** this setting attains the bound $\ell(\theta, x) \geq F(q, \theta)$
  \[
  F(p(z|x, \theta^t), \theta^t) = \sum z p(z|x, \theta^t) \log \frac{p(x, z | \theta^t)}{p(z|x, \theta^t)}
  = \sum z p(z | x, \theta^t) \log p(x | \theta^t)
  = \log p(x | \theta^t) = \ell(\theta^t; x)
  \]

- Can also show this result using variational calculus or the fact that
  \[
  \ell(\theta; x) - F(q, \theta) = KL(q \parallel p(z | x, \theta))
  \]
E-step $\equiv$ plug in posterior expectation of latent variables

- Without loss of generality: assume that $p(x,z|\theta)$ is a generalized exponential family distribution:
  
  $$p(x,z|\theta) = \frac{1}{Z(\theta)} h(x,z) \exp \left\{ \sum_i \theta_i f_i(x,z) \right\}$$

- Special cases: if $p(X|Z)$ are GLIMs, then $f_i(x,z) = \eta_i^T(z) \xi_i(x)$

- The expected complete log likelihood under $q^{t-1} = p(z|x,\theta^t)$ is
  
  $$\langle \ell(\theta^t; x, z) \rangle_{q^{t-1}} = \sum_z q(z|x,\theta^t) \log p(x,z|\theta^t) - A(\theta)$$
  
  $$= \sum_i \theta_i^T \langle f_i(x,z) \rangle_{q(z|x,\theta^t)} - A(\theta)$$
  
  $$\overset{p-GLIM}{=} \sum_i \theta_i^T \langle \eta_i(z) \rangle_{q(z|x,\theta^t)} \xi_i(x) - A(\theta)$$


M-step: maximization of expected $\ell_c$ w.r.t. $\theta$

- Note that the free energy breaks into two terms:
  
  $$F(q,\theta) = \sum_z q(z|x) \log \frac{p(x,z|\theta)}{q(z|x)}$$
  
  $$= \sum_z q(z|x) \log p(x,z|\theta) - \sum_z q(z|x) \log q(z|x)$$
  
  $$= \langle \ell(\theta; x, z) \rangle_q + H_q$$

- The first term is the expected complete log likelihood (energy) and the second term, which does not depend on $\theta$, is the entropy.

- Thus, in the M-step, maximizing with respect to $\theta$ for fixed $q$ we only need to consider the first term:
  
  $$\theta^{t+1} = \arg \max_{\theta} \langle \ell(\theta; x, z) \rangle_{q_{t+1}} = \arg \max_{\theta} \sum_z q(z|x) \log p(x,z|\theta)$$

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

- In the M-step we optimize a lower bound on the likelihood. In the E-step we close the gap, making bound=likelihood.

EM Variants

- Sparse EM:
  Do not re-compute exactly the posterior probability on each data point under all models, because it is almost zero. Instead keep an “active list” which you update every once in a while.

- Generalized (Incomplete) EM:
  It might be hard to find the ML parameters in the M-step, even given the completed data. We can still make progress by doing an M-step that improves the likelihood a bit (e.g. gradient step). Recall the IRLS step in the mixture of experts model.
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

- 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