

Machine Learning

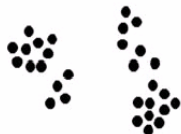
10-701/15-781, Spring 2008

Expectation Maximization

Eric Xing

Lecture 16, March 19, 2008

Reading: Chap. 9, C.B book



Eric Xing

1

Clustering



Eric Xing

2

Unobserved Variables

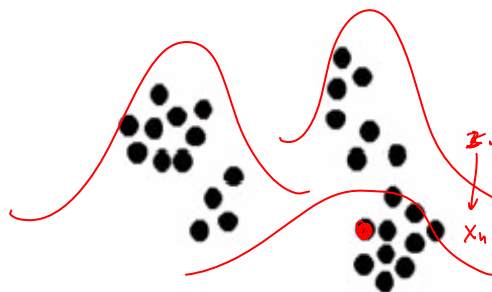


- A variable can be unobserved (latent) because:
 - it is an imaginary quantity meant to provide some simplified and abstractive view of the data generation process
 - e.g., speech recognition models, mixture models ...
 - it is a real-world object and/or phenomena, but difficult or impossible to measure
 - 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).

Eric Xing

3

Mixture Models



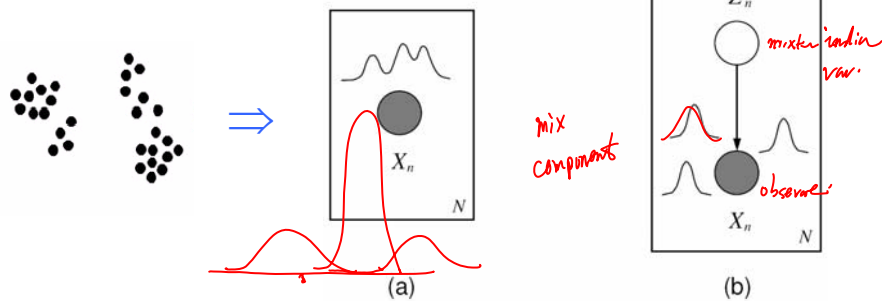
$$x \sim \left\{ N(\mu_k, \Sigma_k) \right\}_{k=1}^K$$

Eric Xing

4

Mixture Models, con'd

- 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).



Eric Xing

5

Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

- Z is a latent class indicator vector:

$$p(\mathbf{z}_n) = \text{multi}(\mathbf{z}_n; \pi) = \prod_k (\pi_k)^{z_n^k}$$

- X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n | z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\right\}$$

$p(x, z) \sim p(z) p(x|z)$

- The likelihood of a sample:

$$p(x_n | \mu, \Sigma) = \sum_k p(z^k = 1 | \pi) p(x_n | z^k = 1, \mu, \Sigma)$$

mixture proportion mixture component

$$= \sum_{z_n} \prod_k (\pi_k)^{z_n^k} \mathcal{N}(x_n; \mu_k, \Sigma_k)^{z_n^k} = \sum_k \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)$$

Eric Xing

6

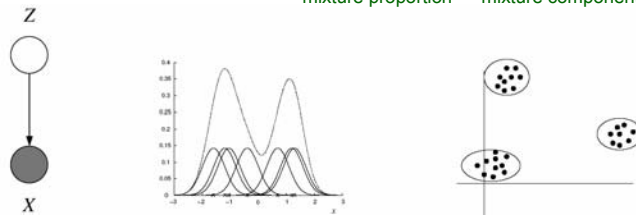
Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

$$p(x_n | \mu, \Sigma) = \sum_k \pi_k N(x, | \mu_k, \Sigma_k)$$

$p(z|x)$
 $z^* = \text{argmax}$

mixture proportion mixture component



- 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.

Eric Xing

7

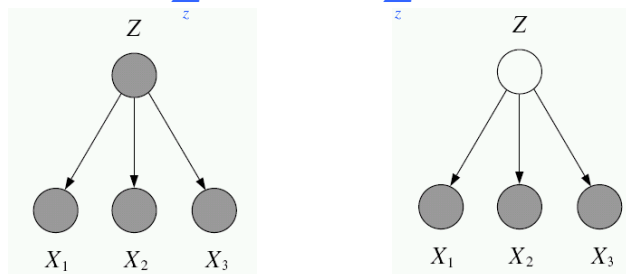
Why is Learning Harder?

- In fully observed iid settings, the log likelihood decomposes into a sum of local terms.

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

- With latent variables, all the parameters become coupled together via marginalization

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



Eric Xing

8

Gradient Learning for mixture models



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

$$\begin{aligned}\mathcal{L}(\theta) &= \log p(x|\theta) = \log \sum_k \pi_k p_k(x|\theta_k) \\ \frac{\partial \mathcal{L}}{\partial \theta} &= \frac{1}{p(x|\theta)} \sum_k \pi_k \frac{\partial p_k(x|\theta_k)}{\partial \theta} \\ &= \sum_k \frac{\pi_k}{p(x|\theta)} p_k(x|\theta_k) \frac{\partial \log p_k(x|\theta_k)}{\partial \theta} \\ &= \sum_k \pi_k \frac{p_k(x|\theta_k)}{p(x|\theta)} \frac{\partial \log p_k(x|\theta_k)}{\partial \theta_k} = \sum_k r_k \frac{\partial \ell_k}{\partial \theta_k}\end{aligned}$$

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

Eric Xing

9

Parameter Constraints

$$\begin{aligned}\pi_k &= \frac{\sum_r z_r^k}{N} \\ \mu_k &= \frac{\sum_r x_r}{N_k}\end{aligned}$$



- Often we have constraints on the parameters, e.g. $\sum_k \pi_k = 1$, Σ being symmetric positive definite (hence $\Sigma_{ii} > 0$).
- We can use constrained optimization, or we can reparameterize in terms of unconstrained values.

- For normalized weights, use the softmax transform: $\pi_k = \frac{\exp(\gamma_k)}{\sum_j \exp(\gamma_j)}$
- For covariance matrices, use the Cholesky decomposition:

$$\Sigma^{-1} = \mathbf{A}^T \mathbf{A} \quad \mathcal{L}(x, \gamma_k, \lambda_i, \eta_{ij})$$

where \mathbf{A} is upper diagonal with positive diagonal:

$$\mathbf{A}_{ii} = \exp(\lambda_i) > 0 \quad \mathbf{A}_{ij} = \eta_{ij} \quad (j > i) \quad \mathbf{A}_{ij} = 0 \quad (j < i)$$

the parameters $\gamma, \lambda, \eta_{ij} \in \mathbb{R}$ are unconstrained.

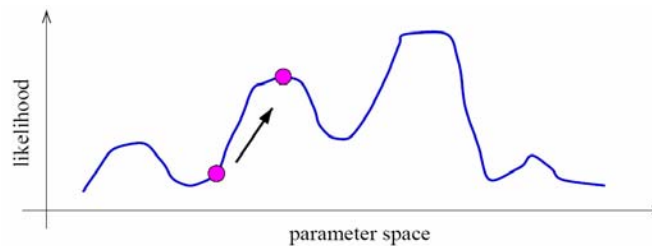
- Use chain rule to compute $\frac{\partial \mathcal{L}}{\partial \pi}, \frac{\partial \mathcal{L}}{\partial \mathbf{A}}$.

Eric Xing

10

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.



Eric Xing

11

Toward the EM algorithm

- E.g., A mixture of K Gaussians:

- Z is a latent class indicator vector

$$p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

- X is a conditional Gaussian variable with a class-specific mean/covariance

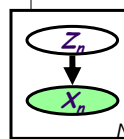
$$p(x_n | z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\right\}$$

- The likelihood of a sample:

$$\begin{aligned} p(x_n | \mu, \Sigma) &= \sum_k p(z_n^k = 1 | \pi) p(x_n | z_n^k = 1, \mu, \Sigma) \\ &= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k} \right) = \sum_k \pi_k N(x_n | \mu_k, \Sigma_k) \end{aligned}$$

Eric Xing

12



Toward the EM algorithm

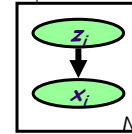


- Recall MLE for completely observed data
- Data log-likelihood

$$\begin{aligned}\mathcal{L}(\theta; D) &= \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma) \\ &= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k} \\ &= \sum_n \sum_k z_n^k \log(\pi_k) - \sum_n \sum_k \frac{z_n^k}{2\sigma^2} (x_n - \mu_k)^2 + C\end{aligned}$$

- MLE

$$\begin{aligned}\hat{\pi}_{k,MLE} &= \arg \max_{\pi} \mathcal{L}(\theta; D), \\ \hat{\mu}_{k,MLE} &= \arg \max_{\mu} \mathcal{L}(\theta; D) \\ \hat{\sigma}_{k,MLE} &= \arg \max_{\sigma} \mathcal{L}(\theta; D)\end{aligned} \quad \Rightarrow \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$
- What is we do not know z_n ?



Eric Xing

13

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.

Eric Xing

14

K-means

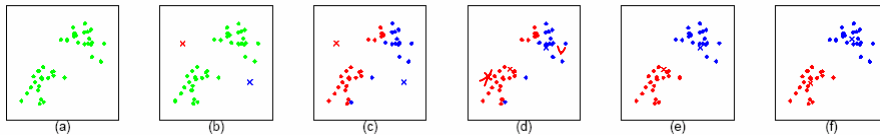
- Start:
 - "Guess" the centroid μ_k and covariance Σ_k of each of the K clusters
- Loop
 - For each point $n=1$ to N ,
compute its cluster label:

$$z_n^{(t)} = \arg \max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

- For each cluster $k=1:K$

$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$

$$\Sigma_k^{(t+1)} = \dots$$



Eric Xing

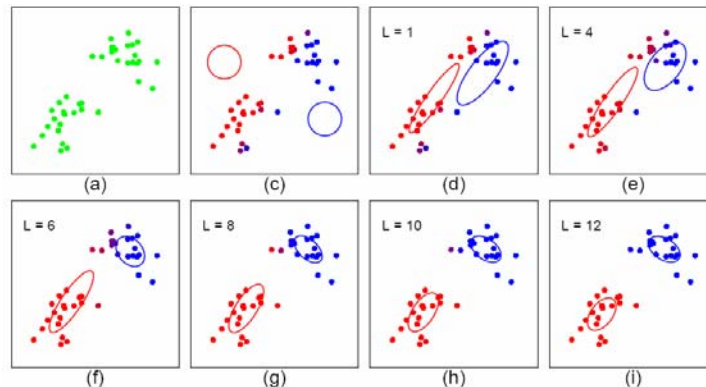
15

Expectation-Maximization

K-me = μ_k

Mixture: $\{\mu_k, \Sigma_k\}$

- Start:
 - "Guess" the centroid μ_k and covariance Σ_k of each of the K clusters
- Loop



Eric Xing

16

Example: Gaussian mixture model



- A mixture of K Gaussians:

- Z is a latent class indicator vector

$$p(z_n) = \text{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

- X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n | z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{-\frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)\right\}$$

- The likelihood of a sample:

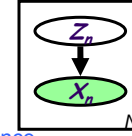
$$\begin{aligned} p(x_n | \mu, \Sigma) &= \sum_k p(z_n^k = 1 | \pi) p(x_n | z_n^k = 1, \mu, \Sigma) \\ &= \sum_{z_n} \prod_k (\pi_k)^{z_n^k} N(x_n; \mu_k, \Sigma_k) = \sum_k \pi_k N(x_n | \mu_k, \Sigma_k) \end{aligned}$$

- The “complete” likelihood

$$p(x_n, z_n^k = 1 | \mu, \Sigma) = p(z_n^k = 1 | \pi) p(x_n | z_n^k = 1, \mu, \Sigma) = \pi_k N(x_n | \mu_k, \Sigma_k)$$

$$p(x_n, z_n | \mu, \Sigma) = \prod_k [\pi_k N(x_n | \mu_k, \Sigma_k)]^{z_n^k}$$

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



Eric Xing

17

Example: Gaussian mixture model

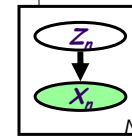


- The complete log likelihood:

$$\begin{aligned} \langle \ell(\theta; D) \rangle_{p(z)} &= \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \Sigma) \\ &= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \Sigma_k)^{z_n^k} \\ &= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k \frac{1}{2\sigma^2} (x_n - \mu_k)^2 + C \end{aligned}$$

- The expected complete log likelihood

$$\begin{aligned} \langle \ell_c(\theta; x, z) \rangle &= \sum_n \langle \log p(z_n | \pi) \rangle_{p(z|x)} + \sum_n \langle \log p(x_n | z_n, \mu, \Sigma) \rangle_{p(z|x)} \\ &= \sum_n \sum_k \langle z_n^k \rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \langle z_n^k \rangle (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log |\Sigma_k| + C \end{aligned}$$

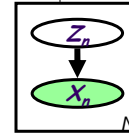


Eric Xing

18

E-step

- We maximize $\langle \ell_c(\theta) \rangle$ iteratively using the following iterative procedure:



- Expectation step:** computing the expected value of the sufficient statistics of the hidden variables (i.e., z) given current est. of the parameters (i.e., π and μ).

$$\tau_n^{k(t)} = \langle z_n^k \rangle_{q^{(t)}} = p(z_n^k = 1 | x, \mu^{(t)}, \Sigma^{(t)}) = \frac{\pi_k^{(t)} \mathcal{N}(x_n | \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_i \pi_i^{(t)} \mathcal{N}(x_n | \mu_i^{(t)}, \Sigma_i^{(t)})}$$

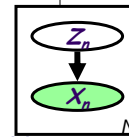
- Here we are essentially doing **inference** $\begin{pmatrix} p(z_n^1 = 1 | x) \\ p(z_n^2 = 1 | x) \\ \vdots \end{pmatrix}$ ~~$p(\mu)$~~

Eric Xing

19

M-step

- We maximize $\langle \ell_c(\theta) \rangle$ iteratively using the following iterative procedure:



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

$$\pi_k^* = \arg \max \langle \ell_c(\theta) \rangle, \quad \Rightarrow \quad \frac{\partial}{\partial \pi_k} \langle \ell_c(\theta) \rangle = 0, \quad \forall k, \quad \text{s.t.} \quad \sum_k \pi_k = 1$$

$$\Rightarrow \pi_k^* = \frac{\sum_n \tau_n^{k(t)} \langle z_n^k \rangle_{q^{(t)}}}{\sum_n \tau_n^{k(t)}} \bigg/ \frac{\sum_n \tau_n^{k(t)}}{\sum_n \tau_n^{k(t)}} = \frac{\langle n_k \rangle}{N}$$

$$\mu_k^* = \arg \max \langle \ell_c(\theta) \rangle, \quad \Rightarrow \quad \mu_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} x_n}{\sum_n \tau_n^{k(t)}}$$

$$\Sigma_k^* = \arg \max \langle \ell_c(\theta) \rangle, \quad \Rightarrow \quad \Sigma_k^{(t+1)} = \frac{\sum_n \tau_n^{k(t)} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_n \tau_n^{k(t)}}$$

Fact:

$$\frac{\partial \log |A^{-1}|}{\partial A^{-1}} = A^T$$

$$\frac{\partial x^T A x}{\partial A} = x x^T$$

- This is isomorphic to **MLE** except that the variables that are hidden are replaced by their expectations (in general they will be replaced by their corresponding "**sufficient statistics**")

Eric Xing

20

Compare: K-means

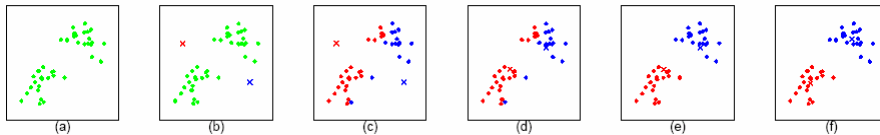


- The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.
- In the K-means "E-step" we do hard assignment:

$$z_n^{(t)} = \arg \max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

- 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:

$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$



Eric Xing

21

IRAS Sky Survey Atlas



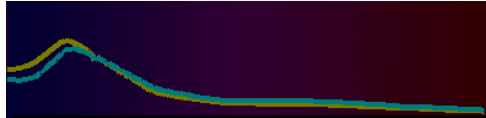
Eric Xing

22

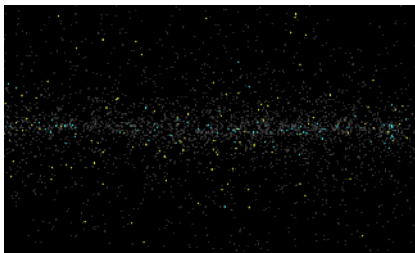
AutoClass Discovery in the IRAS Star Atlas



- From subtle differences between their infrared spectra, two subgroups of stars were distinguished, where previously no difference was suspected.



- The difference is confirmed by looking at their positions on this map of the galaxy.



Eric Xing

23

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?

Eric Xing

24

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

$$\ell_c(\theta; X, Z) \stackrel{\text{def}}{=} \log p(X, Z | \theta)$$

- Usually, optimizing $\ell_c()$ 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, $\ell_c()$ is a random quantity, cannot be maximized directly.**

- Incomplete log likelihood

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

$$\ell_c(\theta; X) = \log p(X | \theta) = \log \sum_z p(X, z | \theta)$$

- This objective won't decouple**

Eric Xing

25

Expected Complete Log Likelihood



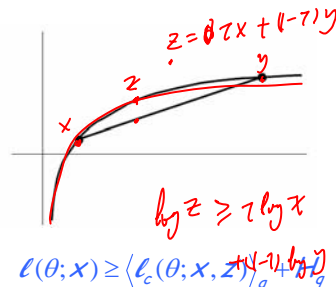
- For **any** distribution $q(z)$, define **expected complete log likelihood**:

$$\langle \ell_c(\theta; X, Z) \rangle_q \stackrel{\text{def}}{=} \sum_z q(z | X, \theta) \log p(X, z | \theta)$$

- A deterministic function of θ
- Linear in $\ell_c()$ --- inherit its factorizability
- Does maximizing this surrogate yield a maximizer of the likelihood?

- Jensen's inequality

$$\begin{aligned} \ell(\theta; X) &= \log p(X | \theta) \\ &= \log \sum_z p(X, z | \theta) \\ &= \log \sum_z q(z | X) \left(\frac{p(X, z | \theta)}{q(z | X)} \right) \\ &\geq \sum_z q(z | X) \log \frac{p(X, z | \theta)}{q(z | X)} \Rightarrow \ell(\theta; X) \geq \langle \ell_c(\theta; X, Z) \rangle_q + \text{KL}(q \| p) \\ &= \sum_z q(z | X) (\log p(X, z | \theta) - \log q(z | X)) \end{aligned}$$



Eric Xing

26

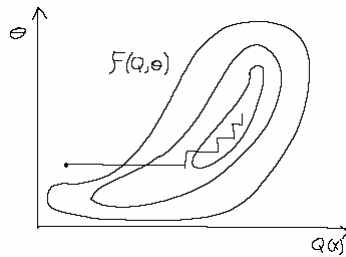
Lower Bounds and Free Energy

- For fixed data x , define a functional called the free energy:

$$F(q, \theta) \stackrel{\text{def}}{=} \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)} \leq \ell(\theta; x)$$

- The EM algorithm is coordinate-ascent on F :

- E-step:** $q^{t+1} = \arg \max_q F(q, \theta^t)$
- M-step:** $\theta^{t+1} = \arg \max_{\theta} F(q^{t+1}, \theta)$



Eric Xing

27

E-step: maximization of expected ℓ_c w.r.t. q

- Claim:

$$q^{t+1} = \arg \max_q F(q, \theta^t) = p(z | x, \theta^t) \quad F(q, \theta^t) \leq \ell(\theta^t; x)$$

- 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 $\ell(\theta, x) \geq F(q, \theta)$

$$\begin{aligned} F(p(z | x, \theta^t), \theta^t) &= \sum_z p(z | x, \theta^t) \log \frac{p(x, z | \theta^t)}{p(z | x, \theta^t)} \\ &= \sum_z \cancel{p(z | x, \theta^t)} \log p(x | \theta^t) \\ &= \log p(x | \theta^t) = \ell(\theta^t; x) \end{aligned}$$

- Can also show this result using variational calculus or the fact that

$$\ell(\theta; x) - F(q, \theta) = \text{KL}(q \| p(z | x, \theta))$$

Eric Xing

28

E-step \equiv plug in posterior expectation of latent variables



- Without loss of generality: assume that $p(\mathbf{x}, \mathbf{z} | \theta)$ is a generalized exponential family distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = \frac{1}{Z(\theta)} h(\mathbf{x}, \mathbf{z}) \exp \left\{ \sum_i \theta_i f_i(\mathbf{x}, \mathbf{z}) \right\}$$

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

- The expected complete log likelihood under $q^{t+1} = p(\mathbf{z} | \mathbf{x}, \theta^t)$ is

$$\begin{aligned} \langle \ell_c(\theta^t; \mathbf{x}, \mathbf{z}) \rangle_{q^{t+1}} &= \sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}, \theta^t) \log p(\mathbf{x}, \mathbf{z} | \theta^t) - A(\theta) \\ &= \sum_i \theta_i^t \langle f_i(\mathbf{x}, \mathbf{z}) \rangle_{q(\mathbf{z} | \mathbf{x}, \theta^t)} - A(\theta) \\ &\stackrel{p\text{-GLIM}}{=} \sum_i \theta_i^t \langle \eta_i(\mathbf{z}) \rangle_{q(\mathbf{z} | \mathbf{x}, \theta^t)} \xi_i(\mathbf{x}) - A(\theta) \end{aligned}$$

Eric Xing

29

M-step: maximization of expected ℓ_c w.r.t. θ



- Note that the free energy breaks into two terms:

$$\begin{aligned} F(q, \theta) &= \sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{z} | \theta)}{q(\mathbf{z} | \mathbf{x})} \\ &= \sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log p(\mathbf{x}, \mathbf{z} | \theta) - \sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log q(\mathbf{z} | \mathbf{x}) \\ &= \langle \ell_c(\theta; \mathbf{x}, \mathbf{z}) \rangle_q + H_q \end{aligned}$$

- The first term is the expected complete log likelihood (energy) and the second term, which does not depend on θ , is the entropy.
- Thus, in the M-step, maximizing with respect to θ for fixed q we only need to consider the first term:

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

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

Eric Xing

30

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. $p(z|x)$
 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.

Eric Xing

31

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.

Eric Xing

32

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

