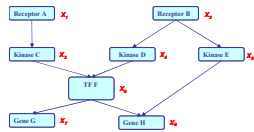# Learning Partially Observed GM: the Expectation-Maximization algorithm

Probabilistic Graphical Models (10-708)

Lecture 11, Oct 24, 2007

Eric Xing

Reading: J-Chap. 10,11; KF-Chap. 17

1

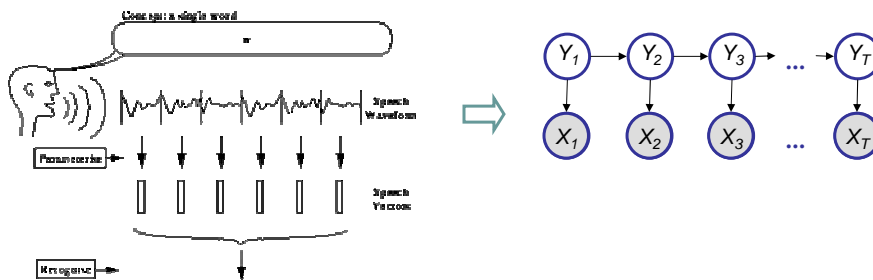# Partially observed GMs

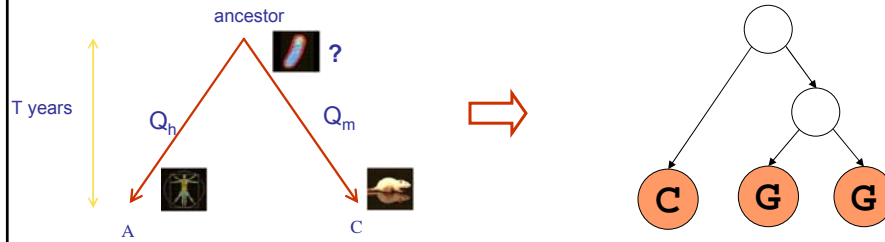- Speech recognition

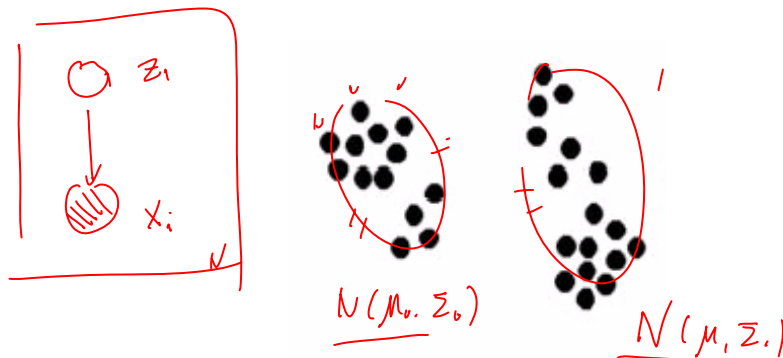Fig. 1.2 Isolated Word Problem

1

Partially observed GM

- Biological Evolution

T years

ancestor ?

Q_h Q_m

A   C

C   G   G

Mixture Models

z_i

x_i

N

$N(\mu_0, \Sigma_0)$

$N(\mu_1, \Sigma_1)$

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).

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, etc.
- Discrete latent variables can be used to partition/cluster data into sub-groups.
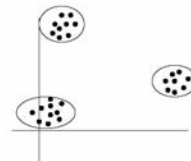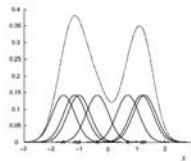- Continuous latent variables (factors) can be used for dimensionality reduction (factor analysis, etc).

Gaussian Mixture Models (GMMs)

- Consider a mixture of *K* Gaussian components:

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

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.
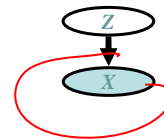

Gaussian Mixture Models (GMMs)

- Consider a mixture of *K* Gaussian components:
  - *Z* is a latent class indicator vector:

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

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

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

  - The likelihood of a sample:

$$\sum_k P(X_n, Z_n)$$

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

mixture proportion   mixture component

$$= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k}\right) = \sum_k \pi_k N(x,|\mu_k,\Sigma_k)$$
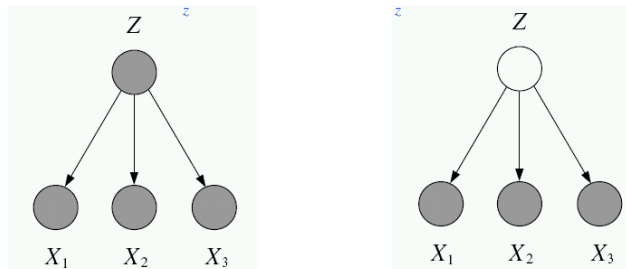
4

Why is Learning Harder?

- In fully observed iid settings, the log likelihood decomposes into a sum of local terms (at least for directed models).

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

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

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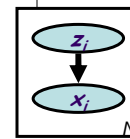

Toward the EM algorithm

- Recall MLE for completely observed data

- Data log-likelihood

$$\ell(\theta; D) = \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n \mid \pi) p(x_n \mid z_n, \mu, \sigma)$$

$$= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k}$$

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

- MLE

$$\hat{\pi}_{k,MLE} = \arg\max_\pi \ell(\theta; D),$$

$$\hat{\mu}_{k,MLE} = \arg\max_\mu \ell(\theta; D) \quad \Rightarrow \quad \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$

$$\hat{\sigma}_{k,MLE} = \arg\max_\sigma \ell(\theta; D)$$

$$\left\langle \frac{\mu}{z_n} \right\rangle$$

- What if we do not know z_n?

Question

- " … We solve problem X using Expectation-Maximization …"
  - What does it mean?

- E
  - What do we take Expectation with?
  - What do we take expectation over?

- M
  - What do we maximize?
  - What do we maximize with respect to?

$$E_{P_{(-)}}(f\ (x\bar{y}))$$

$$(z) \quad \langle zx \rangle$$

$$p\left(z \mid \theta^{(t)}\right)$$

$$\left\langle \log P(x, \underline{z}) \right\rangle_{q(z)}$$

$$\theta.$$

Recall: K-means
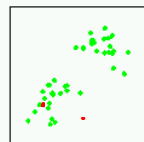
$$p(z \mid x, \theta^T) \implies \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$$

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

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

(a)   (b)   (c)   (d)   (e)   (f)
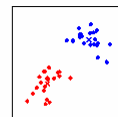
Expectation-Maximization

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

(handwritten annotations: $\binom{v.4}{v.6}$, soft assignment)

(a)   (c)   (d)   (e)
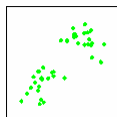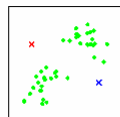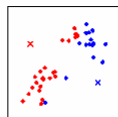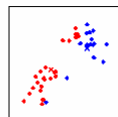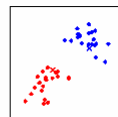
(f)   (g)   (h)   (i)
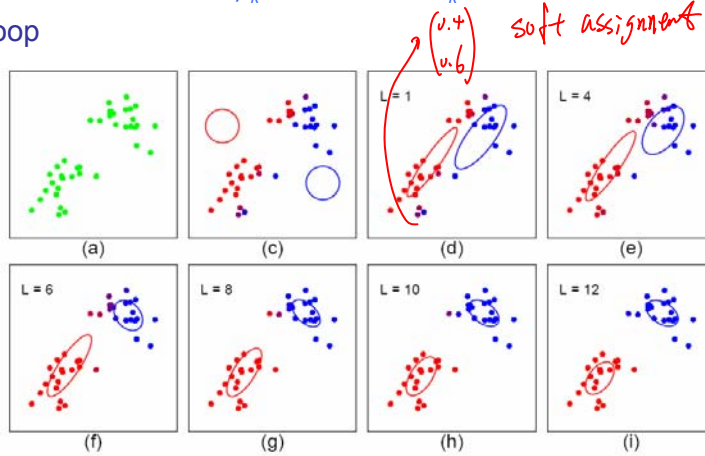
Example: Gaussian mixture model

- 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 class-specific mean/covariance
    $$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2}|\Sigma_k|^{1/2}} \exp\left\{-\tfrac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right\}$$
  - The likelihood of a sample:
    $$p(x_n \mid \mu, \Sigma) = \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma)$$
    $$= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k}\right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k)$$
- The expected complete log likelihood

$$\left\langle \ell_c(\theta; x, z) \right\rangle_p = \sum_n \left\langle \log p(z_n \mid \pi) \right\rangle_{p(z \mid x)} + \sum_n \left\langle \log p(x_n \mid z_n, \mu, \Sigma) \right\rangle_{p(z \mid x)}$$

(handwritten: $p(z \mid x)$, $\gamma(z)$)

$$= \sum_n \sum_k \left\langle z_n^k \right\rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \left\langle z_n^k \right\rangle \left((x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k) + \log|\Sigma_k| + C\right)$$

7
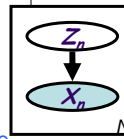
E-step

- We maximize $\langle l_c(\theta) \rangle$ 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).

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

  - Here we are essentially doing **inference**

Recall out objective

- The expected complete log likelihood

$$\left\langle \ell_c(\theta; x, z) \right\rangle = \sum_n \left\langle \log p(z_n \mid \pi) \right\rangle_{p(z|x)} + \sum_n \left\langle \log p(x_n \mid z_n, \mu, \Sigma) \right\rangle_{p(z|x)}$$

$$= \sum_n \sum_k \left\langle z_n^k \right\rangle \log \pi_k - \frac{1}{2} \sum_n \sum_k \left\langle z_n^k \right\rangle \left((x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) + \log|\Sigma_k| + C \right)$$

M-step

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

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

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

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

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

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

Fact:

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

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

- 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 and EM

The EM algorithm for mixtures of Gaussians is like a "soft version" of the K-means algorithm.

- K-means
  - 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:

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

- EM
  - E-step

$$\tau_n^{k(t)} = \langle z_n^k \rangle_{q^{(t)}}$$

$$= p(z_n^k = 1 \mid x, \mu^{(t)}, \Sigma^{(t)}) = \frac{\pi_k^{(t)} N(x_n, \mid \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_i \pi_i^{(t)} N(x_n, \mid \mu_i^{(t)}, \Sigma_i^{(t)})}$$

  - M-step

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

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

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

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

  $$\ell_c(\theta; x, z) \overset{\text{def}}{=} \log p(x, z \mid \theta)$$

  - 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:

  $$\ell_c(\theta; x) = \log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta)$$

  - **This objective won't decouple**

Expected Complete Log Likelihood

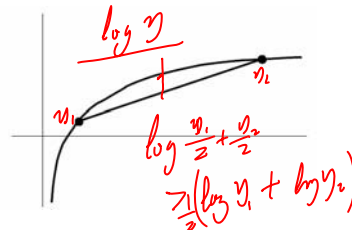- For *any* distribution $q(z)$, define *expected complete log likelihood*:

$$\left\langle \ell_c(\theta; x, z) \right\rangle_q \overset{\text{def}}{=} \sum_z q(z \mid x, \theta) \log p(x, z \mid \theta)$$

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

- Jensen's inequality

$$\ell(\theta; x) = \log p(x \mid \theta)$$
$$= \log \sum_z p(x, z \mid \theta)$$
$$= \log \sum_z q(z \mid x) \left(\frac{p(x, z \mid \theta)}{q(z \mid x)} \right)$$
$$\geq \sum_z q(z \mid x) \log \frac{p(x, z \mid \theta)}{q(z \mid x)} \quad \Rightarrow \quad \ell(\theta; x) \geq \left\langle \ell_c(\theta; x, z) \right\rangle_q + H_q$$

Lower Bounds and Free Energy
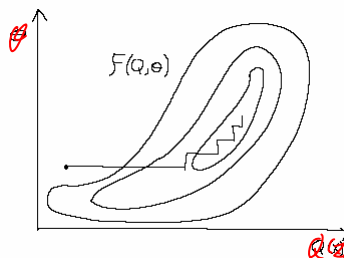
- For fixed data x, define a functional called the free energy:

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

- The EM algorithm is coordinate-ascent on F:

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

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

11

E-step: maximization of expected ℓ_c 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 classification).

- Proof (easy): this setting attains the bound $\ell(\theta, x) \geq F(q, \theta)$

$$F(p(z \mid x, \theta^t), \theta^t) = \sum_z p(z \mid x, \theta^t) \log \frac{p(x, z \mid \theta^t)}{p(z \mid x, \theta^t)}$$

$$= \sum_z q(z \mid x) \log p(x \mid \theta^t)$$

$$= \log p(x \mid \theta^t) = \ell(\theta^t; x)$$

- Can also show this result using variational calculus or the fact that $\ell(\theta; x) - F(q, \theta) = \mathrm{KL}\big(q \,\|\, p(z \mid x, \theta)\big)$

E-step ≡ plug in posterior expectation of latent variables

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

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

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

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

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

$$= \sum_i \theta_i^t \langle f_i(x, z) \rangle_{q(z \mid x, \theta^t)} - A(\theta)$$

$$\overset{p \sim \text{GLIM}}{=} \sum_i \theta_i^t \langle \eta_i(z) \rangle_{q(z \mid x, \theta^t)} \xi_i(x) - A(\theta)$$

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

- Note that the free energy breaks into two terms:

$$F(q,\theta) = \sum_z q(z\mid x)\log\frac{p(x,z\mid\theta)}{q(z\mid x)}$$
$$= \sum_z q(z\mid x)\log p(x,z\mid\theta) - \sum_z q(z\mid x)\log q(z\mid x)$$
$$= \left\langle \ell_c(\theta; x, z)\right\rangle_q + H_q$$

  - 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:

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

  - 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. $p(z\mid x,\theta)$.

Eric Xing 25

Example: HMM

- **Supervised learning**: estimation when the "right answer" is known
  - **Examples:**
    GIVEN: a genomic region $x = x_1\ldots x_{1,000,000}$ where we have good (experimental) annotations of the CpG islands
    GIVEN: the casino player allows us to observe him one evening, as he changes dice and produces 10,000 rolls

- **Unsupervised learning**: estimation when the "right answer" is unknown
  - **Examples:**
    GIVEN: the porcupine genome; we don't know how frequent are the CpG islands there, neither do we know their composition
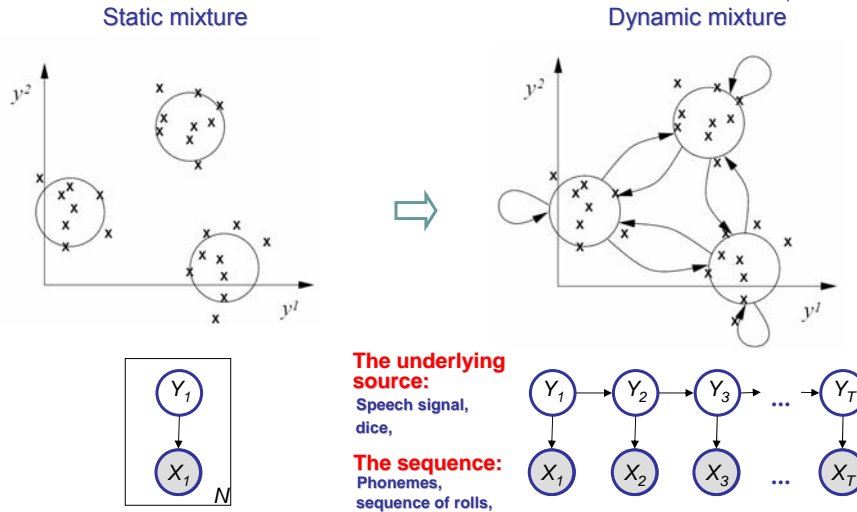   GIVEN: 10,000 rolls of the casino player, but we don't see when he changes dice

- **QUESTION:** Update the parameters θ of the model to maximize $P(x\mid\theta)$ -- Maximal likelihood (ML) estimation

Eric Xing 26

13

Hidden Markov Model:
from static to dynamic mixture models

Static mixture | Dynamic mixture

The underlying source: Speech signal, dice,

The sequence: Phonemes, sequence of rolls,

Eric Xing

The Baum Welch algorithm

$y_t. \forall t.$

$p(y_t | X)$

- The complete log likelihood

$$\ell_c(\theta; \mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}, \mathbf{y}) = \log \prod_n \left(p(y_{n,1}) \prod_{t=2}^{T} p(y_{n,t} \mid y_{n,t-1}) \prod_{t=1}^{T} p(x_{n,t} \mid x_{n,t}) \right)$$

- The expected complete log likelihood

$$\langle \ell_c(\theta; \mathbf{x}, \mathbf{y}) \rangle = \sum_n \left(\langle y_{n,1}^i \rangle_{p(y_{n,1}|\mathbf{x}_n)} \log \pi_i \right) + \sum_n \sum_{t=2}^{T} \left(\langle y_{n,t-1}^i y_{n,t}^j \rangle_{p(y_{n,t-1}, y_{n,t}|\mathbf{x}_n)} \log a_{i,j} \right) + \sum_n \sum_{t=1}^{T} \left(x_{n,t}^k \langle y_{n,t}^i \rangle_{p(y_{n,t}|\mathbf{x}_n)} \log b_{i,k} \right)$$

- EM

  - The **E** step

  $$\gamma_{n,t}^i = \langle y_{n,t}^i \rangle = p(y_{n,t}^i = 1 | \mathbf{x}_n)$$

  $$\xi_{n,t}^{i,j} = \langle y_{n,t-1}^i y_{n,t}^j \rangle = p(y_{n,t-1}^i = 1, y_{n,t}^j = 1 | \mathbf{x}_n)$$

  (handwritten annotations):
  $\# \langle y_{n,t-1}^t y_{n,t}^i \rangle \ni \langle y_{n,t-1}^t \rangle \langle y_{n,t}^j \rangle$

  $l_c(\theta, x, y) = \sum_n \left(y_{n,1}^i \log \pi_i + \sum_{n,t} y_t^i y_{t-1}^i \log a_{ij} + \cdots \right)$

  - The **M** step ("symbolically" identical to MLE)

  $$\pi_i^{ML} = \frac{\sum_n \gamma_{n,1}^i}{N} \qquad a_{ij}^{ML} = \frac{\sum_n \sum_{t=2}^{T} \xi_{n,t}^{i,j}}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i} \qquad b_{ik}^{ML} = \frac{\sum_n \sum_{t=1}^{T} \gamma_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i}$$

Eric Xing

Unsupervised ML estimation

- Given $x = x_1 \ldots x_N$ for which the true state path $y = y_1 \ldots y_N$ is unknown,

  - **EXPECTATION MAXIMIZATION**

  0. Starting with our best guess of a model M, parameters θ:

  1. Estimate A_{ij}, B_{ik} in the training data
     - How? $A_{ij} = \sum_{n,t} \langle y_{n,t-1}^i y_{n,t}^j \rangle$, $B_{ik} = \sum_{n,t} \langle y_{n,t}^i \rangle x_{n,t}^k$,
  2. Update θ according to A_{ij}, B_{ik}
     - Now a "supervised learning" problem
  3. Repeat 1 & 2, until convergence

  This is called the Baum-Welch Algorithm

  We can get to a provably more (or equally) likely parameter set θ each iteration

---

EM for general BNs

while not converged
    % E-step
    for each node i
       $ESS_i = 0$ % reset expected sufficient statistics
    for each data sample n
       do inference with $X_{n,H}$
       for each node i
          $ESS_i += \left\langle SS_i(x_{n,i}, x_{n,\pi_i}) \right\rangle_{p(x_{n,H} | x_{n,-H})}$
    % M-step
   for each node i
       $\theta_i := \text{MLE}(ESS_i)$

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

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

Conditional mixture model: Mixture of experts
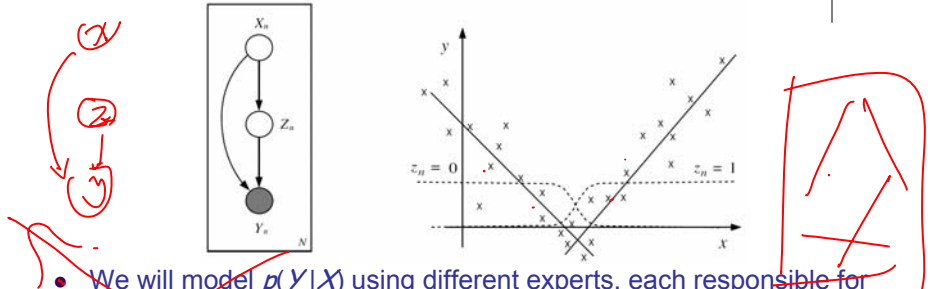
- We will model $p(Y|X)$ using different experts, each responsible for different regions of the input space.

  - Latent variable Z chooses expert using softmax gating function:
    $$P(z^k = 1|x) = \text{Softmax}(\xi^T x)$$
  - Each expert can be a linear regression model: $P(y|x, z^k = 1) = \mathcal{N}(y; \theta_k^T x, \sigma_k^2)$
  - The posterior expert responsibilities are
    $$P(z^k = 1|x, y, \theta) = \frac{p(z^k = 1|x)p_k(y|x, \theta_k, \sigma_k^2)}{\sum_j p(z^j = 1|x)p_j(y|x, \theta_j, \sigma_j^2)}$$

EM for conditional mixture model

- Model:

$$P(y|x) = \sum_k p(z^k = 1 | x, \xi) p(y | z^k = 1, x, \theta_i, \sigma)$$

- The objective function

$$\langle \ell_c(\theta; x, y, z) \rangle = \sum_n \langle \log p(z_n | x_n, \xi) \rangle_{p(z|x,y)} + \sum_n \langle \log p(y_n | x_n, z_n, \theta, \sigma) \rangle_{p(z|x,y)}$$

$$= \sum_n \sum_k \langle z_n^k \rangle \log\left(\text{softmax}(\xi_k^T x_n)\right) - \frac{1}{2} \sum_n \sum_k \langle z_n^k \rangle \left(\frac{(y_n - \theta_k^T x_n)}{\sigma_k^2} + \log \sigma_k^2 + C \right)$$

- EM:

  - E-step: $\tau_n^{k(t)} = P(z_n^k = 1 | x_n, y_n, \theta) = \dfrac{p(z_n^k = 1 | x_n) p_k(y_n | x_n, \theta_k, \sigma_k^2)}{\sum_j p(z_n^j = 1 | x_n) p_j(y_n | x_n, \theta_j, \sigma_j^2)}$
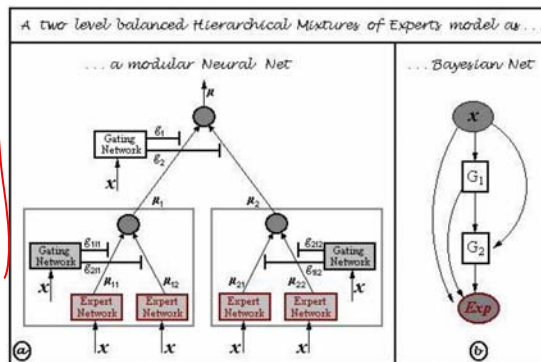
  - M-step:
    - using the normal equation for standard LR $\theta = (X^T X)^{-1} X^T Y$, but with the data re-weighted by τ (homework)
    - IRLS and/or weighted IRLS algorithm to update $\{\xi_k, \theta_k, \sigma_k\}$ based on data pair (x_n, y_n), with weights $\tau_n^{k(t)}$ (homework?)
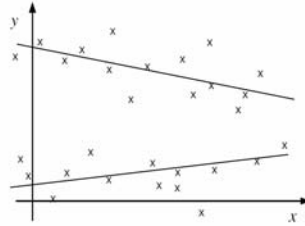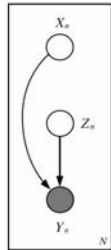
Hierarchical mixture of experts

- This is like a soft version of a depth-2 classification/regression tree.
- $P(Y|X, G_1, G_2)$ can be modeled as a GLIM, with parameters dependent on the values of G_1 and G_2 (which specify a "conditional path" to a given leaf in the tree).

Mixture of overlapping experts

- By removing the $X \rightarrow Z$ arc, we can make the partitions independent of the input, thus allowing overlap.
- This is a mixture of linear regressors; each subpopulation has a different conditional mean.

$$P(z^k = 1 | x, y, \theta) = \frac{p(z^k = 1) p_k(y | x, \theta_k, \sigma_k^2)}{\sum_j p(z^j = 1) p_j(y | x, \theta_j, \sigma_j^2)}$$

Partially Hidden Data

- Of course, we can learn when there are missing (hidden) variables on some cases and not on others.
- In this case the cost function is:

$$\ell_c(\theta; D) = \sum_{n \in \text{Complete}} \log p(x_n, y_n | \theta) + \sum_{m \in \text{Missing}} \log \sum_{y_m} p(x_m, y_m | \theta)$$

  - Note that y_m do not have to be the same in each case --- the data can have different missing values in each different sample

- Now you can think of this in a new way: in the E-step we estimate the hidden variables on the incomplete cases only.
- The M-step optimizes the log likelihood on the complete data plus the expected likelihood on the incomplete data using the E-step.

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