

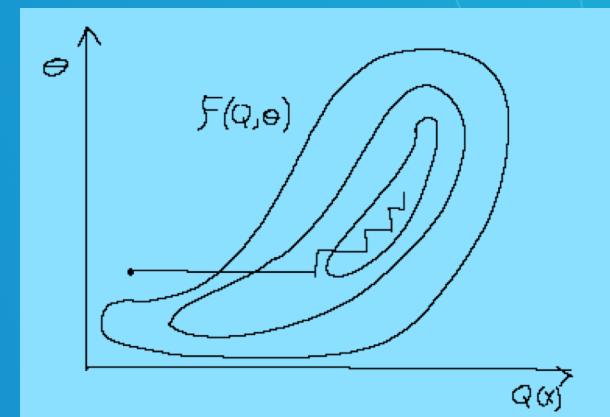
Probabilistic Graphical Models

Learning Partially Observed GM:
the Expectation-Maximization algorithm

Maruan Al-Shedivat

Lecture 6, February 4, 2019

Reading: see class homepage





Before we start, some logistics...

- ❑ HW1 was released last week
 - ❑ **Due date:** Wed, Feb 13 (start early!)
- ❑ Project:
 - ❑ Form teams of 3-4 people (hurry up if you haven't found the team yet)
 - ❑ TAs have posted some project suggestions on the course page
 - ❑ Project proposals are due Feb 22
- ❑ Lecture 5 → self-study (skipping in the interest of time)
 - ❑ Some brief highlights will be covered today
 - ❑ The scribe team is to write notes based on slides + reading material





Recall: Learning Graphical Models

- Scenarios:
 - Completely observed GMs
 - directed
 - undirected
 - Partially or unobserved GMs
 - directed
 - undirected (an open research topic)
- Estimation principles:
 - Maximal likelihood estimation (MLE)
 - Maximal conditional likelihood
 - Bayesian estimation
 - Maximal "Margin"
 - Maximum entropy
- We use **learning** as a name for the process of **estimating the parameters**, and in some cases, the topology of the network, from data.

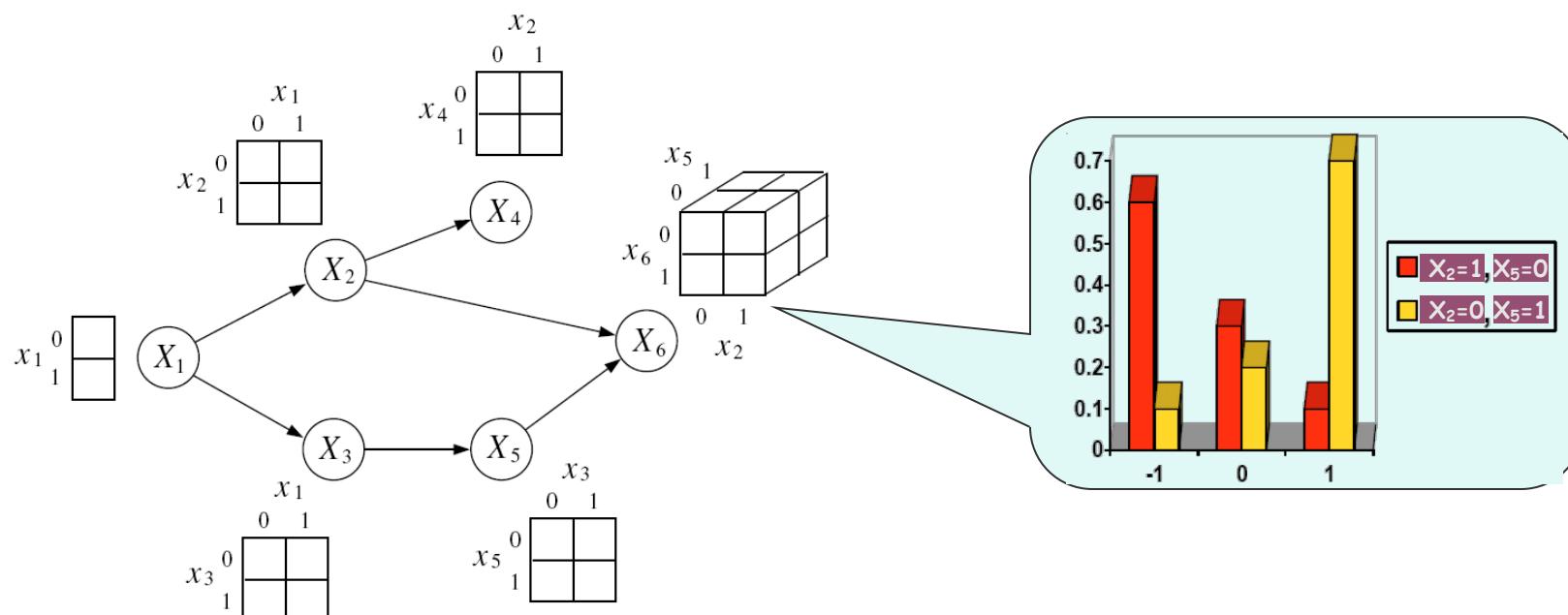




MLE for general BNs

- If we assume the parameters for each CPD are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node:

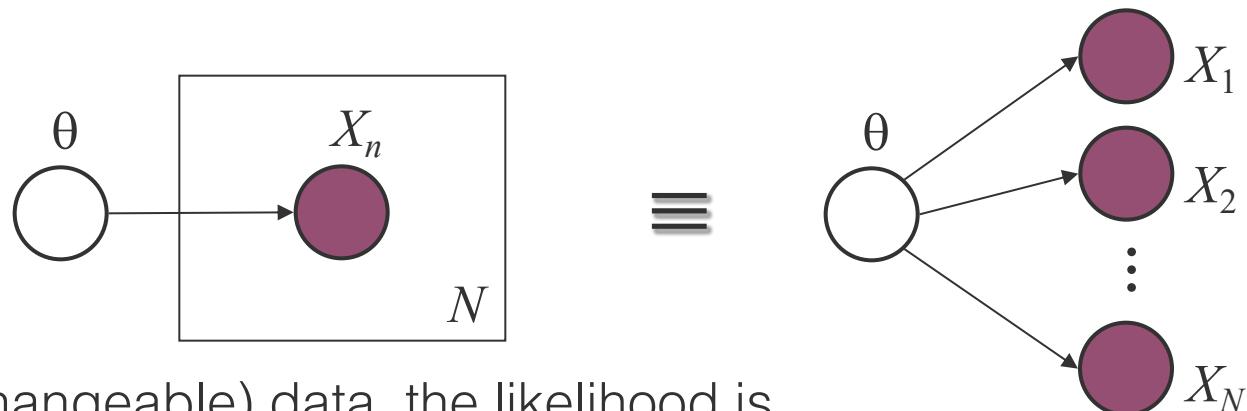
$$\ell(\theta; D) = \log p(D | \theta) = \log \prod_n \left(\prod_i p(x_{n,i} | \mathbf{x}_{n,\pi_i}, \theta_i) \right) = \sum_i \left(\sum_n \log p(x_{n,i} | \mathbf{x}_{n,\pi_i}, \theta_i) \right)$$





Plates

- A plate is a “macro” that allows subgraphs to be replicated



- For iid (exchangeable) data, the likelihood is

$$p(D|\theta) = \prod_n p(x_n|\theta)$$

- We can represent this as a Bayes net with N nodes.
 - The rules of plates are simple: repeat every structure in a box a number of times given by the integer in the corner of the box (e.g. N), updating the plate index variable (e.g. n) as you go.
 - Duplicate every arrow going into the plate and every arrow leaving the plate by connecting the arrows to each copy of the structure.



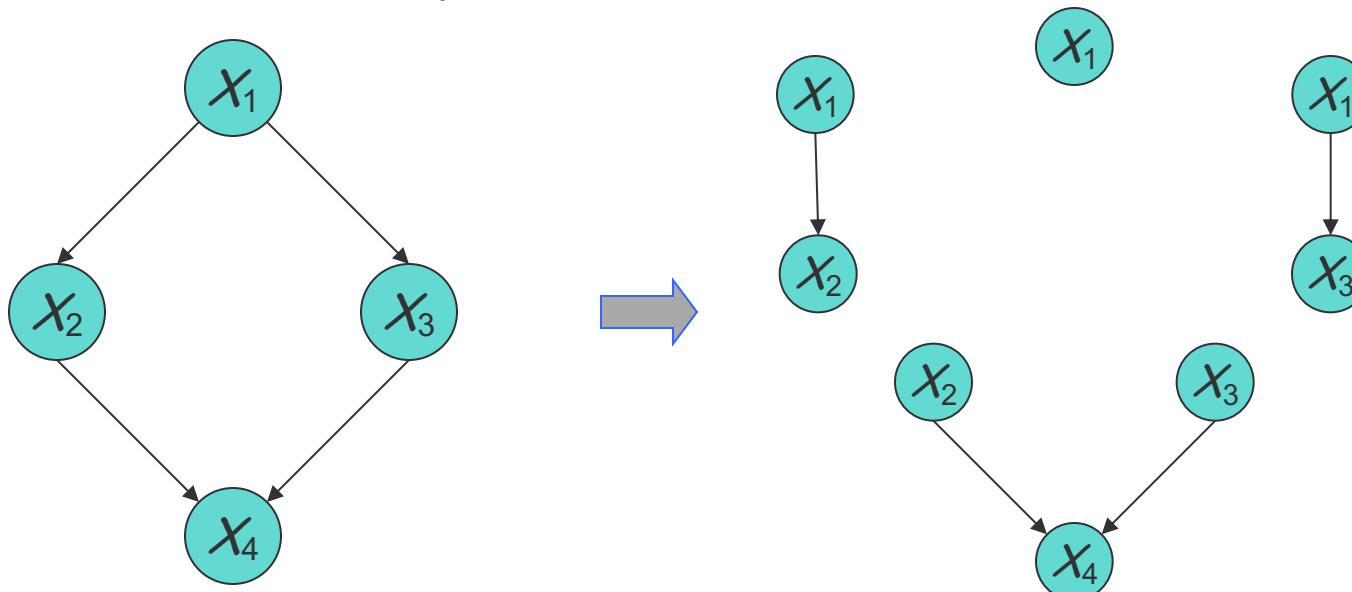


Decomposable likelihood of a BN

- Consider the distribution defined by the directed acyclic GM:

$$p(x|\theta) = p(x_1|\theta_1)p(x_2|x_1,\theta_2)p(x_3|x_1,\theta_3)p(x_4|x_2,x_3,\theta_4)$$

- This is exactly like learning four separate small BNs, each of which consists of a node and its parents.

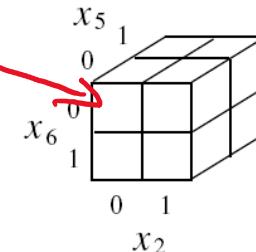




MLE for BNs with tabular CPDs

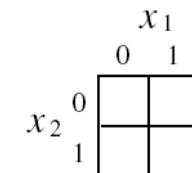
- Assume each CPD is represented as a table (multinomial) where

$$\theta_{ijk} \stackrel{\text{def}}{=} p(X_i = j | X_{\pi_i} = k)$$



- Note that in case of multiple parents, X_{π_i} will have a composite state, and the CPD will be a high-dimensional table
- The sufficient statistics are counts of family configurations

$$n_{ijk} \stackrel{\text{def}}{=} \sum_n x_{n,i}^j x_{n,\pi_i}^k$$



- The log-likelihood is

$$\ell(\theta; D) = \log \prod_{i,j,k} \theta_{ijk}^{n_{ijk}} = \sum_{i,j,k} n_{ijk} \log \theta_{ijk}$$

- Using a Lagrange multiplier to enforce $\sum_j \theta_{ijk} = 1$, we get:

$$\theta_{ijk}^{ML} = \frac{n_{ijk}}{\sum_{j'} n_{ij'k}}$$





Example: HMM: two scenarios

- Supervised learning: estimation when the “right answer” is known
 - Examples:
GIVEN: a genomic region $x = x_1 \dots x_{1,000,000}$ 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 $P(x|\theta)$ --- Maximal likelihood (ML) estimation





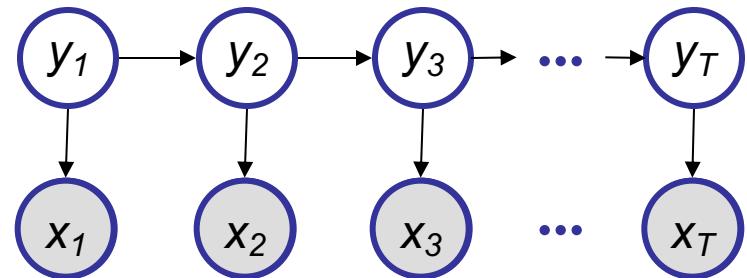
Recall definition of HMM

- Transition probabilities between any two states

$$p(y_t^j = 1 | y_{t-1}^i = 1) = a_{i,j},$$

or

$$p(y_t | y_{t-1}^i = 1) \sim \text{Multinomial}(a_{i,1}, a_{i,2}, \dots, a_{i,M}), \forall i \in I.$$



- Start probabilities

$$p(y_1) \sim \text{Multinomial}(\pi_1, \pi_2, \dots, \pi_M).$$

- Emission probabilities associated with each state

$$p(x_t | y_t^i = 1) \sim \text{Multinomial}(b_{i,1}, b_{i,2}, \dots, b_{i,K}), \forall i \in I.$$

or in general:

$$p(x_t | y_t^i = 1) \sim f(\cdot | \theta_i), \forall i \in I.$$





Supervised ML estimation

- Given $x = x_1 \dots x_N$ for which the true state path $y = y_1 \dots y_N$ is known,
 - Define:

A_{ij} = # times state transition $i \rightarrow j$ occurs in y

B_{ik} = # times state i in y emits k in x

- We can show that the **maximum likelihood** parameters θ are:

$$a_{ij}^{ML} = \frac{\#(i \rightarrow j)}{\#(i \rightarrow \bullet)} = \frac{\sum_n \sum_{t=2}^T y_{n,t-1}^i y_{n,t}^j}{\sum_n \sum_{t=2}^T y_{n,t-1}^i} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$

$$b_{ik}^{ML} = \frac{\#(i \rightarrow k)}{\#(i \rightarrow \bullet)} = \frac{\sum_n \sum_{t=1}^T y_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^T y_{n,t}^i} = \frac{B_{ik}}{\sum_{k'} B_{ik'}}$$

- What if x is continuous? We can treat $\{(x_{n,t}, y_{n,t}): t=1:T, n=1:N\}$ as NT observations of, e.g., a Gaussian, and apply learning rules for Gaussian ...





Supervised ML estimation, cont'd

- Intuition:
 - When we know the underlying states, the best estimate of θ is the average frequency of transitions & emissions that occur in the training data
- Drawback:
 - Given little data, there may be overfitting:
 - $P(x|\theta)$ is maximized, but θ is unreasonable: **0 probabilities – VERY BAD**
- Example:
 - Given 10 casino rolls, we observe
 - $x = 2, 1, 5, 6, 1, 2, 3, 6, 2, 3$
 - $y = F, F, F, F, F, F, F, F, F, F$
 - Then: $a_{FF} = 1; a_{FL} = 0$
 $b_{F1} = b_{F3} = .2;$
 $b_{F2} = .3; b_{F4} = 0; b_{F5} = b_{F6} = .1$





Pseudocounts

- Solution for small training sets:
 - Add pseudocounts
$$A_{ij} = \text{\# times state transition } i \rightarrow j \text{ occurs in } \mathbf{y} + R_{ij}$$
$$B_{ik} = \text{\# times state } i \text{ in } \mathbf{y} \text{ emits } k \text{ in } \mathbf{x} + S_{ik}$$
 - R_{ij}, S_{ij} are pseudocounts representing our prior belief
 - Total pseudocounts: $R_i = \sum_j R_{ij}$, $S_i = \sum_k S_{ik}$,
 - --- "strength" of prior belief,
 - --- total number of imaginary instances in the prior
- Larger total pseudocounts \Rightarrow strong prior belief
- Small total pseudocounts: just to avoid 0 probabilities \Rightarrow smoothing
- This is equivalent to Bayesian est. under a uniform prior with "parameter strength" equals to the pseudocounts





Summary: Learning fully observed GMs

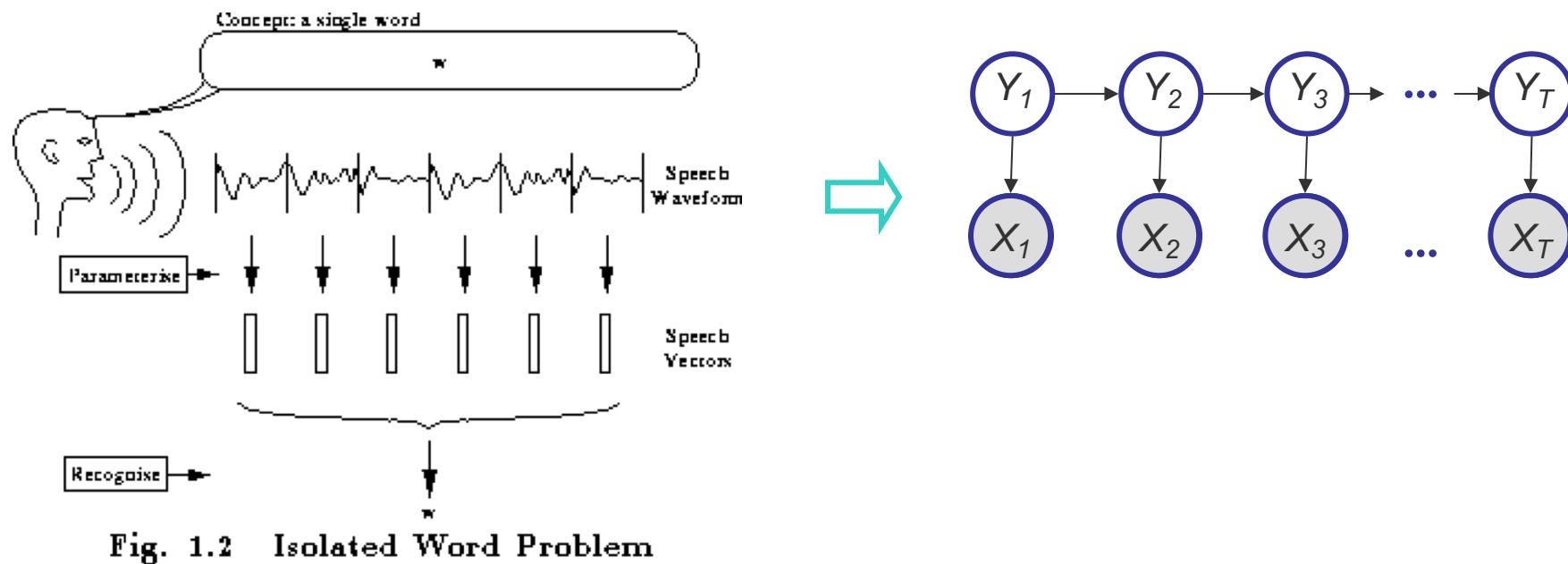
- ❑ For fully observed BN, the log-likelihood function decomposes into a sum of local terms, one per node; thus learning is also factored
- ❑ See lecture 5 for the following:
 - ❑ Structural learning
 - ❑ Chow-Liu algorithm
 - ❑ Neighborhood selection
 - ❑ Learning single-node GM – density estimation: exponential family dist.
 - ❑ Typical discrete distribution
 - ❑ Typical continuous distribution
 - ❑ Conjugate priors
 - ❑ Learning two-node BN: GLIM
 - ❑ Conditional Density Est.
 - ❑ Classification
 - ❑ Learning BN with more nodes
 - ❑ Local operations





Partially observed GMs

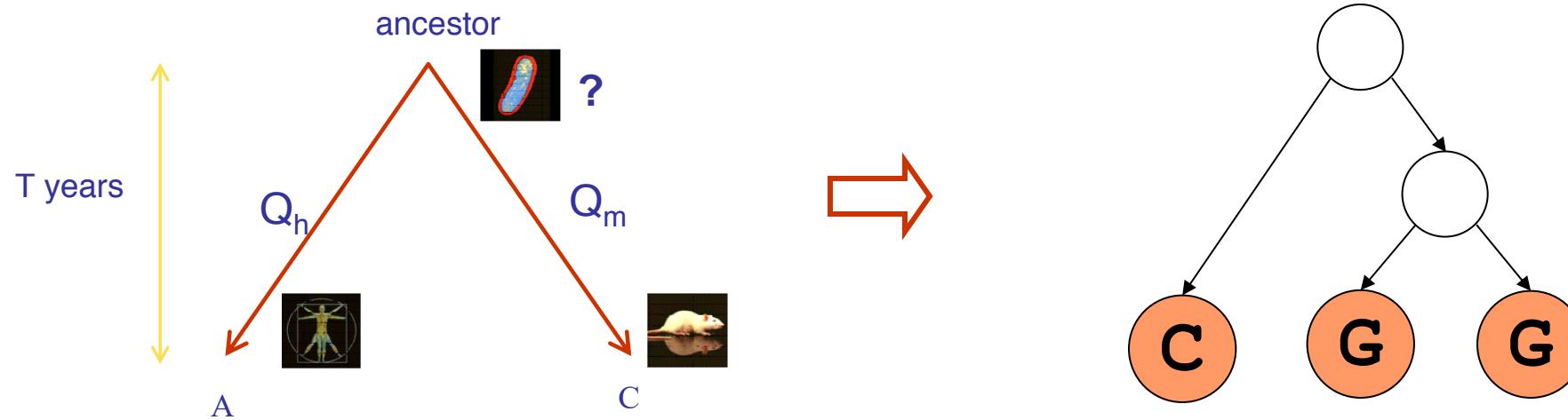
- Speech recognition





Partially observed GM

- Biological Evolution



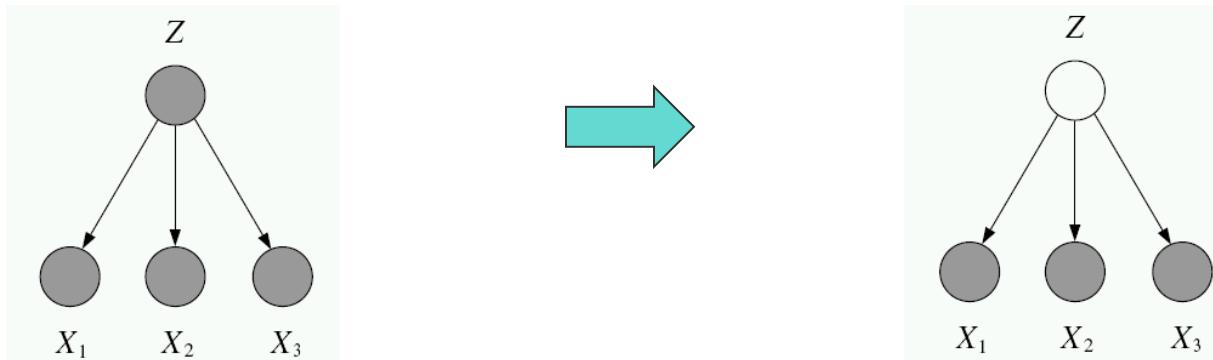


Inference is a subroutine for Learning

- Directed, but partially observed GM: imputing miss variables

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

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



- Undirected, but fully observed GM: compute clique marginals

$$\ell = \sum_c \sum_{\mathbf{x}_c} m(\mathbf{x}_c) \log \psi_c(\mathbf{x}_c) - N \log Z$$



$$\begin{aligned}\frac{\partial \log Z}{\partial \psi_c(\mathbf{x}_c)} &= \frac{1}{Z} \frac{\partial}{\partial \psi_c(\mathbf{x}_c)} \left(\sum_{\tilde{\mathbf{x}}} \prod_d \psi_d(\tilde{\mathbf{x}}_d) \right) \\ &= \frac{1}{Z} \sum_{\tilde{\mathbf{x}}} \delta(\tilde{\mathbf{x}}_c, \mathbf{x}_c) \frac{\partial}{\partial \psi_c(\mathbf{x}_c)} \left(\prod_d \psi_d(\tilde{\mathbf{x}}_d) \right) \\ &= \sum_{\tilde{\mathbf{x}}} \delta(\tilde{\mathbf{x}}_c, \mathbf{x}_c) \frac{1}{\psi_c(\tilde{\mathbf{x}}_c)} \frac{1}{Z} \prod_d \psi_d(\tilde{\mathbf{x}}_d) \\ &= \frac{1}{\psi_c(\mathbf{x}_c)} \sum_{\tilde{\mathbf{x}}} \delta(\tilde{\mathbf{x}}_c, \mathbf{x}_c) p(\tilde{\mathbf{x}}) = \frac{p(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)}\end{aligned}$$





Probabilistic Inference

- We now have compact representations of probability distributions:
Graphical Models
- A GM M describes a unique probability distribution P
- Typical tasks:
 - Task 1: How do we answer **queries** about P_M , e.g., $P_M(X|Y)$?
 - We use **inference** as a name for the process of computing answers to such queries
 - Task 2: How do we estimate a **plausible model** M from data D ?
 - i. We use **learning** as a name for the process of obtaining point estimate of M
 - ii. But for **Bayesian**, they seek $p(M|D)$, which is actually an **inference** problem
 - iii. **Inference** can be a subroutine of many more types of learning problems

To learn partially observable GMs \Rightarrow iterate between Task 1 & 2





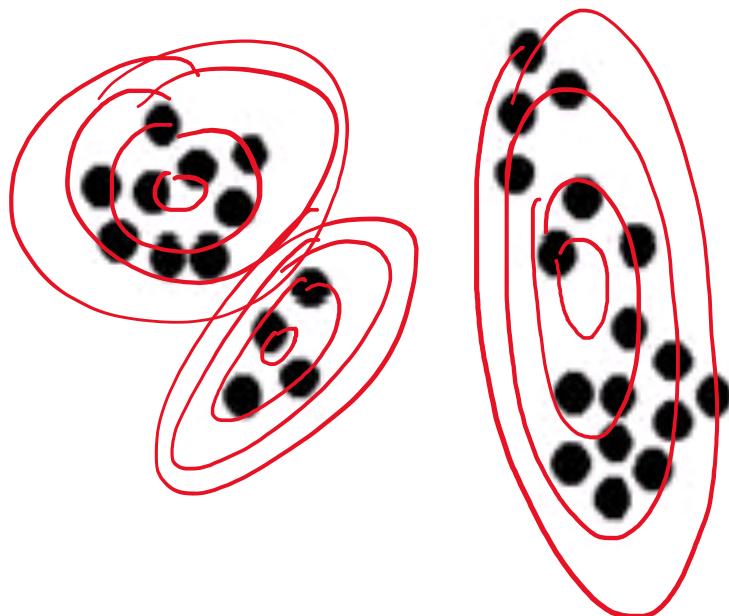
Approaches to inference

- ❑ Exact inference algorithms
 - ❑ The elimination algorithm
 - ❑ Message-passing algorithm (sum-product, belief propagation)
 - ❑ The junction tree algorithms
- ❑ Approximate inference techniques
 - ❑ Stochastic simulation / sampling methods
 - ❑ Markov chain Monte Carlo methods
 - ❑ Variational algorithms





Mixture Models



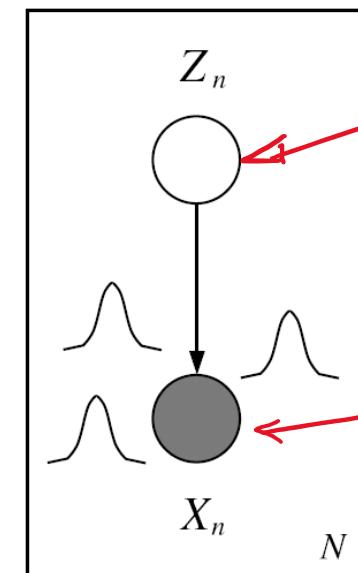
Represent a complex
density through
a mixture of
Simpler densities





Mixture Models, cont'd

- ❑ A density model $p(x)$ may be multi-modal.
- ❑ But we may be able to model it as a mixture of uni-modal distributions (e.g., Gaussians).
- ❑ Example: each mode of the distribution may correspond to a different sub-population (e.g., male and female).



discrete latent
variables
index modes

model intra-
mode distrib's





Unobserved Variables

- ❑ A variable can be unobserved (latent) because:
 1. **imaginary quantity**: meant to provide some simplified and abstractive view of the data generation process
 - ❑ e.g., speech recognition models, mixture models, ...
 2. **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 ...
 3. **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 (e.g., factor analysis, etc.)



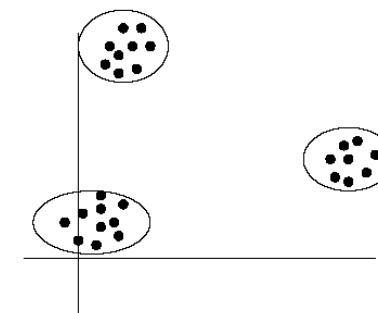
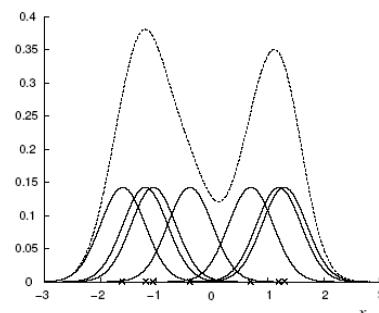


Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

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

↑
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.



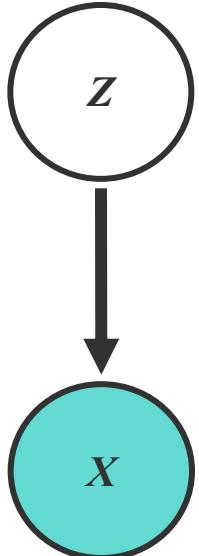


Gaussian Mixture Models (GMMs)

- 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

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

- The likelihood of a sample:

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

mixture component
mixture proportion





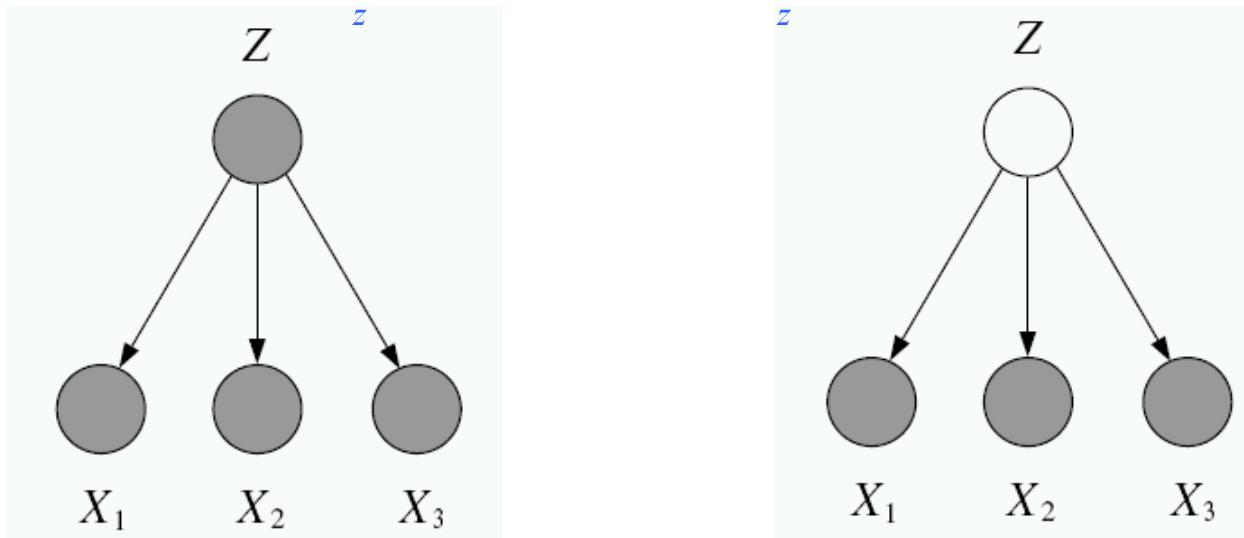
Why is Learning Harder?

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

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

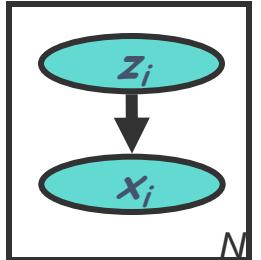
- With latent variables, all the parameters become coupled together via marginalization

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





Toward the EM algorithm



- Recall MLE for completely observed data

- Data log-likelihood

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

- MLE

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

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

$$\Rightarrow \hat{\mu}_{k,MLE} = \frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$

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

- What if we do not know z_n ?





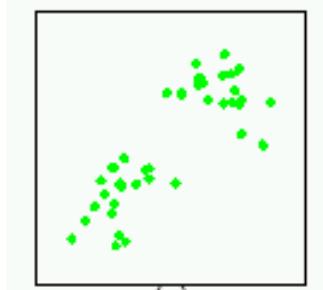
Question

- ❑ “ ... We solve problem X using Expectation-Maximization ... ”
 - ❑ What does it mean?
- ❑ E-step
 - ❑ What do we take expectation with?
 - ❑ What do we take expectation over?
- ❑ M-step
 - ❑ What do we maximize?
 - ❑ What do we maximize with respect to?





Recall: K-means



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

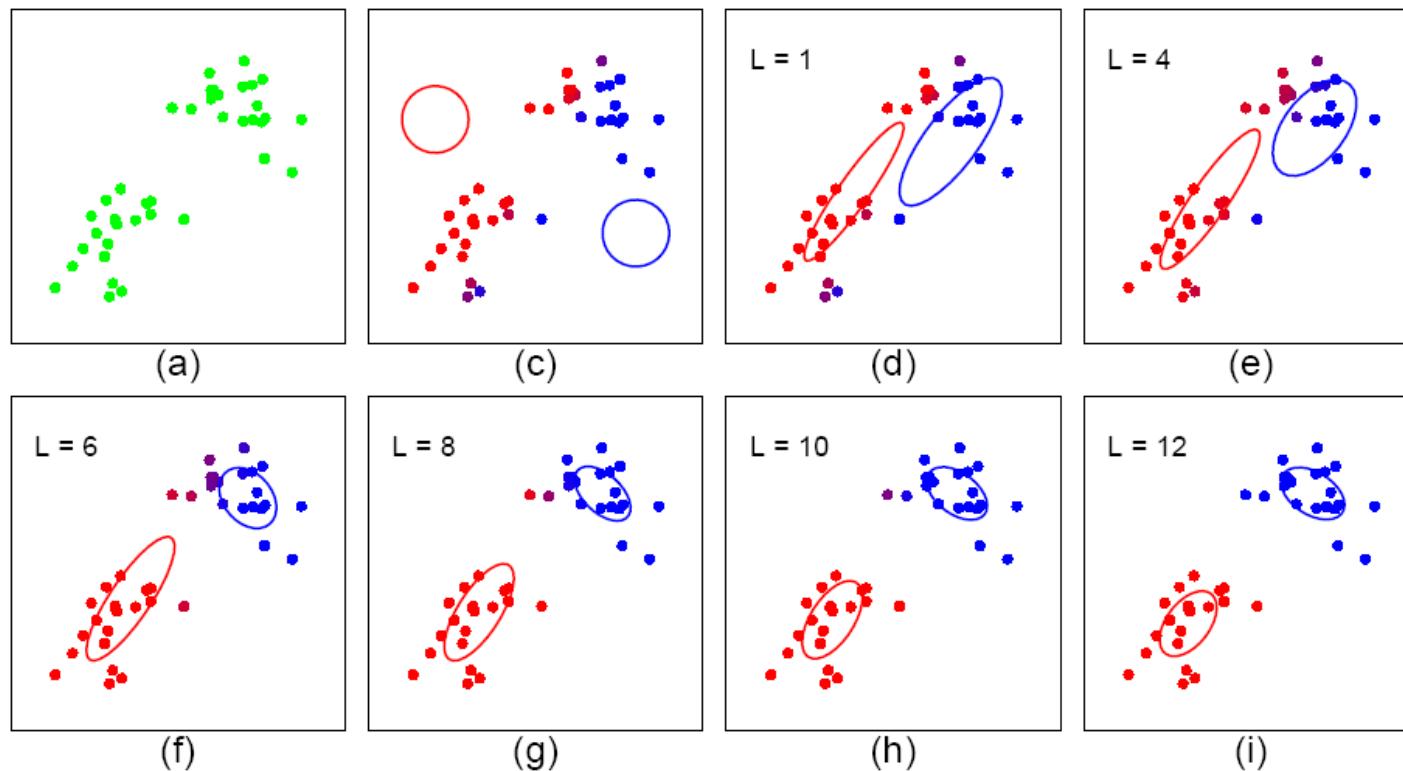
$$\mu_k^{(t+1)} = \frac{\sum_n \delta(z_n^{(t)}, k) x_n}{\sum_n \delta(z_n^{(t)}, k)}$$





Expectation-Maximization

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





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

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

- The likelihood of a sample:

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

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

*we want to est.
parameters*

- The expected complete log likelihood

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

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





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., \mathbf{z}) given current est. of the parameters (i.e., π and μ).

$$\tau_n^{k(t)} = \langle z_n^k \rangle_{q^{(t)}} = p(z_n^k = 1 | x, \mu^{(t)}, \Sigma^{(t)}) = \frac{\pi_k^{(t)} N(x_n | \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_i \pi_i^{(t)} N(x_n | \mu_i^{(t)}, \Sigma_i^{(t)})} \quad \left. \begin{array}{l} P(z_n^k = 1, x, \mu^{(+)}, \Sigma^{(+)}) \\ \underbrace{\pi_k^{(t)} N(x_n | \mu_k^{(t)}, \Sigma_k^{(t)})}_{P(x, \mu^{(+)}, \Sigma^{(+)})} \end{array} \right\} \text{Bayes rule}$$

(Here we are essentially doing **inference**)





M-step

- We maximize $\langle l_c(\theta) \rangle$ iteratively using the following iterative procedure:
 - **Maximization step:** compute the parameters under current results of the expected value of the hidden variables

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

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

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

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

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")





Compare: K-means and EM

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

- EM

- E-step

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

- M-step

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

- 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:

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





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

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

- 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 (or marginal) 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





Expected Complete Log Likelihood

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

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

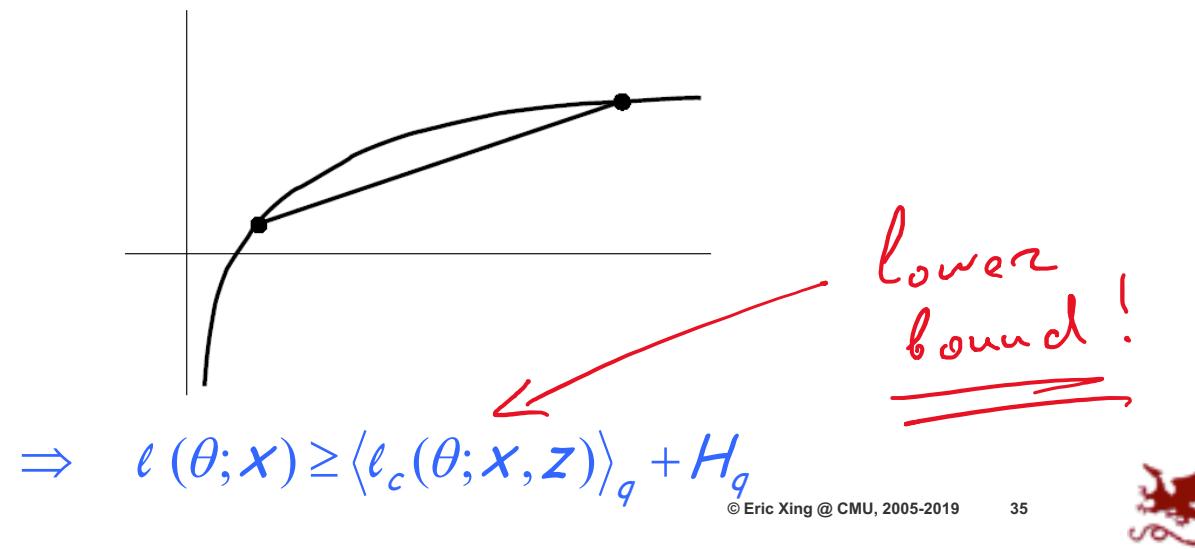
- A deterministic function of θ
- Linear in $\ell_c()$ → inherit its factorizability
- Does maximizing this surrogate yield a maximizer of the likelihood?

- Jensen's inequality

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

when is it tight?

$$\rightarrow \geq \sum_z q(z | x) \log \frac{p(x, z | \theta)}{q(z | x)}$$





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 :

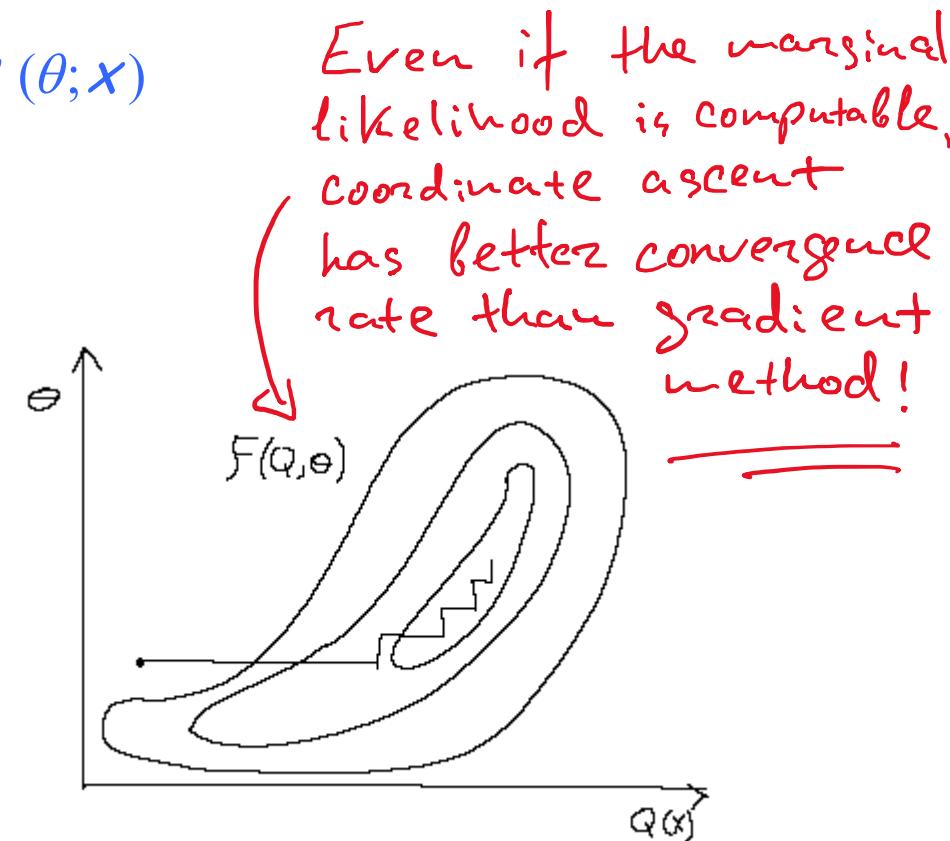
- E-step:

$$q^{t+1} = \arg \max_q F(q, \theta^t)$$

- M-step:

$$\theta^{t+1} = \arg \max_{\theta} F(q^{t+1}, \theta^t)$$

(generalization of EM: the MM-algorithm)





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

- Claim:

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

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 $\ell(\theta, x) \geq \mathcal{F}(q, \theta)$

$$\begin{aligned}\mathcal{F}(p(z|x, \theta^t), \theta^t) &= \sum_z p(z|x, \theta^t) \log \frac{p(x, z|\theta^t)}{p(z|x, \theta^t)} \\ &= \sum_z q(z|x) \log p(x|\theta^t) \quad \text{Bayes rule} \\ &= \log p(x|\theta^t) = \ell(\theta^t; x)\end{aligned}$$

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

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





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

- Note that the free energy breaks into two terms:

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

- 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:

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

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





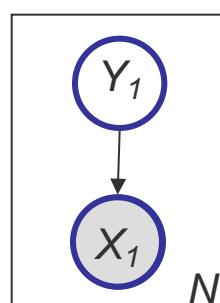
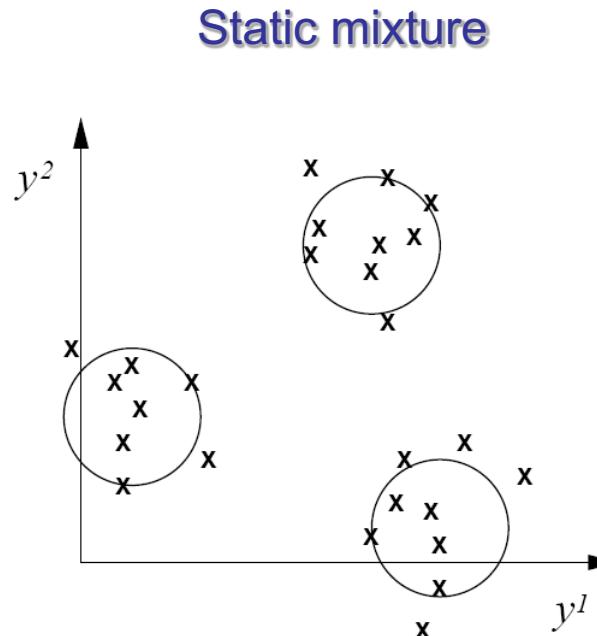
Example: HMM

- ❑ Supervised learning: estimation when the “right answer” is known
 - ❑ Examples:
GIVEN: a genomic region $x = x_1 \dots x_{1,000,000}$ 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 $P(x|\theta)$ --- Maximal likelihood (ML) estimation

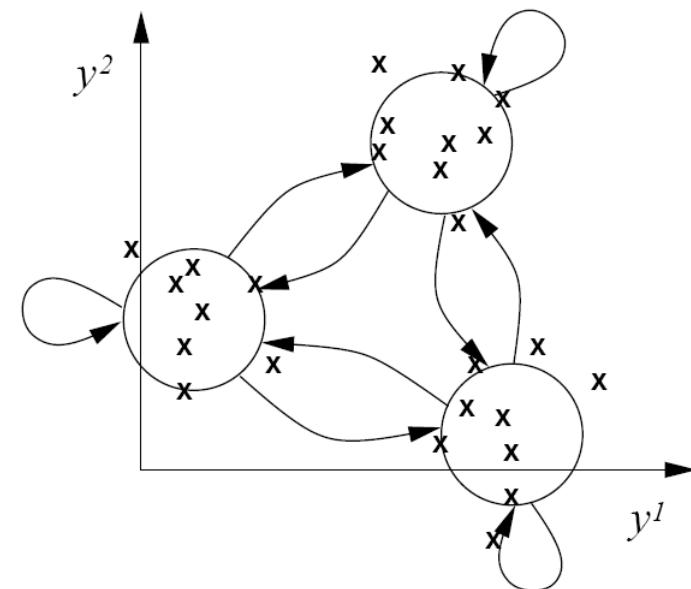




Hidden Markov Model: from static to dynamic mixture models



Dynamic mixture

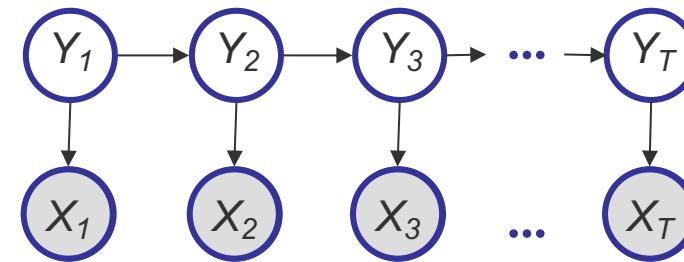


The underlying source:

Speech signal,
dice, ...

The sequence:
Phonemes,
sequence of rolls,

...





Supervised ML estimation

- Given $x = x_1 \dots x_N$ for which the true state path $y = y_1 \dots y_N$ is known,
 - Define:

A_{ij} = # times state transition $i \rightarrow j$ occurs in y

B_{ik} = # times state i in y emits k in x

- We can show that the **maximum likelihood** parameters θ are:

$$a_{ij}^{ML} = \frac{\#(i \rightarrow j)}{\#(i \rightarrow \bullet)} = \frac{\sum_n \sum_{t=2}^T y_{n,t-1}^i y_{n,t}^j}{\sum_n \sum_{t=2}^T y_{n,t-1}^i} = \frac{A_{ij}}{\sum_{j'} A_{ij'}}$$

$$b_{ik}^{ML} = \frac{\#(i \rightarrow k)}{\#(i \rightarrow \bullet)} = \frac{\sum_n \sum_{t=1}^T y_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^T y_{n,t}^i} = \frac{B_{ik}}{\sum_{k'} B_{ik'}}$$

- What if x is continuous? We can treat $\{(x_{n,t}, y_{n,t}): t=1:T, n=1:N\}$ as $N'T$ observations of, e.g., a Gaussian, and apply learning rules for Gaussian ...





The Baum Welch algorithm (or EM for HMM)

- The complete log likelihood

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

- 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

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

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

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

$$\pi_i^{ML} = \frac{\sum_n \gamma_{n,1}^i}{N}$$

$$a_{ij}^{ML} = \frac{\sum_n \sum_{t=2}^T \xi_{n,t}^{i,j}}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i}$$

$$b_{ik}^{ML} = \frac{\sum_n \sum_{t=1}^T \gamma_{n,t}^i x_{n,t}^k}{\sum_n \sum_{t=1}^{T-1} \gamma_{n,t}^i}$$





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

% M-step

$$ESS_i += \langle SS_i(x_{n,i}, x_{n,\pi_i}) \rangle_{p(x_{n,H}|x_{n,-H})}$$

for each node i

$\theta_i := \text{MLE}(ESS_i)$





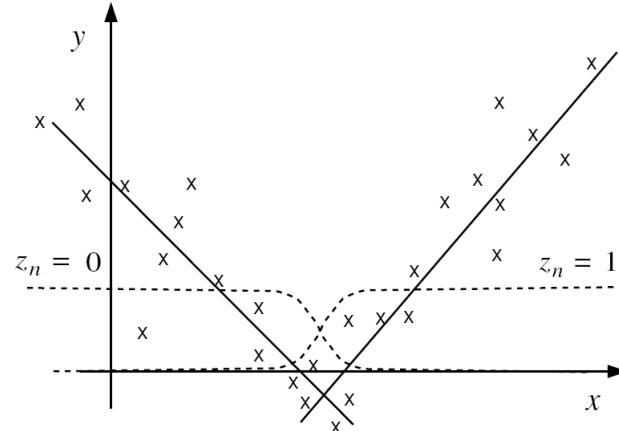
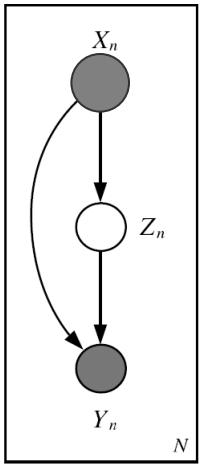
Summary: EM Algorithm

- ❑ 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: $q^{t+1} = \arg \max_q \mathcal{F}(q, \theta^t)$
 - ❑ M-step: $\theta^{t+1} = \arg \max_{\theta} \mathcal{F}(q^{t+1}, \theta^t)$
- ❑ 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:
$$P(y|x, z^k=1) = n(y; \theta_k^T x, \sigma_k^2)$$
 - 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

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

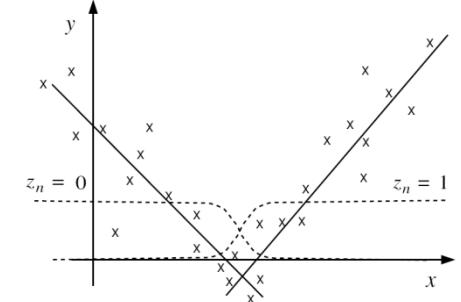
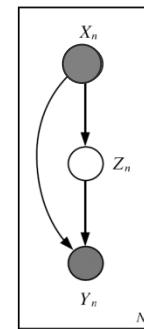
$$= \sum_n \sum_k \langle z_n^k \rangle \log(\text{softmax}(\xi_k^T x_n)) - \frac{1}{2} \sum_n \sum_k \langle z_n^k \rangle \left(\frac{(y_n - \theta_k^T x_n)^2}{\sigma_k^2} + \log \sigma_k^2 + C \right)$$

- EM:

- E-step: $\tau_n^{k(t)} = P(z_n^k=1|x_n, y_n, \theta) = \frac{p(z_n^k=1|x_n)p_k(y_n|x_n, \theta_k, \sigma_k^2)}{\sum_j p(z_n^j=1|x_n)p_j(y_n|x_n, \theta_j, \sigma_j^2)}$

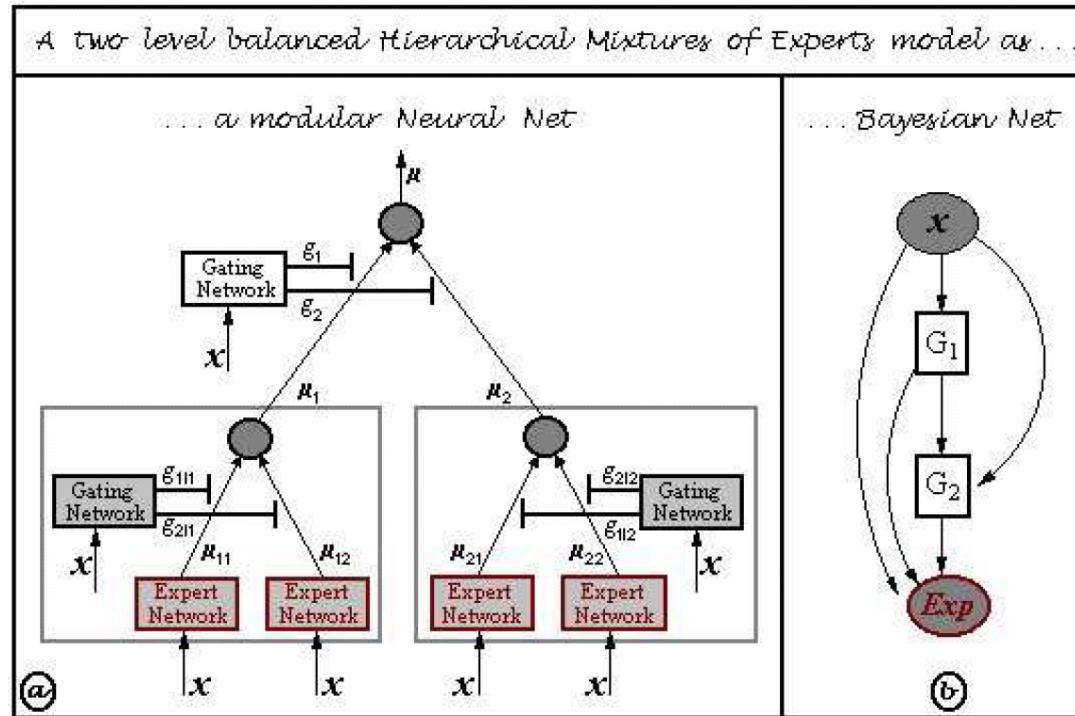
- M-step:

- using the normal equation for standard LR $\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$, 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 (homework)





Hierarchical mixture of experts

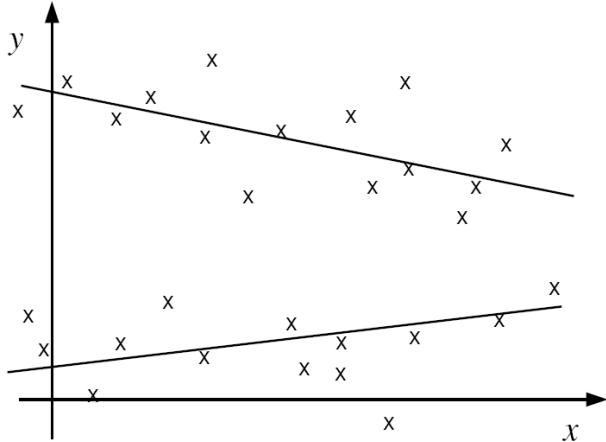
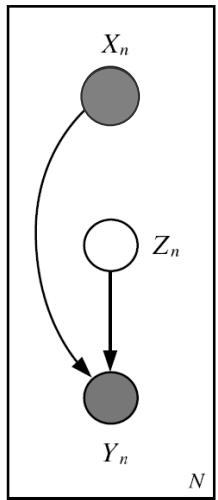


- This is like a soft version of a depth-2 classification/regression tree.
- $P(Y|X, G_1, G_2)$ can be modeled as a GLIM, with parameters dependent on the values of G_1 and G_2 (which specify a "conditional path" to a given leaf in the tree).





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 the likelihood
- ❑ Some bad things about EM:
 - ❑ Can get stuck in local minima
 - ❑ Can be slower than conjugate gradient (especially near convergence)
 - ❑ Requires computationally expensive inference step
 - ❑ Is a maximum likelihood/MAP method

