

# Learning Partially Observed GM: the Expectation-Maximization algorithm

**Kayhan Batmanghelich**

# 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)
  - Bayesian estimation
  - Maximal conditional likelihood
  - 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.

# Recall: 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

# Partially observed GMs

- Speech recognition

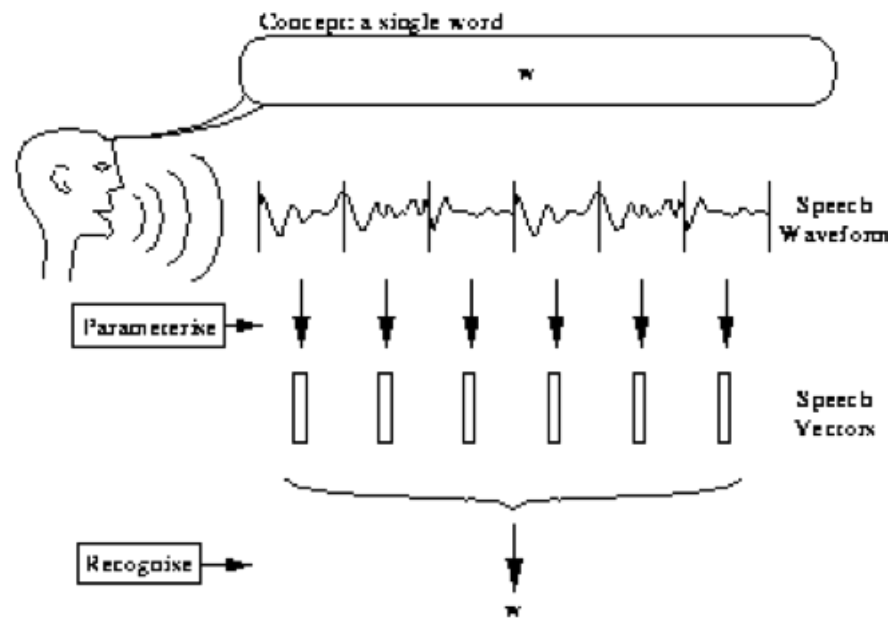
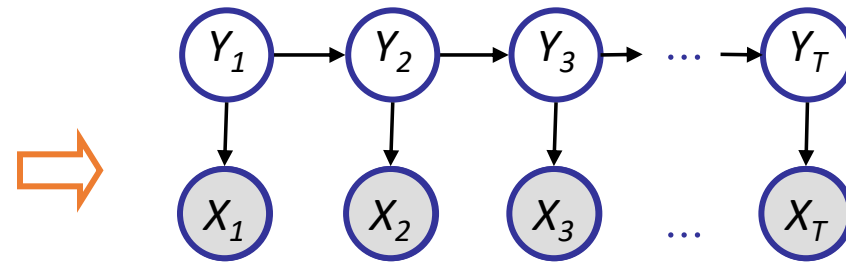
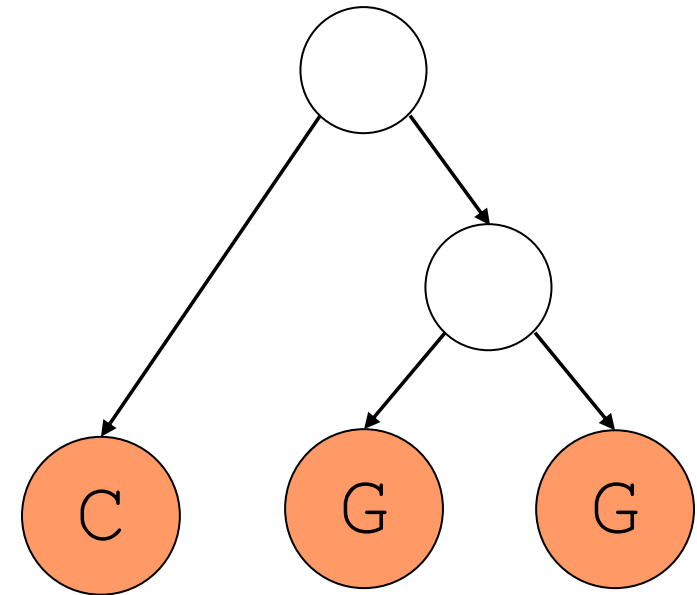
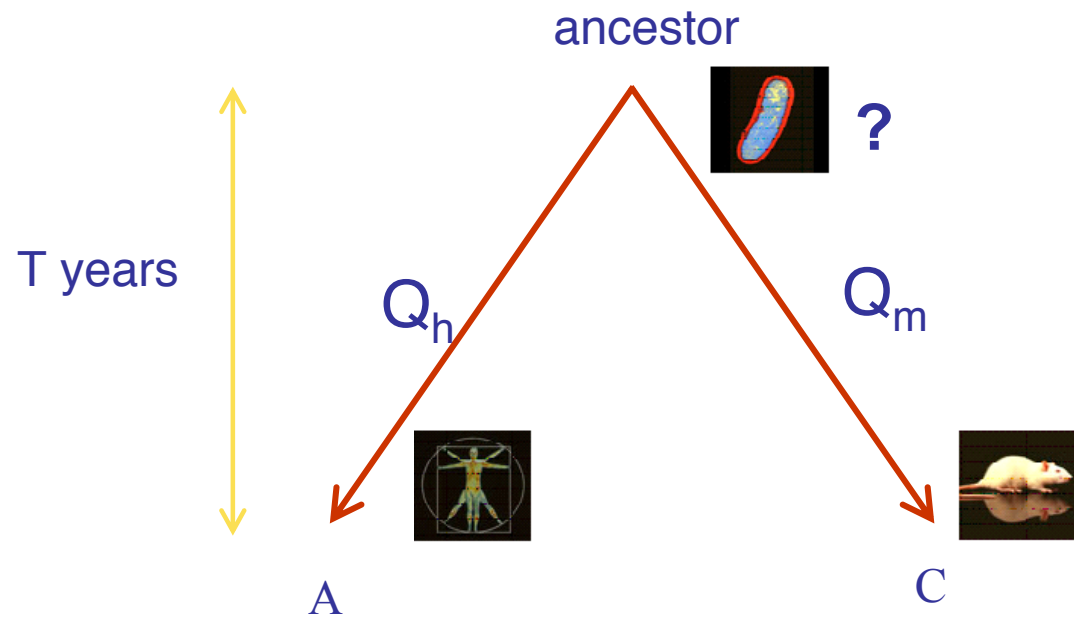


Fig. 1.2 Isolated Word Problem

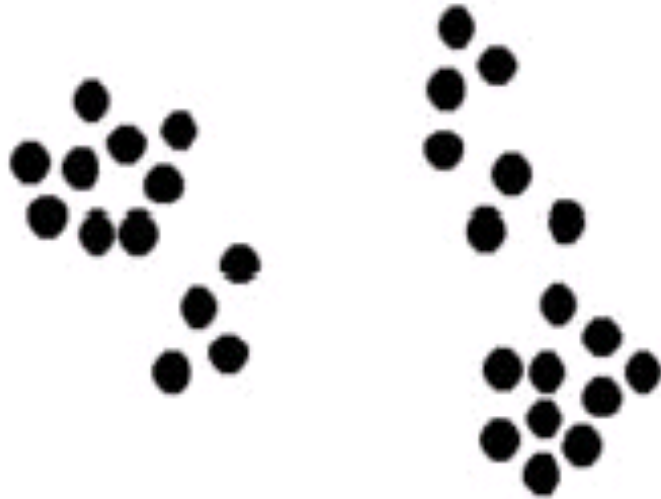


# Partially observed GM

- Biological Evolution

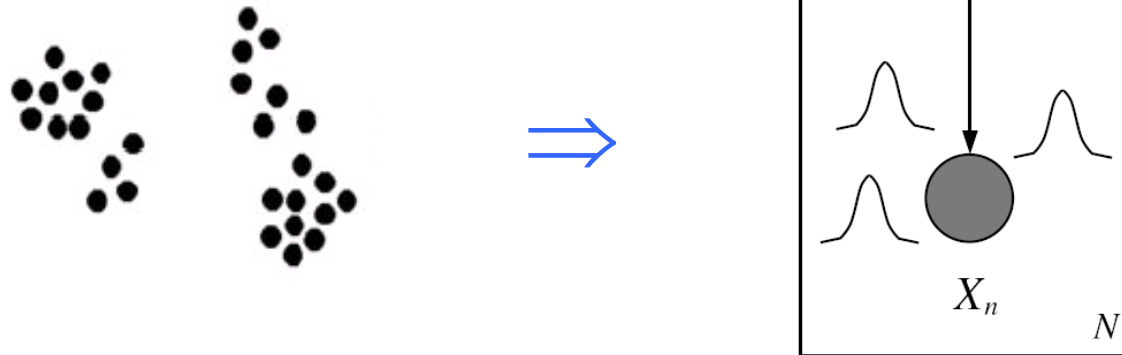


# Mixture Models



# Mixture Models, con'd

- A density model  $p(\mathbf{x})$  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

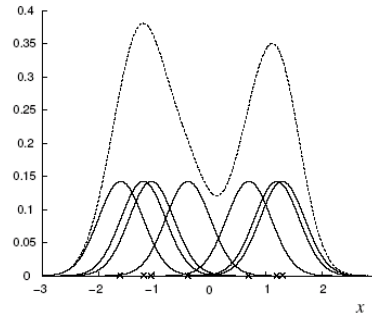
- A variable can be unobserved (latent) because:
  - it is an imaginary quantity meant to provide some **simplified and abstractive** view of the data generation process
    - e.g., speech recognition models, mixture models ...
  - it is a **real-world** object and/or phenomena, but difficult or **impossible to measure**
    - e.g., the temperature of a star, causes of a disease, evolutionary ancestors ...
  - it is a **noisy measurement** of the a real-world object (i.e. the true value is unobserved).
- **Example:** Discrete latent variables can be used to partition/cluster data into sub-groups.
- **Example:** Continuous latent variables (factors) can be used for dimensionality reduction (factor analysis, etc).



# Gaussian Mixture Models (GMMs)

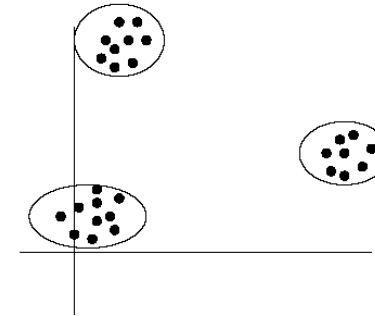
- Consider a mixture of  $K$  Gaussian components:

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



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:

$$p(z_n) = \text{Cat}(z_n; \pi) = \prod_k (\pi_k)^{\mathbb{I}(z_n=k)}$$

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

$$p(x_n | z_n = k; \{\mu_k, \Sigma_k\}_{k=1}^K) = \frac{1}{(2\pi)^{m/2} \det(\Sigma_k)^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right]$$

- The likelihood of a sample:

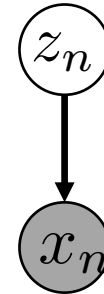
$$p(x_n | \{\mu_k, \Sigma_k\}_{k=1}^K) = \sum_k p(z_n = k; \pi) p(x_n | z_n = k; \{\mu_k, \Sigma_k\}_{k=1}^K)$$

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

mixture proportion

$$= \sum_k \pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)$$

mixture component



# 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

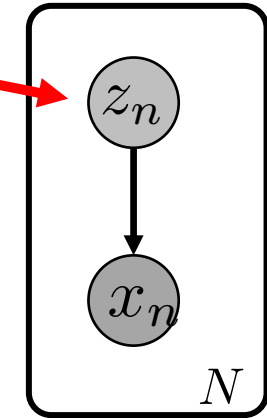
$$\begin{aligned}\ell(\boldsymbol{\theta}; D) &= \log \prod_n p(z_n, x_n) = \log \prod_n p(z_n | \pi) p(x_n | z_n, \mu, \sigma) \\ &= \sum_n \log \prod_k \pi_k^{z_n^k} + \sum_n \log \prod_k N(x_n; \mu_k, \sigma)^{z_n^k} \\ &= \sum_n \sum_k z_n^k \log \pi_k - \sum_n \sum_k z_n^k \frac{1}{2\sigma^2} (x_n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

- Separate** MLE

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

- What if we do not know  $z_n$ ?

Let's pretend it is observed

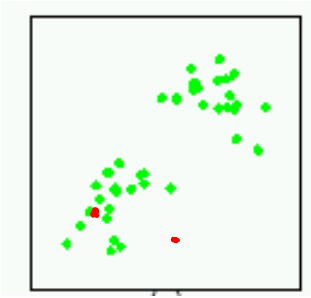


Let's assume  $\sigma$  is known

# Question

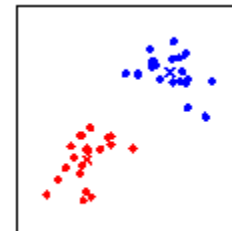
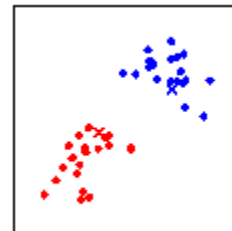
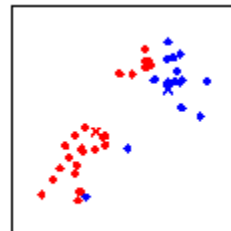
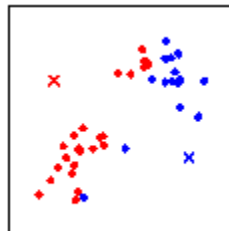
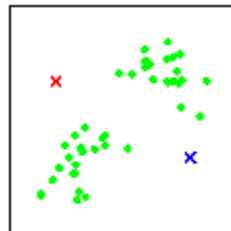
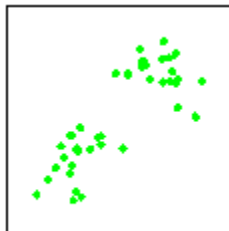
- “ ... We solve problem X using **Expectation-Maximization** ...”
  - What does it mean?
- E
  - What do we take expectation with?
  - What do we take expectation over?
- M
  - What do we maximize?
  - What do we maximize with respect to?

# Recall: K-means



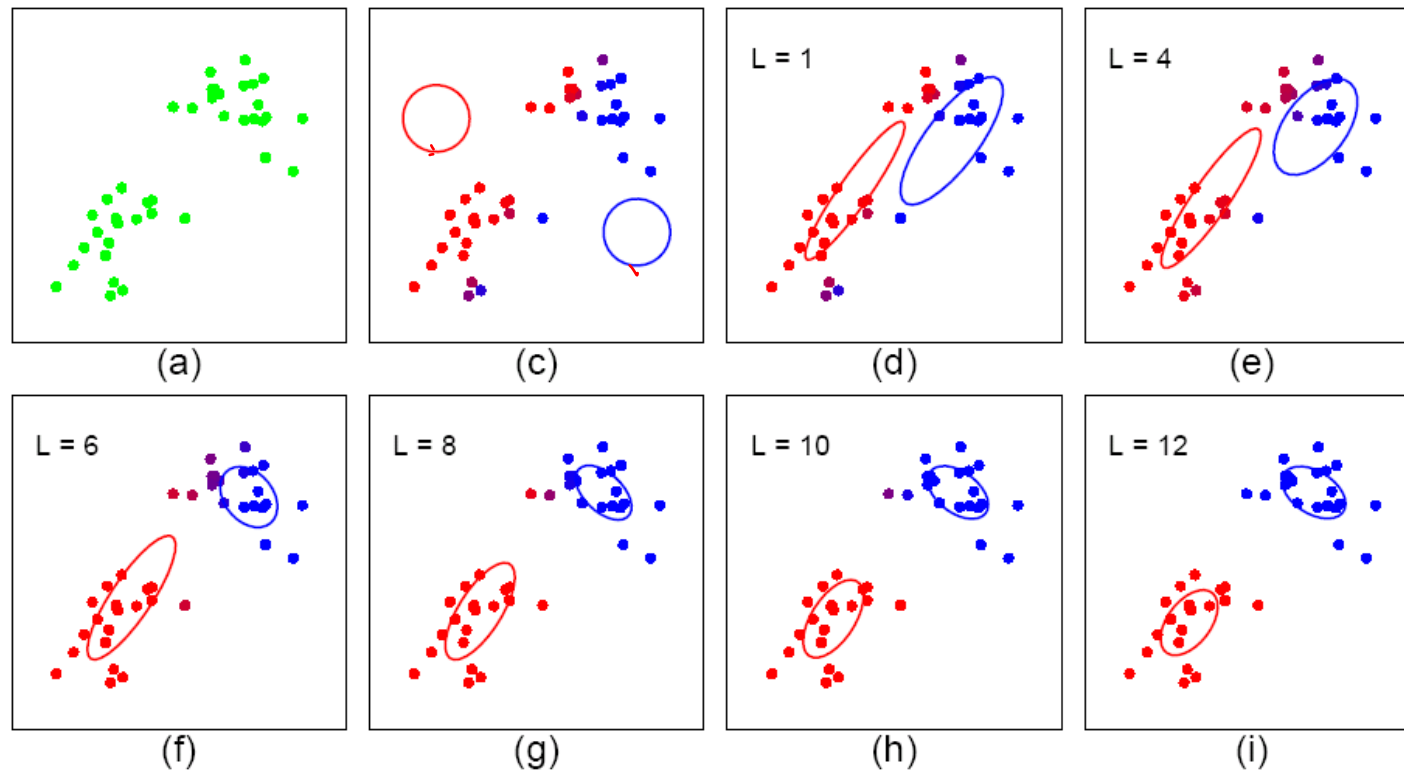
$$z_n^{(t)} = \arg \max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

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



# Expectation-Maximization

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



# Example: Gaussian mixture model

- A mixture of K Gaussians:

- $Z$  is a latent class indicator vector

$$p(z_n) = \text{Cat}(z_n; \pi) = \prod_k (\pi_k)^{\mathbb{I}(z_n = k)} \quad \text{with } z_n^k \text{ highlighted in orange}$$

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

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

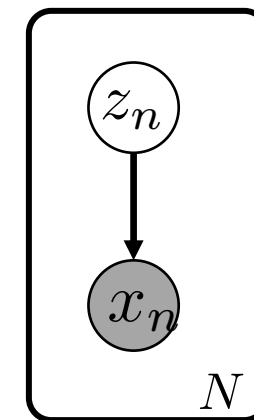
- The likelihood of a sample:

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

The final expression is circled in red, with a red arrow pointing from the parameter  $\theta$  (highlighted in orange) to the summation over  $z_n$ .

- The expected complete log likelihood

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





# 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.,  $\mathbf{z}$ ) given current est. of the parameters (i.e.,  $\pi$  and  $\mu$ ).

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

↑  
Like soft count

- Here we are essentially doing **inference**

# M-step

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

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

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

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

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

Fact :

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

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

- 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.

- 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

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

- 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

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

is difficult!

- What shall we do?

# Complete & Incomplete Log Likelihoods

- Complete log likelihood

Let  $\mathcal{X}$  denote the observable variable(s), and  $\mathcal{Z}$  denote the latent variable(s).  
If  $\mathcal{Z}$  could be observed, then

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

- 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 (c.f. **MLE for fully observed models**).
- But given that  $\mathcal{Z}$  is not observed,  $\ell_c()$  is a **random quantity**, cannot be maximized directly.

- Incomplete log likelihood

With  $\mathcal{Z}$  unobserved, our objective becomes the log of a marginal probability:

- **This objective won't decouple**

$$\ell_c(\theta; \mathbf{x}) = \log p(\mathbf{x} \mid \theta) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z} \mid \theta)$$

# Expected Complete Log Likelihood

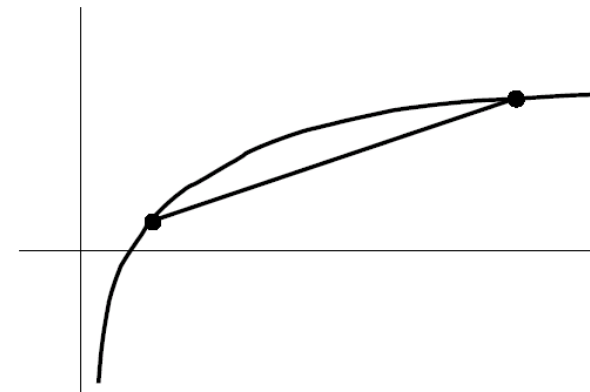
- For **any** distribution  $q(\mathbf{z})$ , define **expected complete log likelihood**:

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

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

- Jensen's inequality

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



$$\Rightarrow \ell(\theta; \mathbf{x}) \geq \langle \ell_c(\theta; \mathbf{x}, \mathbf{z}) \rangle_q + H_q$$

# Lower Bounds and Free Energy

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

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

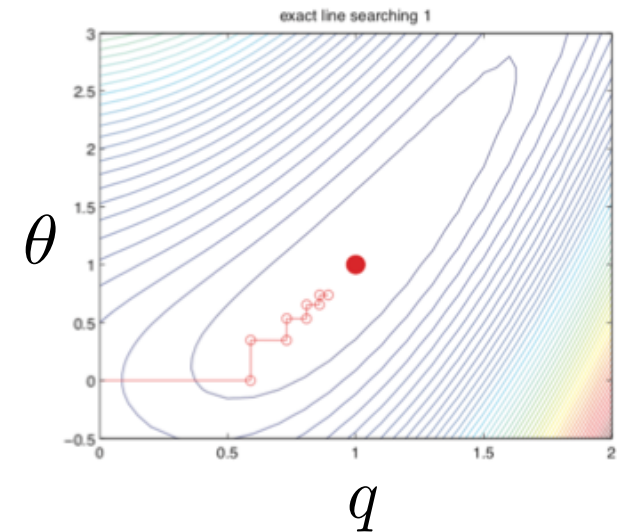
- The EM algorithm is coordinate-ascent on  $F$ :

- **E-step:**

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

- **M-step:**

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



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

- **Claim:**  $q^{t+1} = \arg \max_q F(q, \theta^t) = p(z | x, \theta^t)$ 
  - The **best solution** is the posterior 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  $l(\theta; x) \geq F(q, \theta)$

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

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

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



# 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 GLMs, 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

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

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

- Note that the free energy breaks into two terms:

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

- Thus, in the M-step, maximizing with respect to  $\theta$  for fixed  $q$  we only need to consider the first term:

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

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

# 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 annotations of the CpG islands (experimental)

**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

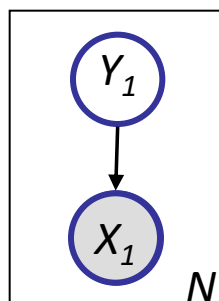
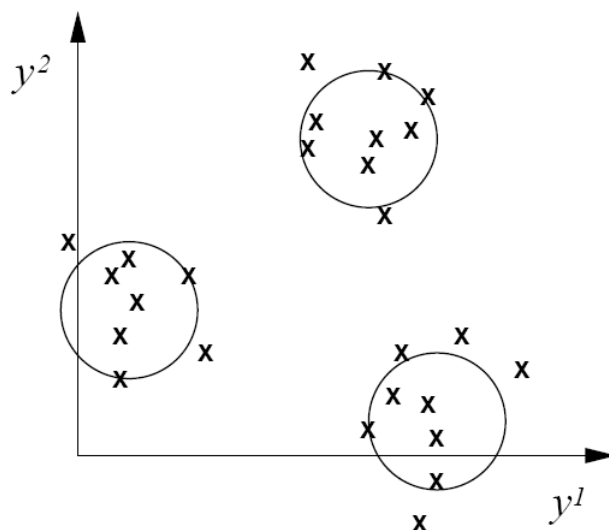
**GIVEN:** 10,000 rolls of the casino player, but we don't see when he changes dice

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

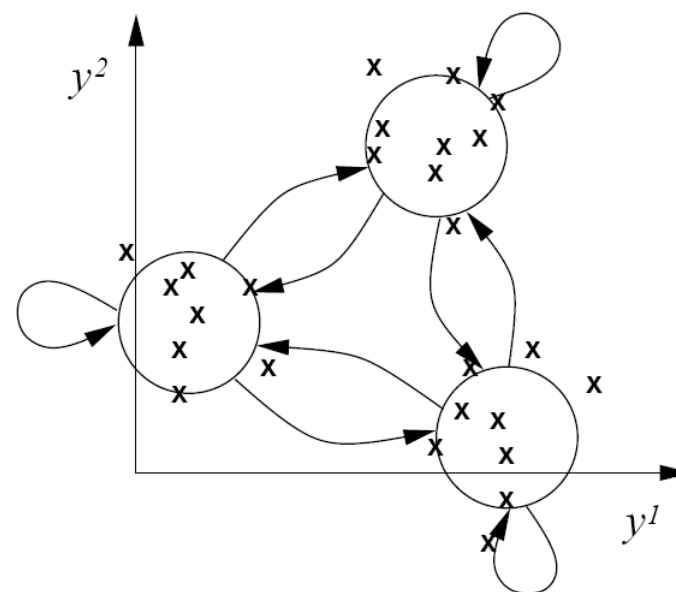
# Hidden Markov Model:

from static to dynamic mixture models

Static mixture



Dynamic mixture

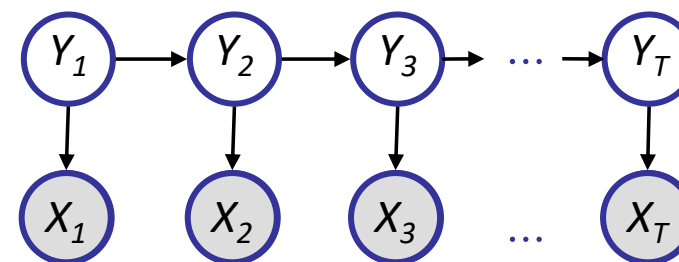


**The underlying source:**

Speech signal,  
dice,

**The sequence:**

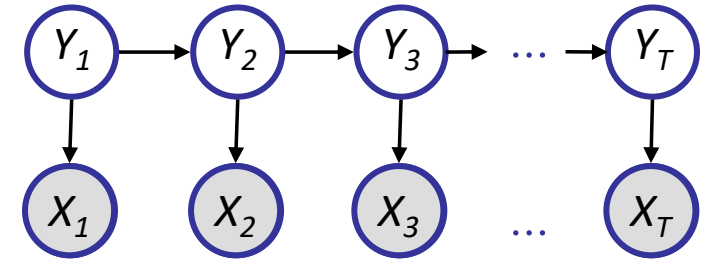
Phonemes,  
sequence of rolls,



# The Baum Welch algorithm

- The complete log likelihood

$$\ell_c(\mathbf{x}, \mathbf{y}; \theta) = \log p(\mathbf{x}, \mathbf{y}; \theta) = \log \prod_n \left( p(y_1^n) \prod_{t=2}^T p(y_t^n | y_{t-1}^n) \prod_{t=1}^T p(x_t^n | y_t^n) \right)$$



$$\langle \ell_c(\mathbf{x}, \mathbf{y}; \theta) \rangle = \sum_n (\langle y_1^n \rangle \log \pi) + \sum_n \sum_t \text{tr} (\langle y_t^n y_{t-1}^n \rangle \log A) + \sum_n \sum_t \langle y_t^n \rangle \log b(x_t^n)$$

$$\pi_i = p(y_1 = i)$$

$$p(y_t = j | y_{t-1} = i) = \{a_{ij}\} = A$$

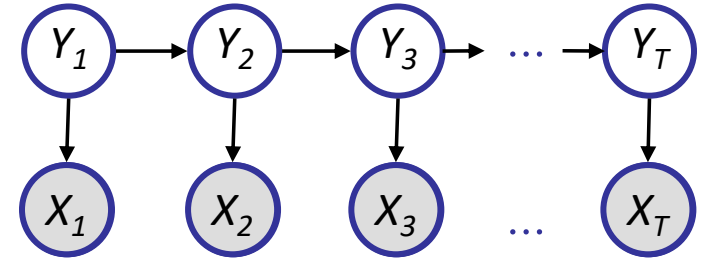
$$b_j(z) = p(x_t = z | y_t = j)$$

$\theta = (A, B, \pi)$  are the parameters of the model

# The Baum Welch algorithm

- The complete log likelihood

$$\ell_c(\mathbf{x}, \mathbf{y}; \theta) = \log p(\mathbf{x}, \mathbf{y}; \theta) = \log \prod_n \left( p(y_1^n) \prod_{t=2}^T p(y_t^n | y_{t-1}^n) \prod_{t=1}^T p(x_t^n | y_t^n) \right)$$



$$\langle \ell_c(\mathbf{x}, \mathbf{y}; \theta) \rangle = \sum_n (\langle y_1^n \rangle \log \pi) + \sum_n \sum_t \text{tr} (\langle y_t^n y_{t-1}^n \rangle \log A) + \sum_n \sum_t \langle y_t^n \rangle \log b(x_t^n)$$

$$\pi_i = p(y_1 = i)$$

$$p(y_t = j | y_{t-1} = i) = \{a_{ij}\} = A$$

$$b_j(z) = p(x_t = z | y_t = j)$$

- Fix  $\theta$  and compute the marginal posterior:
  - $p(y_t = i | \mathbf{x}; \theta)$ ,
  - $p(y_t = i, y_{t-1} = j | \mathbf{x}; \theta)$
- Update  $\theta$  by MLE (closed-form) – remember the soft count

Extension to general BN

# EM for general BNs

$$p(x) = \prod_i p(x_i | \text{pa}(x_i)).$$

$x$  represents both hidden and observed:  $x^n = (v^n, h^n)$

$$\mathcal{V} = \{v^1, \dots, v^N\}$$

$$\log p(\mathcal{V} | \theta) \geq \tilde{L}(\{q\}, \theta) \equiv \underbrace{- \sum_{n=1}^N \langle \log q(h^n | v^n) \rangle_{q(h^n | v^n)}}_{\text{entropy}} + \underbrace{\sum_{n=1}^N \langle \log p(h^n, v^n | \theta) \rangle_{q(h^n | v^n)}}_{\text{energy}}$$

$$\sum_n \langle \log p(x^n) \rangle_{q_t(h^n | v^n)} = \sum_n \sum_i \langle \log p(x_i^n | \text{pa}(x_i^n)) \rangle_{q_t(h^n | v^n)}$$

A bit of notation:

$$q_t^n(x) = q_t(h^n | v^n) \delta(v, v^n)$$

$$q_t(x) = \frac{1}{N} \sum_{n=1}^N q_t^n(x)$$

$$N \langle \log p(x) \rangle_{q_t(x)} = N \sum_x [\log p(x)] \frac{1}{N} \sum_n q_t(h^n | v^n) \delta(v, v^n) = \sum_n \langle \log p(x^n) \rangle_{q_t(h^n | v^n)}$$



# EM for general BNs

$$p(x) = \prod_i p(x_i | \text{pa}(x_i)).$$

$x$  represents both hidden and observed:  $x^n = (v^n, h^n)$

$$\mathcal{V} = \{v^1, \dots, v^N\}$$

$$\log p(\mathcal{V} | \theta) \geq \tilde{L}(\{q\}, \theta) \equiv \underbrace{- \sum_{n=1}^N \langle \log q(h^n | v^n) \rangle_{q(h^n | v^n)}}_{\text{entropy}} + \underbrace{\sum_{n=1}^N \langle \log p(h^n, v^n | \theta) \rangle_{q(h^n | v^n)}}_{\text{energy}}$$

$$KL(q_t(x_i | \text{pa}(x_i)) || p(x_i | \text{pa}(x_i)))$$

$$N \langle \log p(x) \rangle_{q_t(x)}$$

$$\sum_i \left\langle \langle \log p(x_i | \text{pa}(x_i)) \rangle_{q_t(x_i | \text{pa}(x_i))} \right\rangle_{q_t(\text{pa}(x_i))}$$

$$\sum_i \left\langle \langle \log q_t(x_i | \text{pa}(x_i)) \rangle_{q_t(x_i | \text{pa}(x_i))} - \langle \log p(x_i | \text{pa}(x_i)) \rangle_{q_t(x_i | \text{pa}(x_i))} \right\rangle_{q_t(\text{pa}(x_i))}$$

# EM for general BNs

$$p(x) = \prod_i p(x_i | \text{pa}(x_i)).$$

$x$  represents both hidden and observed:  $x^n = (v^n, h^n)$

$$\mathcal{V} = \{v^1, \dots, v^N\}$$

```

1:  $t = 1$ 
2: Set  $p_t(x_i | \text{pa}(x_i))$  to initial values.
3: while  $p(x_i | \text{pa}(x_i))$  not converged (or likelihood not converged)
4:    $t \leftarrow t + 1$ 
5:   for  $n = 1$  to  $N$  do
6:      $q_t^n(x) = p_t(h^n | v^n) \delta(v, v^n)$ 
7:   end for
8:   for  $i = 1$  to  $K$  do
9:      $p_{t+1}(x_i | \text{pa}(x_i)) = \frac{\sum_{n=1}^N q_t^n(x_i, \text{pa}(x_i))}{\sum_{n'=1}^N q_t^{n'}(\text{pa}(x_i))}$ 
10:  end for
11: end while
12: return  $p_t(x_i | \text{pa}(x_i))$ 

```

$$KL(q_t(x_i | \text{pa}(x_i)) || p(x_i | \text{pa}(x_i)))$$

$$p^{new}(x_i | \text{pa}(x_i)) = q_t(x_i | \text{pa}(x_i))$$

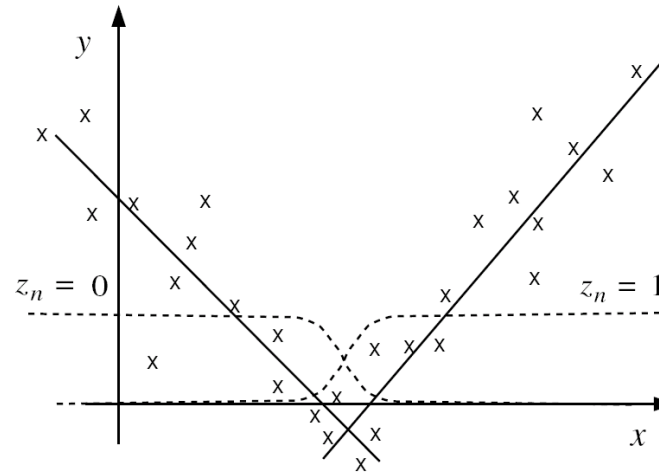
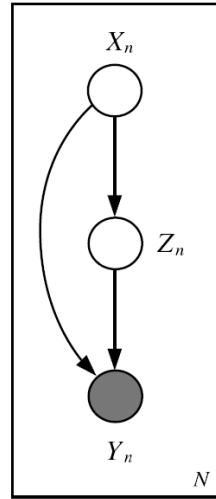
$$p^{new}(x_i | \text{pa}(x_i)) = \frac{\sum_n q_t^n(x_i, \text{pa}(x_i))}{\sum_{n'} q_t^{n'}(\text{pa}(x_i))}$$

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

More Examples

# Conditional mixture model: Mixture of experts



- We will model  $p(\mathcal{Y} | \mathcal{X})$  using different experts, each responsible for different regions of the input space.

- Latent variable  $\mathbf{Z}$  chooses expert using softmax gating function:

$$P(\mathbf{z}^k = 1 | \mathbf{x}) = \text{Softmax}(\boldsymbol{\xi}^T \mathbf{x})$$

- Each expert can be a linear regression model:  $P(y | \mathbf{x}, \mathbf{z}^k = 1) = \mathcal{N}(y; \theta_k^T \mathbf{x}, \sigma_k^2)$

- The posterior expert responsibilities are

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

# EM for conditional mixture model

- Model:

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

- The objective function

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

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

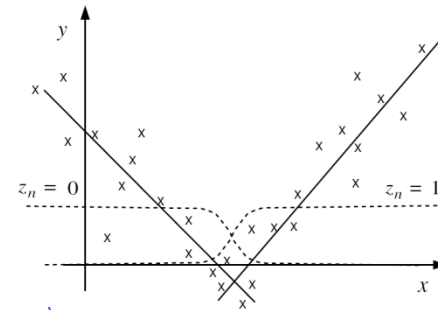
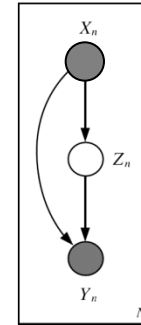
- EM:

- E-step:

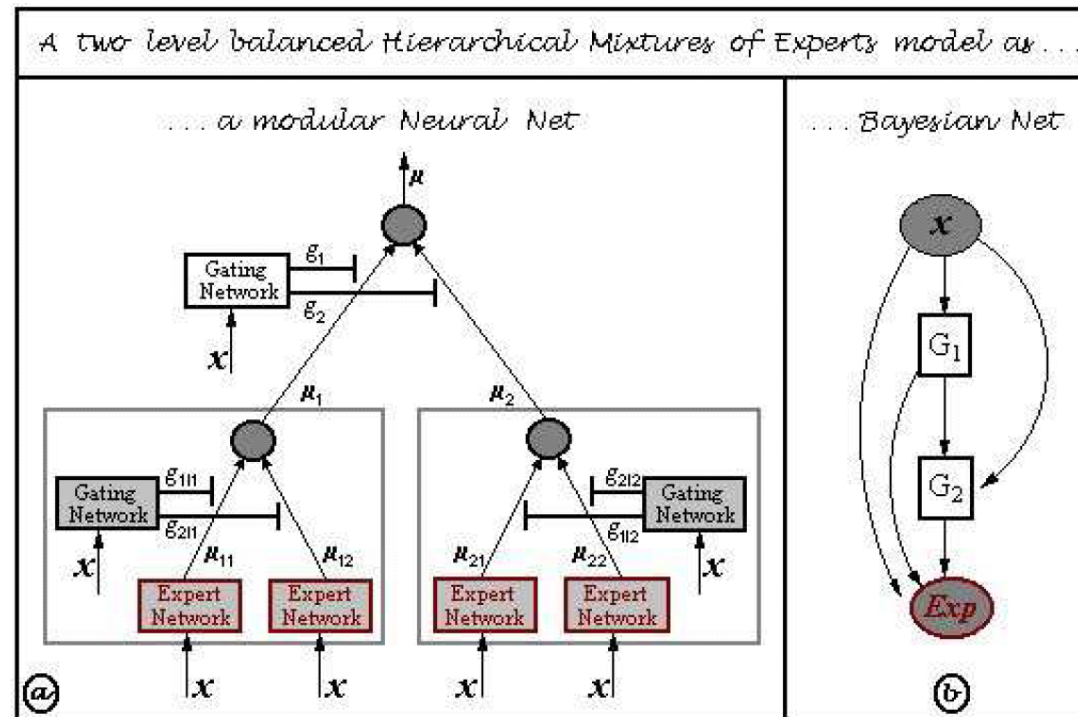
$$\tau_n^{k(t)} = p(z_n^k = 1 | x_n, y_n, \theta) = \frac{p(z_n^k = 1 | x_n) p_k(y_n | x_n, \theta_k, \sigma_k^2)}{\sum_j p(z_n^j = 1 | x_n) p_j(y_n | x_n, \theta_j, \sigma_j^2)}$$

- M-step:

- using the normal equation for standard LR  $\theta = (X^T X)^{-1} X^T Y$ , but with the data re-weighted by  $\tau$  (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?)

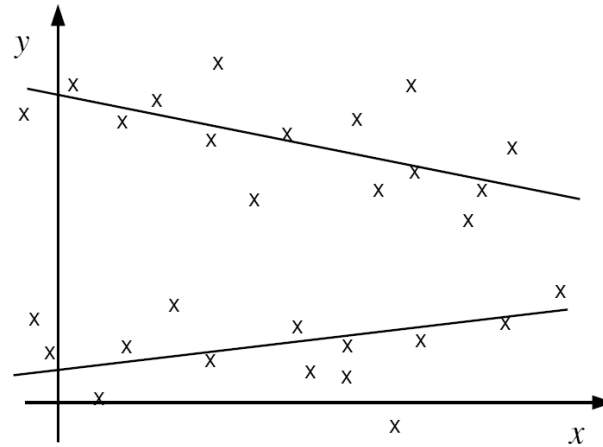
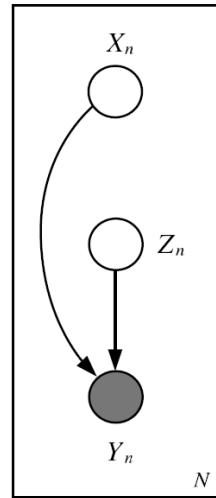


# Hierarchical mixture of experts



- 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.

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



# 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