PETUUM

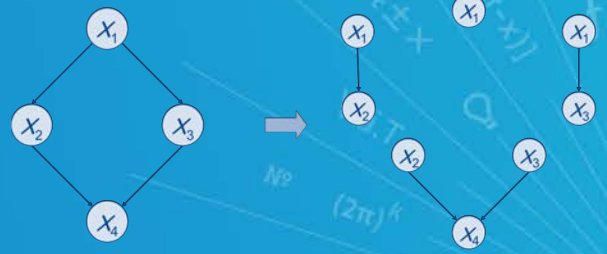
Carnegie Mellon University
School of Computer Science

Probabilistic Graphical Models

Parameter Estimation

Eric Xing

Lecture 5, January 29, 2020

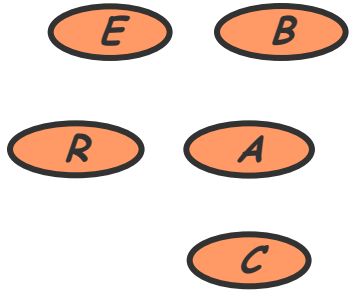


Reading: see class homepage

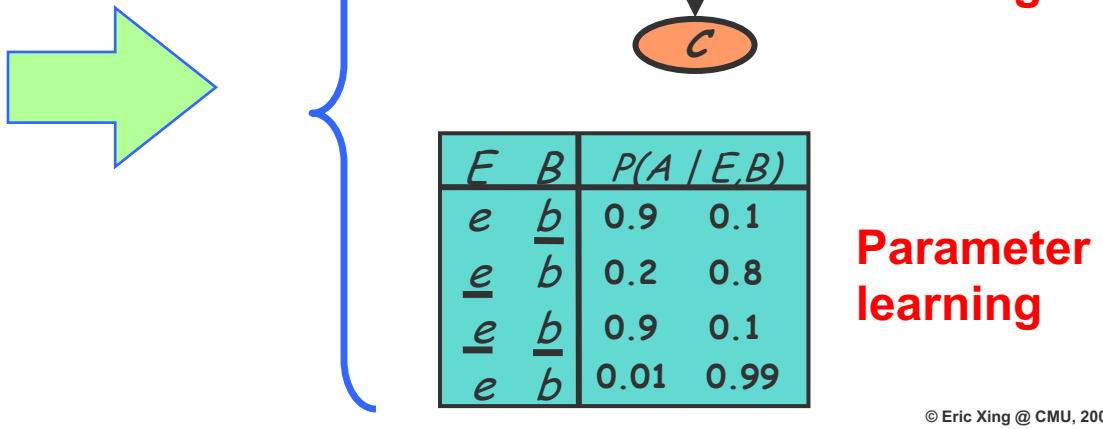
Learning Graphical Models

The goal:

- Given set of independent samples (*assignments* of random variables), find the *best* (the most likely?) Bayesian Network (both DAG and CPDs)

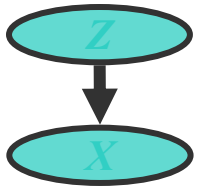


(B,E,A,C,R)=(T,F,F,T,F)
(B,E,A,C,R)=(T,F,T,T,F)
……..
(B,E,A,C,R)=(F,T,T,T,F)



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.



ML Parameter Est. for completely observed GMs of given structure

- The data:

$$\{ (z_1,x_1), (z_2,x_2), (z_3,x_3), \ldots (z_N,x_N)\}$$

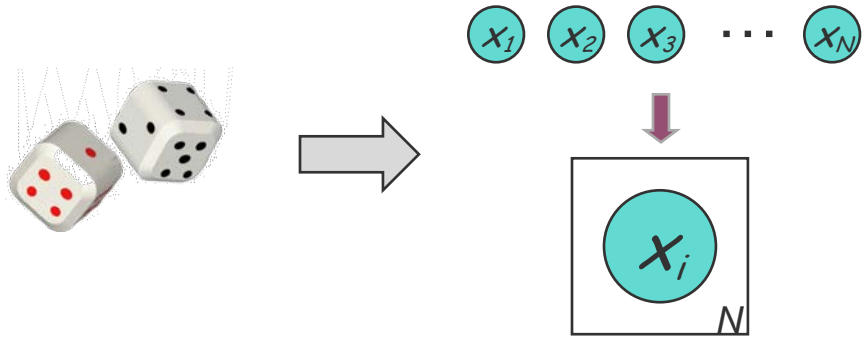
Parameter Learning

- Assume G is known and fixed,
  - from expert design
  - from an intermediate outcome of iterative structure learning
- Goal: estimate from a dataset of N independent, identically distributed (*iid*) training cases $D = \{x_1, \ldots, x_N\}$.
- In general, each training case $\mathbf{x}_n = (x_{n,1}, \ldots, x_{n,M})$ is a vector of M values, one per node,
  - the model can be completely observable, i.e., every element in x_n is known (no missing values, no hidden variables),
  - or, partially observable, i.e., $\exists i$, s.t. $x_{n,i}$ is not observed.
- In this lecture we consider learning parameters for a BN with given structure and is completely observable

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

Review of density estimation

- Can be viewed as single-node GMs
- Instances of
  Exponential Family Dist.
- Building blocks of general GM
- MLE and Bayesian estimate
- See supplementary slides



$$P(x_i) = P(\{x_{n,k} = 1, \text{where } k \text{ index the die-side of the } n\text{th roll}\})$$

$$= \theta_k = \theta_1^{x_{n,1}} \times \theta_2^{x_{n,2}} \times \cdots \times \theta_K^{x_{n,K}} = \prod_{k=1}^{K} \theta_k^{x_{n,k}}$$

$$P(x_1, x_2, \ldots, x_N \mid \theta) = \prod_{n=1}^{N} P(x_n \mid \theta) = \prod_{n=1}^{N} \left(\prod_k \theta_k^{x_{n,k}} \right) = \prod_k \theta_k^{\sum_{n=1}^{N} x_{n,k}} = \prod_k \theta_k^{n_k}$$

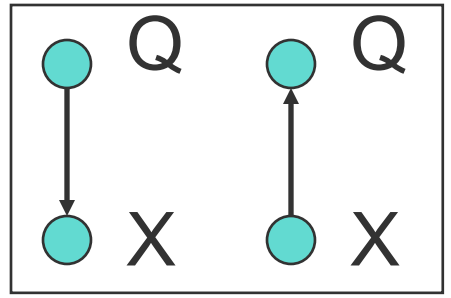
$$\frac{\partial \ell}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0$$

$$n_k = \lambda \theta_k \Rightarrow \sum_k n_k = N = \lambda \sum_k \theta_k = \lambda$$

$$\Rightarrow \hat{\theta}_{k,MLE} = \frac{n_k}{N} = \frac{1}{N} \sum_n x_{n,k}$$

Estimation of conditional density

- Can be viewed as two-node graphical models
- Instances of GLIM (Generalized Linear Models)
- Building blocks of general GM
- MLE and Bayesian estimate
- See supplementary slides



Exponential family, a basic building block

- For a numeric random variable X

$$p(x \mid \eta) = h(x)\exp\left\{\eta^T T(x) - A(\eta)\right\}$$
$$= \frac{1}{Z(\eta)} h(x)\exp\left\{\eta^T T(x)\right\}$$

is an exponential family distribution with natural (canonical) parameter η

- Function $T(x)$ is a *sufficient statistic*.
- Function $A(\eta) = \log Z(\eta)$ is the log normalizer.
- Examples: Bernoulli, multinomial, Gaussian, Poisson, gamma,...

Example: Multivariate Gaussian Distribution

- For a continuous vector random variable $X \in R^k$:

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

Moment parameter

$$= \frac{1}{(2\pi)^{k/2}} \exp\left\{-\tfrac{1}{2}\mathrm{tr}\left(\Sigma^{-1}xx^T\right) + \mu^T\Sigma^{-1}x - \tfrac{1}{2}\mu^T\Sigma^{-1}\mu - \log|\Sigma|\right\}$$

- Exponential family representation

Natural parameter

$$\eta = \left[\Sigma^{-1}\mu; -\tfrac{1}{2}\mathrm{vec}\left(\Sigma^{-1}\right)\right] = \left[\eta_1, \mathrm{vec}(\eta_2)\right], \quad \eta_1 = \Sigma^{-1}\mu \text{ and } \eta_2 = -\tfrac{1}{2}\Sigma^{-1}$$

$$T(x) = \left[x; \mathrm{vec}\left(xx^T\right)\right]$$

$$A(\eta) = \tfrac{1}{2}\mu^T\Sigma^{-1}\mu + \log|\Sigma| = -\tfrac{1}{2}\mathrm{tr}(\eta_2\eta_1\eta_1^T) - \tfrac{1}{2}\log(-2\eta_2)$$

$$h(x) = (2\pi)^{-k/2}$$

  - Note: a k-dimensional Gaussian is a $(d+d^2)$-parameter distribution with a $(d+d^2)$-element vector of sufficient statistics (but because of symmetry and positivity, parameters are constrained and have lower degree of freedom)

Example: Multinomial distribution

- For a binary vector random variable $x \sim \text{multi}(x \mid \pi)$,

$$p(x|\pi) = \pi_1^{x_1}\pi_2^{x_1}\cdots\pi_K^{x_K} = \exp\left\{\sum_k x_k \ln \pi_k\right\}$$

$$= \exp\left\{\sum_{k=1}^{K-1} x_k \ln \pi_k + \left(1-\sum_{k=1}^{K-1} x_K\right)\ln\left(1-\sum_{k=1}^{K-1}\pi_k\right)\right\}$$

$$= \exp\left\{\sum_{k=1}^{K-1} x_k \ln\left(\frac{\pi_k}{1-\sum_{k=1}^{K-1}\pi_k}\right) + \ln\left(1-\sum_{k=1}^{K-1}\pi_k\right)\right\}$$

- Exponential family representation

$$\eta = \left[\ln\left(\pi_k / \pi_K\right); 0\right]$$

$$T(x) = [x]$$

$$A(\eta) = -\ln\left(1-\sum_{k=1}^{K-1}\pi_k\right) = \ln\left(\sum_{k=1}^{K} e^{\eta_k}\right)$$

$$h(x) = 1$$

Why exponential family?

- Moment generating property

$$\frac{dA}{d\eta} = \frac{d}{d\eta}\log Z(\eta) = \frac{1}{Z(\eta)}\frac{d}{d\eta}Z(\eta)$$
$$= \frac{1}{Z(\eta)}\frac{d}{d\eta}\int h(x)\exp\{\eta^T T(x)\}dx$$
$$= \int T(x)\frac{h(x)\exp\{\eta^T T(x)\}}{Z(\eta)}dx$$
$$= E[T(x)]$$

$$\frac{d^2 A}{d\eta^2} = \int T^2(x)\frac{h(x)\exp\{\eta^T T(x)\}}{Z(\eta)}dx - \int T(x)\frac{h(x)\exp\{\eta^T T(x)\}}{Z(\eta)}dx\frac{1}{Z(\eta)}\frac{d}{d\eta}Z(\eta)$$
$$= E[T^2(x)] - E^2[T(x)]$$
$$= Var[T(x)]$$

Moment estimation

- We can easily compute moments of any exponential family distribution by taking the derivatives of the log normalizer $A(\eta)$.
- The q^{th} derivative gives the q^{th} centered moment.

$$\frac{dA(\eta)}{d\eta} = \text{mean}$$

$$\frac{d^2 A(\eta)}{d\eta^2} = \text{variance}$$

$$\ldots$$

- When the sufficient statistic is a stacked vector, partial derivatives need to be considered.

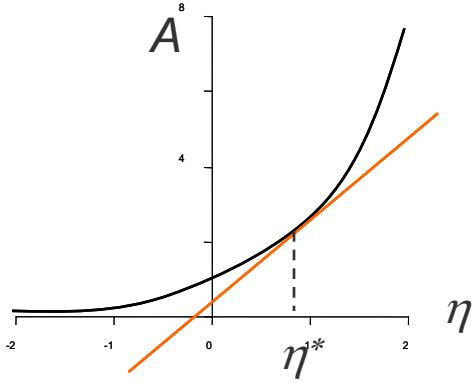
Moment vs canonical parameters

- The moment parameter μ can be derived from the natural (canonical) parameter

$$\frac{dA(\eta)}{d\eta} = E[T(x)] \overset{\text{def}}{=} \mu$$

- $A(\eta)$ is convex since

$$\frac{d^2 A(\eta)}{d\eta^2} = Var[T(x)] > 0$$



- Hence we can invert the relationship and infer the canonical parameter from the moment parameter (1-to-1):

$$\eta \overset{\text{def}}{=} \psi(\mu)$$

  - A distribution in the exponential family can be parameterized not only by η – the canonical parameterization, but also by μ – the moment parameterization.

MLE for Exponential Family

- For *iid* data, the log-likelihood is

$$\ell(\eta; D) = \log \prod_n h(x_n) \exp\left\{\eta^T T(x_n) - A(\eta)\right\}$$

$$= \sum_n \log h(x_n) + \left(\eta^T \sum_n T(x_n)\right) - NA(\eta)$$

- Take derivatives and set to zero:

$$\frac{\partial \ell}{\partial \eta} = \sum_n T(x_n) - N \frac{\partial A(\eta)}{\partial \eta} = 0$$

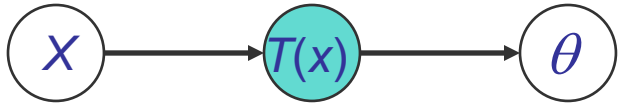
$$\Rightarrow \quad \frac{\partial A(\eta)}{\partial \eta} = \frac{1}{N} \sum_n T(x_n)$$

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_n T(x_n)$$

- This amounts to moment matching.
- We can infer the canonical parameters using $\hat{\eta}_{MLE} = \psi(\hat{\mu}_{MLE})$

Sufficiency

- For $p(x|\theta)$, $T(x)$ is *sufficient* for θ if there is no information in X regarding θ beyond that in $T(x)$.
  - We can throw away X for the purpose of inference w.r.t. θ .
  - Bayesian view

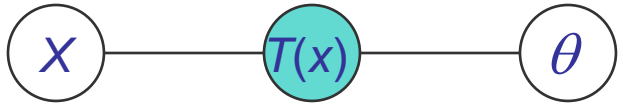


$$p(\theta \mid T(x), x) = p(\theta \mid T(x))$$

  - Frequentist view

$$p(x \mid T(x), \theta) = p(x \mid T(x))$$

  - The Neyman factorization theorem
    - $T(x)$ is *sufficient* for θ if



$$p(x, T(x), \theta) = \psi_1(T(x), \theta)\psi_2(x, T(x))$$

$$\Rightarrow p(x \mid \theta) = g(T(x), \theta)h(x, T(x))$$

Examples

- Gaussian:

$$\eta = \left[\Sigma^{-1}\mu; -\tfrac{1}{2}\,\mathrm{vec}\left(\Sigma^{-1}\right)\right]$$
$$T(x) = \left[x; \mathrm{vec}\left(xx^T\right)\right]$$
$$A(\eta) = \tfrac{1}{2}\,\mu^T\Sigma^{-1}\mu + \tfrac{1}{2}\log|\Sigma|$$
$$h(x) = (2\pi)^{-k/2}$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n T_1(x_n) = \frac{1}{N}\sum_n x_n$$

- Multinomial:

$$\eta = \left[\ln\left(\pi_k / \pi_K\right); 0\right]$$
$$T(x) = [x]$$
$$A(\eta) = -\ln\left(1 - \sum_{k=1}^{K-1}\pi_k\right) = \ln\left(\sum_{k=1}^{K} e^{\eta_k}\right)$$
$$h(x) = 1$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n x_n$$

- Poisson:

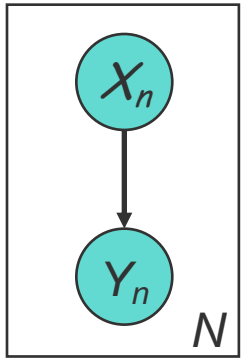
$$\eta = \log\lambda$$
$$T(x) = x$$
$$A(\eta) = \lambda = e^{\eta}$$
$$h(x) = \frac{1}{x!}$$

$$\Rightarrow \mu_{MLE} = \frac{1}{N}\sum_n x_n$$

Generalized Linear Models (GLIMs)

- The graphical model
  - Linear regression
  - Discriminative linear classification
  - Commonality:

    model $\mathrm{E}_p(Y)=\mu=f(\theta^T X)$
    - What is $p()$? the cond. dist. of Y.
    - What is $f()$? the response function.



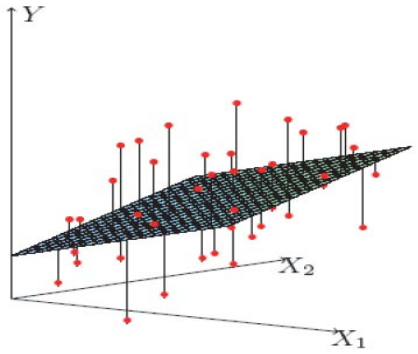
- GLIM
  - The observed input x is assumed to enter into the model via a linear combination of its elements
  - The conditional mean μ is represented as a function $f(\xi)$ of ξ, where f is known as the response function $\xi = \theta^T x$
  - The observed output y is assumed to be characterized by an exponential family distribution with conditional mean μ.

Recall Linear Regression

- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

where ε is an error term of unmodeled effects or random noise



- Now assume that ε follows a Gaussian $N(0,\sigma)$, then we have

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2}\right)$$

- We can use LMS algorithm, which is a gradient ascent/descent approach, to estimate the parameter

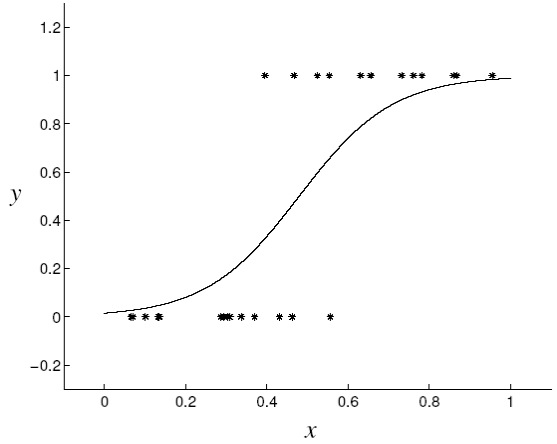
Recall: Logistic Regression (sigmoid classifier, perceptron, etc.)

- The condition distribution: a Bernoulli

$$p(y \mid x) = \mu(x)^y (1 - \mu(x))^{1-y}$$

where μ is a logistic function

$$\mu(x) = \frac{1}{1 + e^{-\theta^T x}}$$



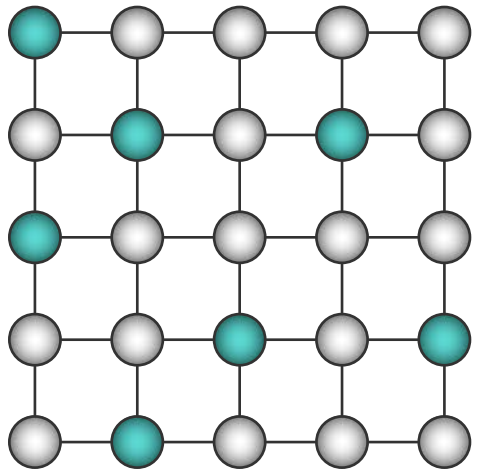
- We can used the brute-force gradient method as in LR

- But we can also apply generic laws by observing the $p(y|x)$ is an exponential family function, more specifically, a generalized linear model!

More examples: parameterizing graphical models

- Markov random fields

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\sum_{c\in C}\phi_c(\mathbf{x}_c)\right\} = \frac{1}{Z}\exp\{-H(\mathbf{x})\}$$

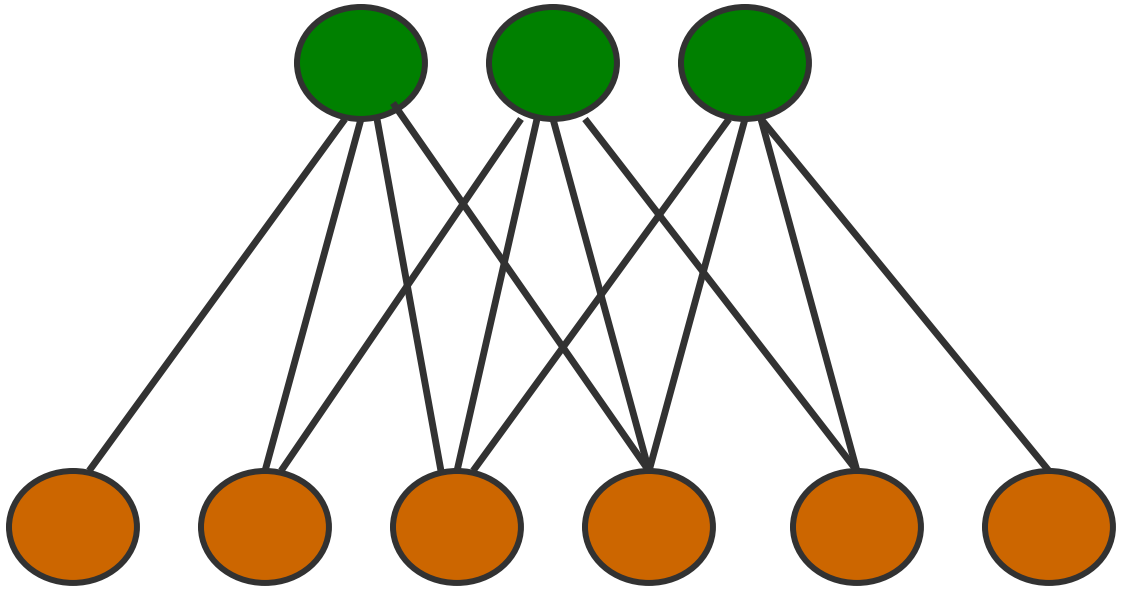


$$p(X) = \frac{1}{Z}\exp\left\{\sum_{i,j\in N_i}\theta_{ij}X_iX_j + \sum_i \theta_{i0}X_i\right\}$$

Restricted Boltzmann Machines

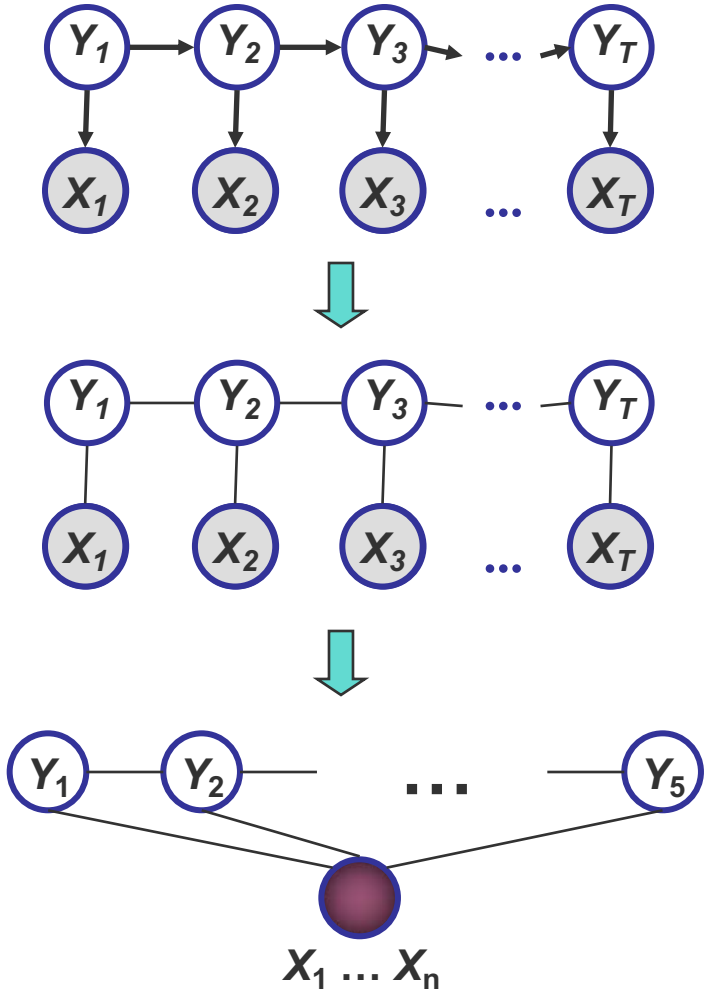
hidden units

visible units



$$p(x,h\,|\,\theta)=\exp\Big\{\sum_i\theta_i\phi_i(x_i)+\sum_j\theta_j\phi_j(h_j)+\sum_{i,j}\theta_{i,j}\phi_{i,j}(x_i,h_j)-A(\boldsymbol{\theta})\Big\}$$

Conditional Random Fields



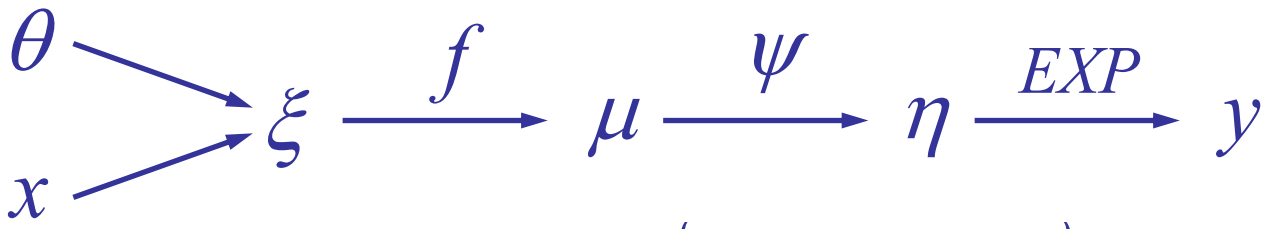
- Discriminative

$$p_\theta(y \mid x) = \frac{1}{Z(\theta, x)} \exp\left\{\sum_c \theta_c f_c(x, y_c)\right\}$$

- X_i's are assumed as features that are inter-dependent

- When labeling X_i future observations are taken into account

GLIM, cont.



$$p(y \mid \eta) = h(y) \exp\left\{\eta^T(x) y - A(\eta)\right\}$$

$$\Rightarrow p(y \mid \eta, \phi) = h(y, \phi) \exp\left\{\frac{1}{\phi}\left(\eta^T(x) y - A(\eta)\right)\right\}$$

- The choice of exp family is constrained by the nature of the data y
  - Example: y is a continuous vector → multivariate Gaussian
    y is a class label → Bernoulli or multinomial
- The choice of the response function
  - Following some mild constrains, e.g., [0,1]. Positivity …
  - Canonical response function:
    - In this case $\theta^T x$ directly corresponds to canonical parameter η. $f = \psi^{-1}(\cdot)$

Example canonical response functions

Model	Canonical response function
Gaussian	$\mu = \eta$
Bernoulli	$\mu = 1/(1+e^{-\eta})$
multinomial	$\mu_i = \eta_i / \sum_j e^{\eta_j}$
Poisson	$\mu = e^{\eta}$
gamma	$\mu = -\eta^{-1}$

MLE for GLIMs with natural response

- Log-likelihood

$$\ell = \sum_n \log h(y_n) + \sum_n \left(\theta^T x_n y_n - A(\eta_n)\right)$$

- Derivative of Log-likelihood

$$\frac{d\ell}{d\theta} = \sum_n \left(x_n y_n - \frac{dA(\eta_n)}{d\eta_n}\frac{d\eta_n}{d\theta} \right)$$
$$= \sum_n (y_n - \mu_n) x_n$$
$$= X^T (y - \mu)$$

This is a fixed point function because μ is a function of θ

- Online learning for canonical GLIMs
  - Stochastic gradient ascent:

$$\theta^{t+1} = \theta^t + \rho\left(y_n - \mu_n^t\right)x_n$$

where $\mu_n^t = \left(\theta^t\right)^T x_n$ and ρ is a step size

Batch learning for canonical GLIMs

- The Hessian matrix

$$H = \frac{d^2\ell}{d\theta d\theta^T} = \frac{d}{d\theta^T}\sum_n (y_n - \mu_n)x_n = \sum_n x_n \frac{d\mu_n}{d\theta^T}$$

$$= -\sum_n x_n \frac{d\mu_n}{d\eta_n}\frac{d\eta_n}{d\theta^T}$$

$$= -\sum_n x_n \frac{d\mu_n}{d\eta_n} x_n^T \quad \text{since } \eta_n = \theta^T x_n$$

$$= -X^T W X$$

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1 & -- \\ -- & \mathbf{x}_2 & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n & -- \end{bmatrix}$$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

where $X = \left[x_n^T\right]$ is the design matrix and

$$W = \text{diag}\left(\frac{d\mu_1}{d\eta_1}, \ldots, \frac{d\mu_N}{d\eta_N}\right)$$

which can be computed by calculating the 2nd derivative of $A(\eta_n)$

Iteratively Reweighted Least Squares (IRLS)

- Recall Newton-Raphson methods with cost function J

$$\theta^{t+1} = \theta^t - H^{-1}\nabla_\theta J$$

- We now have

$$\nabla_\theta J = X^T(y-\mu)$$

$$H = -X^T W X$$

- Now:

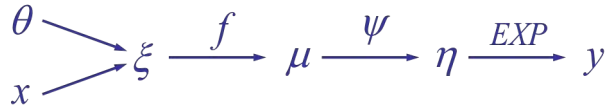
$$\begin{aligned}\theta^{t+1} &= \theta^t + H^{-1}\nabla_\theta \ell \\ &= \left(X^T W^t X\right)^{-1}\left[X^T W^t X\theta^t + X^T(y-\mu^t)\right] \\ &= \left(X^T W^t X\right)^{-1} X^T W^t z^t\end{aligned}$$

- $$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$

where the adjusted response is $z^t = X\theta^t + \left(W^t\right)^{-1}(y-\mu^t)$

- This can be understood as solving the following " Iteratively reweighted least squares " problem

$$\theta^{t+1} = \arg\min_\theta (z - X\theta)^T W (z - X\theta)$$



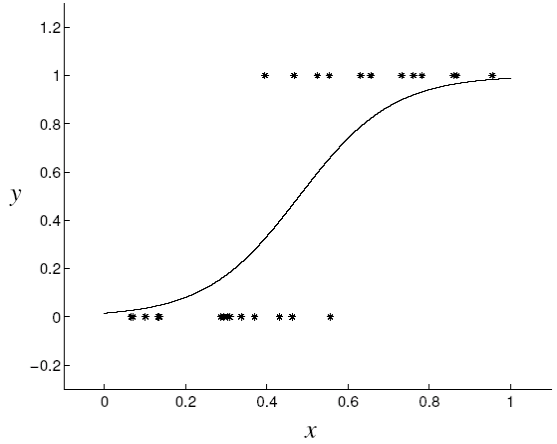
Example 1: logistic regression (sigmoid classifier)

- The condition distribution: a Bernoulli

$$p(y \mid x) = \mu(x)^y (1-\mu(x))^{1-y}$$

where μ is a logistic function

$$\mu(x) = \frac{1}{1+e^{-\eta(x)}}$$

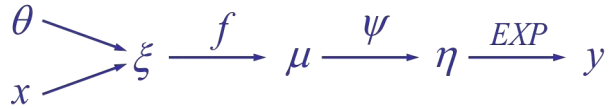


- $p(y|x)$ is an exponential family function, with
  - mean: $E[y \mid x] = \mu = \frac{1}{1+e^{-\eta(x)}}$
  - and canonical response function $\eta = \xi = \theta^T x$

- IRLS

$$\frac{d\mu}{d\eta} = \mu(1-\mu)$$

$$W = \begin{pmatrix} \mu_1(1-\mu_1) & & \\ & \ddots & \\ & & \mu_N(1-\mu_N) \end{pmatrix}$$



Example 2: linear regression

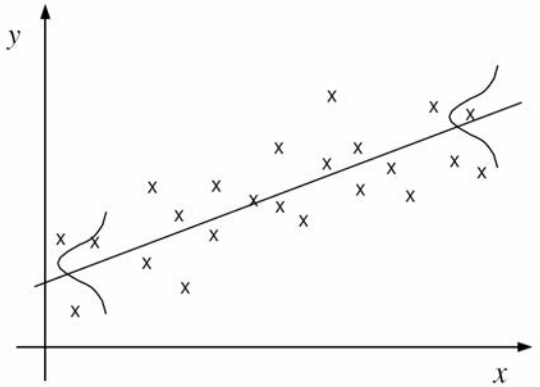
- The condition distribution: a Gaussian

$$p(y|x,\theta,\Sigma)=\frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}\exp\left\{-\frac{1}{2}(y-\mu(x))^T\Sigma^{-1}(y-\mu(x))\right\}$$

Rescale $\Rightarrow h(x)\exp\left\{-\frac{1}{2}\Sigma^{-1}\left(\eta^T(x)y-A(\eta)\right)\right\}$

where μ is a linear function

$$\mu(x)=\theta^T x=\eta(x)$$



- $p(y|x)$ is an exponential family function, with
  - mean: $E[y\,|\,x]=\mu=\theta^T x$
  - and canonical response function $\eta_1=\xi=\theta^T x$
- IRLS

$$\frac{d\mu}{d\eta}=1,\quad W=I \Rightarrow \begin{aligned}\theta^{t+1}&=\left(X^TW^tX\right)^{-1}X^TW^tz^t\\&=\left(X^TX\right)^{-1}X^T\left(X\theta^t+(y-\mu^t)\right)\\&=\theta^t+\left(X^TX\right)^{-1}X^T(y-\mu^t)\end{aligned}$$

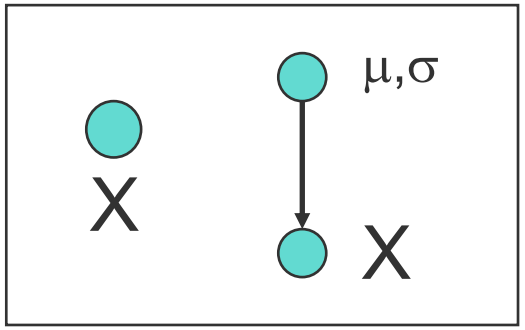
Steepest descent

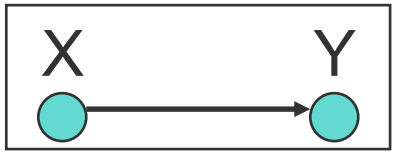
$$\overset{t\to\infty}{\Rightarrow}\quad \theta=(X^TX)^{-1}X^TY$$

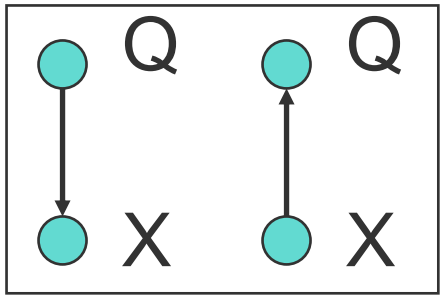
Normal equation

Simple GMs are the building blocks of complex GMs

- Density estimation
  - Parametric and nonparametric methods
- Regression
  - Linear, conditional mixture, nonparametric
- Classification
  - Generative and discriminative approach
- Clustering



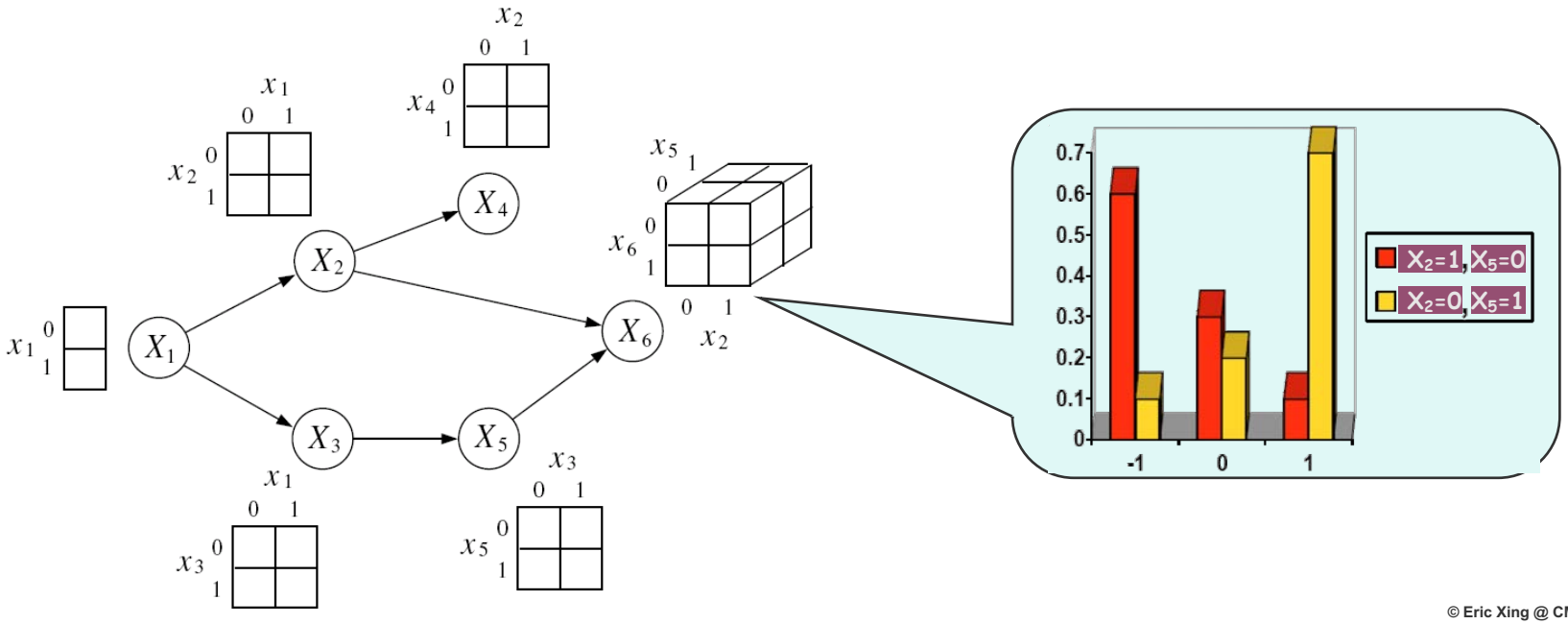




MLE for general BNs

- 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:

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

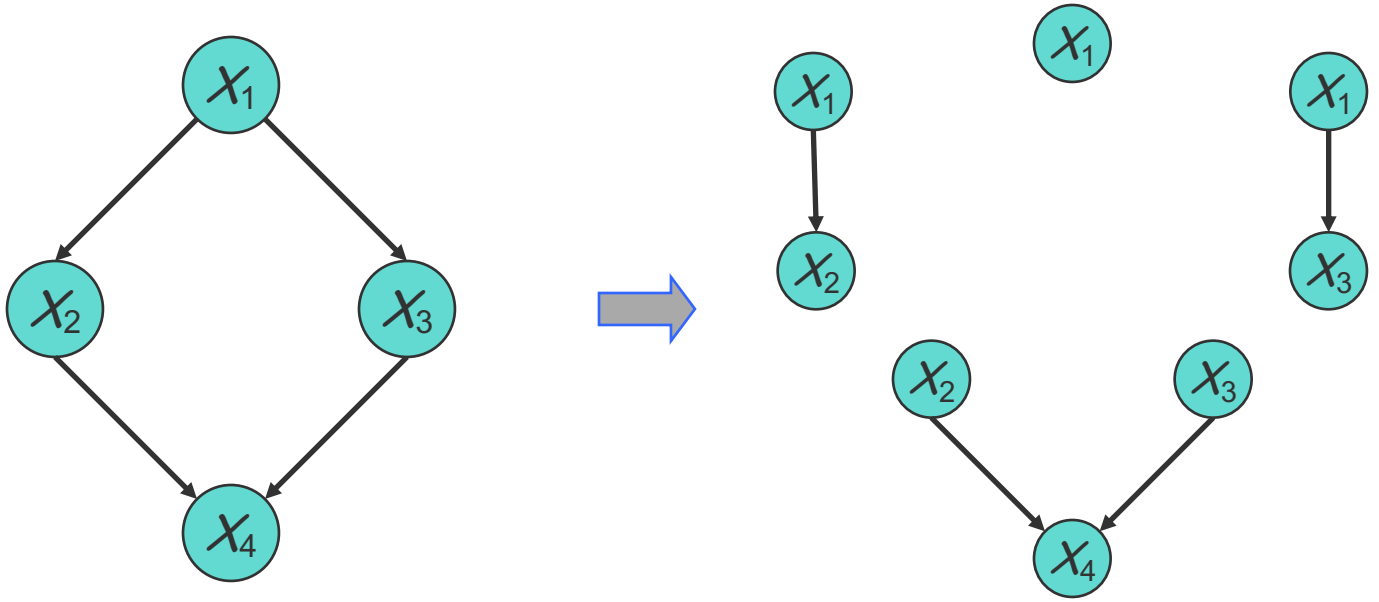


Decomposable likelihood of a BN

- Consider the distribution defined by the directed acyclic GM:

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

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

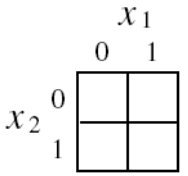


MLE for BNs with tabular CPDs

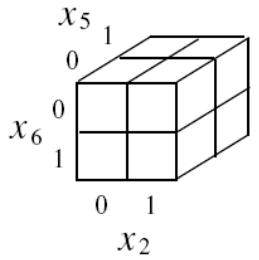
- Assume each CPD is represented as a table (multinomial) where

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

  - Note that in case of multiple parents, $\mathbf{X}_{\pi_i}$ will have a composite state, and the CPD will be a high-dimensional table
  - The sufficient statistics are counts of family configurations



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



- 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:

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

Summary: Learning GM

- For fully observed BN, the log-likelihood function decomposes into a sum of local terms, one per node; thus learning is also factored
  - 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

ML Parameter Est. for partially observed GMs: EM algorithm

Partially observed GMs

- Speech recognition

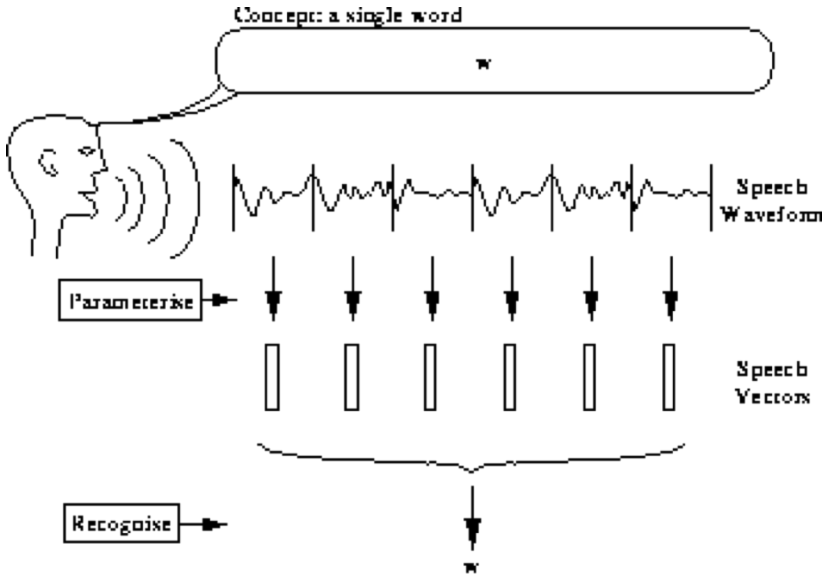
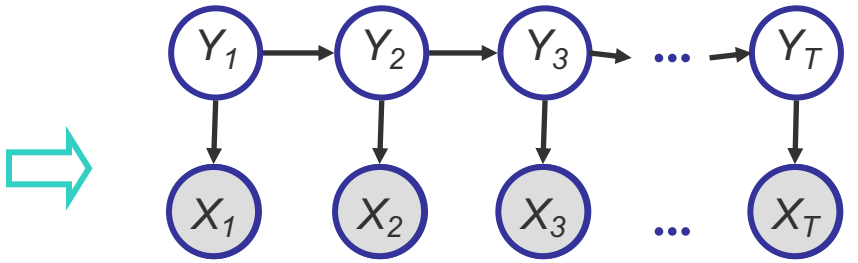
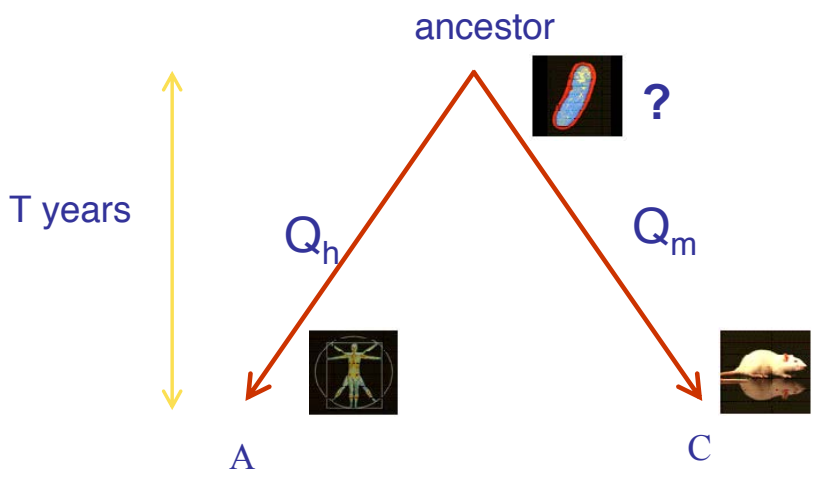


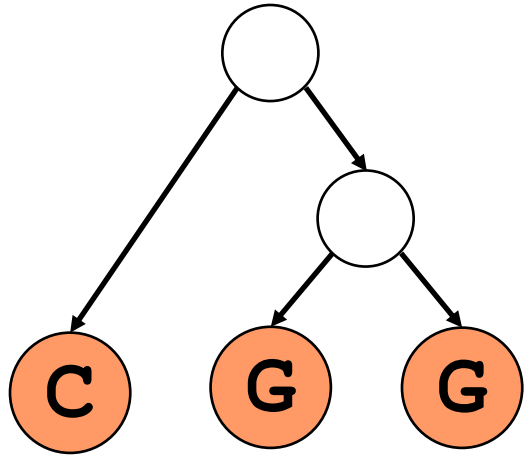
Fig. 1.2 Isolated Word Problem



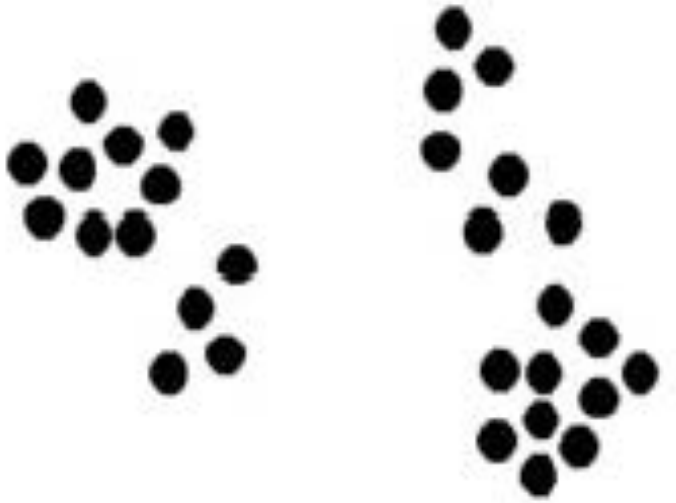
Partially observed GM

- Biological Evolution



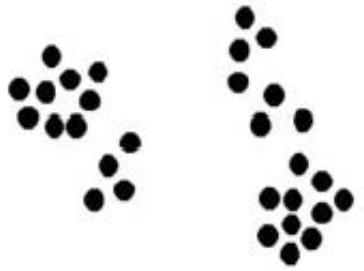


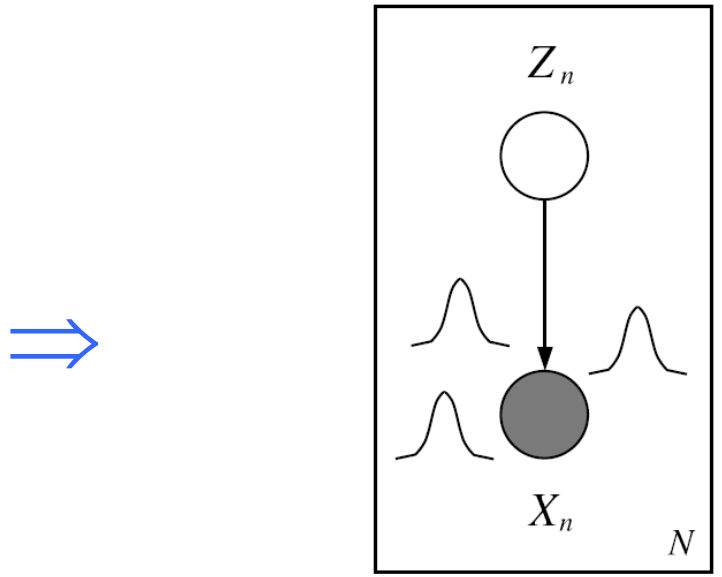
Mixture Models



Mixture Models, con'd

- A density model $p(x)$ 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

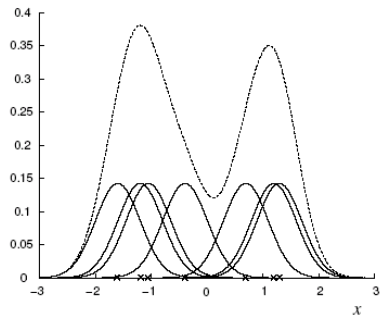
- A variable can be unobserved (latent) because:
  - it is an imaginary quantity meant to provide some simplified and abstractive view of the data generation process
    - e.g., speech recognition models, mixture models …
  - it is a real-world object and/or phenomena, but difficult or impossible to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

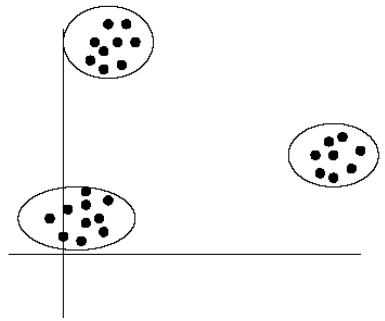
Gaussian Mixture Models (GMMs)

- 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



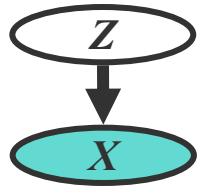


- This model can be used for unsupervised clustering.
  - This model (fit by AutoClass) has been used to discover new kinds of stars in astronomical data, etc.

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:

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

mixture component

mixture proportion

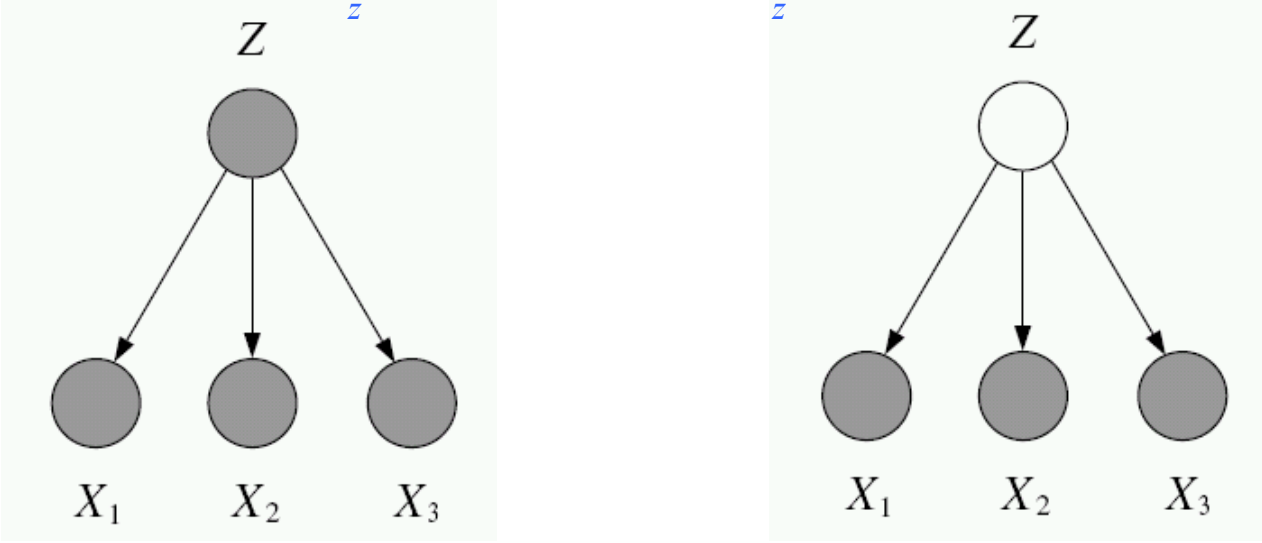
Why is Learning Harder?

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

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

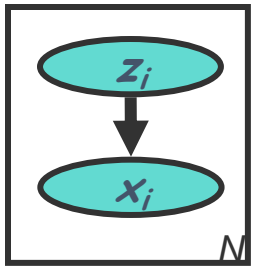
- With latent variables, all the parameters become coupled together via marginalization

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



Toward the EM algorithm

- Recall MLE for completely observed data



- Data log-likelihood

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

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

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

- MLE

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

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

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

- What if we do not know z_n?

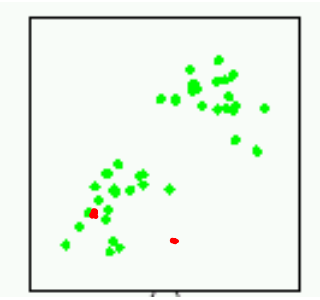
Question

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

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

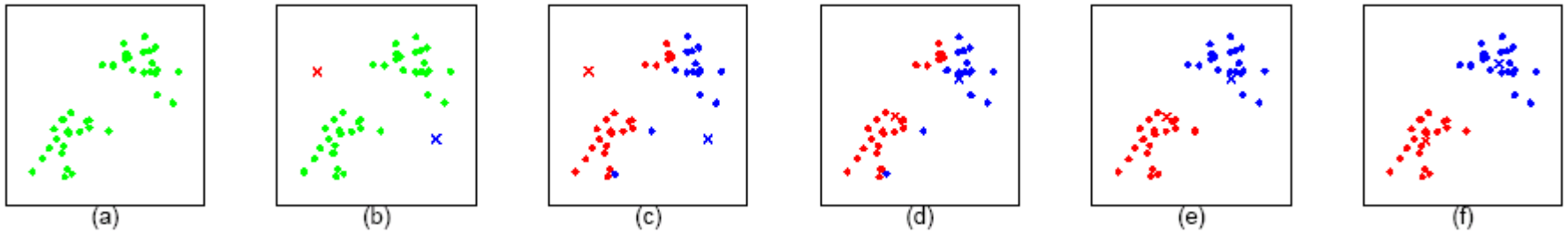
- M
  - What do we maximize?
  - What do we maximize with respect to?

Recall: K-means



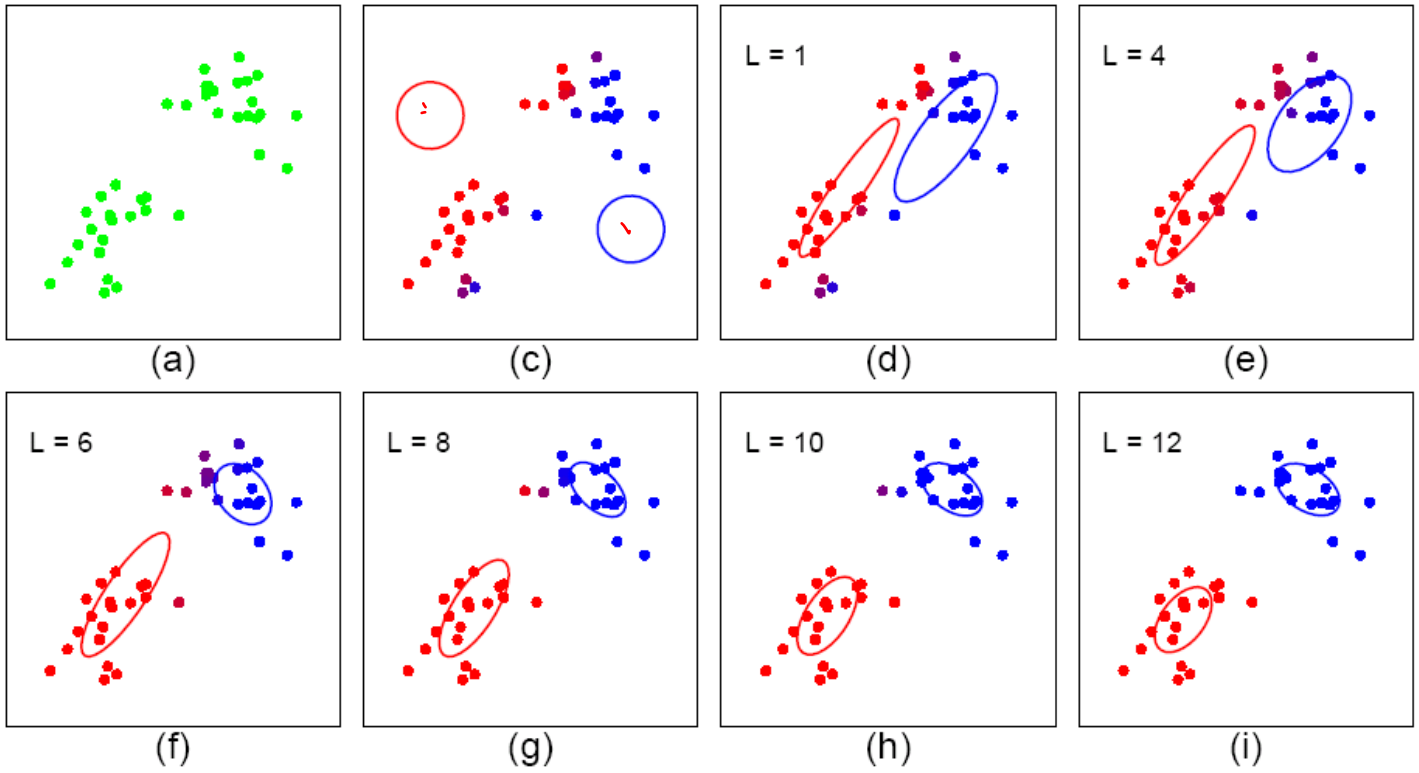
$$z_n^{(t)} = \arg\max_k (x_n - \mu_k^{(t)})^T \Sigma_k^{-1(t)} (x_n - \mu_k^{(t)})$$

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



Expectation-Maximization

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



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

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

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

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

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

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

$$\text{Fact}: \quad \frac{\partial \log |A^{-1}|}{\partial A^{-1}} = A^T \qquad \frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial A} = \mathbf{x}\mathbf{x}^T$$

  - This is isomorphic to MLE except that the variables that are hidden are replaced by their expectations (in general they will 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.

- EM
  - E-step

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

  - M-step

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

- K-means
  - In the K-means "E-step" we do hard assignment:

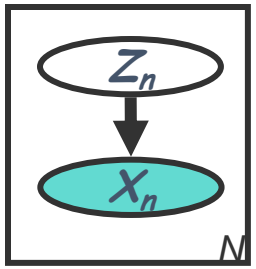
$$\tau_n^{k(t)} = \left\langle z_n^k \right\rangle_{q^{(t)}}$$

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

  - 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:

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

The EM Objective for 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:
$$p(x_n|\mu,\Sigma) = \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma)$$
$$= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k}\right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k)$$

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

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 $l_c()$ given both z and x is straightforward (c.f. MLE for fully observed models).
  - Recalled that in this case the objective for, e.g., MLE, decomposes into a sum of factors, the parameter for each factor can be estimated separately.
  - But given that Z is not observed, $l_c()$ is a random quantity, cannot be maximized directly.

- Incomplete log likelihood

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

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

  - This objective won't decouple

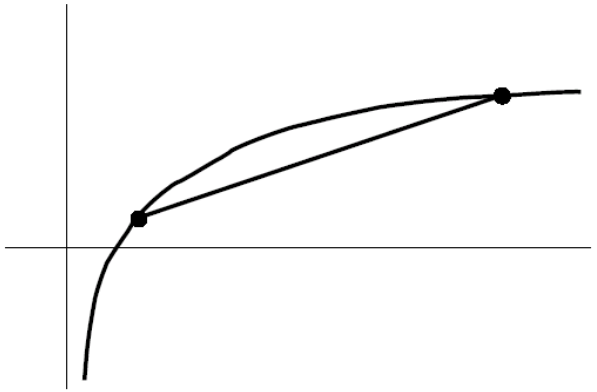
Expected Complete Log Likelihood

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

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

  - A deterministic function of θ
  - Linear in $l_c()$ --- inherit its factorizabiility
  - Does maximizing this surrogate yield a maximizer of the likelihood?
- Jensen's inequality

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

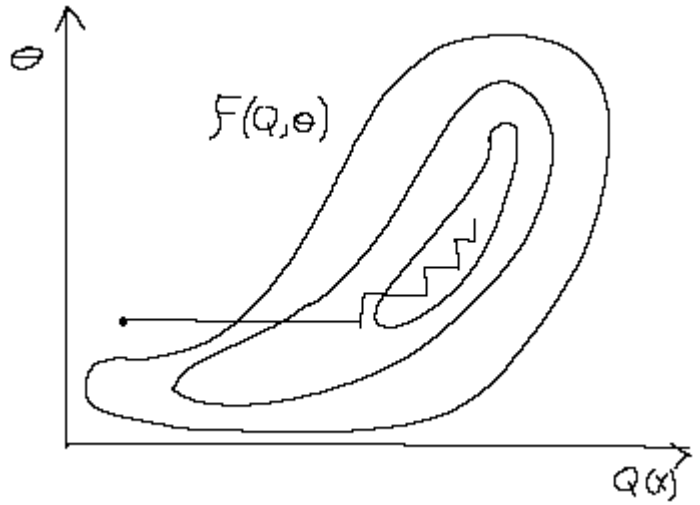


Lower Bounds and Free Energy

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

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

- 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: $$q^{t+1} = \arg\max_q F(q, \theta^t) = p(z \mid x, \theta^t)$$
  - 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 $l(\theta; x) \geq F(q, \theta)$

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

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

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

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

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

E-step ≡ plug in posterior expectation of latent variables

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

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

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

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

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

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

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:

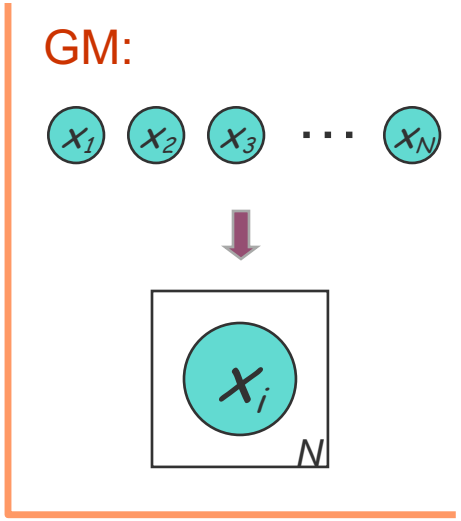
  - E-step: $q^{t+1} = \arg\max_q F(q, \theta^t)$
  - M-step: $\theta^{t+1} = \arg\max_\theta F(q^{t+1}, \theta^t)$

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

Supplementary materials

I: Review of density estimation

- Can be viewed as single-node graphical models
- Instances of exponential family dist.
- Building blocks of general GM
- MLE and Bayesian estimate



Discrete Distributions

- Bernoulli distribution: Ber(p)

$$P(x) = \begin{cases} 1-p & \text{for } x = 0 \\ p & \text{for } x = 1 \end{cases} \quad \Rightarrow \quad P(x) = p^x (1-p)^{1-x}$$

- Multinomial distribution: Mult(1, θ)
  - Multinomial (indicator) variable:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix}, \quad \text{where} \quad X_j = [0,1], \quad \text{and} \sum_{j \in [1,\ldots,6]} X_j = 1$$

$$X_j = 1 \text{ w.p. } \theta_j, \quad \sum_{j \in [1,\ldots,6]} \theta_j = 1 \ .$$

$$p(x(j)) = P\left(\{X_j = 1, \text{where } j \text{ index the dice-face}\}\right)$$
$$= \theta_j = \theta_A^{x_A} \times \theta_C^{x_C} \times \theta_G^{x_G} \times \theta_T^{x_T} = \prod_k \theta_k^{x_k} = \theta^x$$

Discrete Distributions

- Multinomial distribution: Mult(n, θ)
  - Count variable:

$$n = \begin{bmatrix} n_1 \\ \vdots \\ n_K \end{bmatrix}, \qquad \text{where} \sum_j n_j = N$$

$$p(n) = \frac{N!}{n_1! n_2! \cdots n_K!} \theta_1^{n_1} \theta_2^{n_2} \cdots \theta_K^{n_K} = \frac{N!}{n_1! n_2! \cdots n_K!} \theta^n$$

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

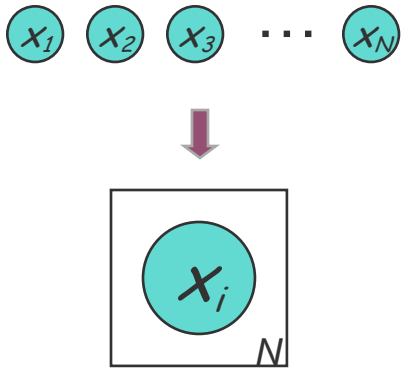
Example: multinomial model

- Data:
  - We observed N *iid* die rolls (K-sided): D={5, 1, K, …, 3}
- Representation:

  Unit basis vectors:

$$x_n = \begin{pmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,K} \end{pmatrix}, \quad \text{where } x_{n,k} = \{0,1\}, \text{ and } \sum_{k=1}^{K} x_{n,k} = 1$$

GM:



- Model:

$$X_{n,k} = 1 \text{ w.p. } \theta_k, \text{ and } \sum_{k \in \{1,\ldots K\}} \theta_k = 1$$

- How to write the likelihood of a single observation x_n?

$$P(x_i) = P(\{x_{n,k} = 1, \text{where } k \text{ index the die-side of the } n\text{th roll}\})$$

$$= \theta_k = \theta_1^{x_{n,1}} \times \theta_2^{x_{n,2}} \times \cdots \times \theta_K^{x_{n,K}} = \prod_{k=1}^{K} \theta_k^{x_{n,k}}$$

- The likelihood of dataset D={x_1, …,x_N}:

$$P(x_1, x_2, \ldots, x_N \mid \theta) = \prod_{n=1}^{N} P(x_n \mid \theta) = \prod_{n=1}^{N} \left(\prod_k \theta_k^{x_{n,k}} \right) = \prod_k \theta_k^{\sum_{n=1}^{N} x_{n,k}} = \prod_k \theta_k^{n_k}$$

MLE: constrained optimization with Lagrange multipliers

- Objective function:

$$\ell(\theta; D) = \log P(D \mid \theta) = \log \prod_k \theta_k^{n_k} = \sum_k n_k \log \theta_k$$

- We need to maximize this subject to the constrain $\sum_{k=1}^{K} \theta_k = 1$
- Constrained cost function with a Lagrange multiplier

$$\ell^- = \sum_k n_k \log \theta_k + \lambda \left(1 - \sum_{k=1}^{K} \theta_k\right)$$

- Take derivatives wrt θ_k

$$\frac{\partial \ell}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0$$

$$n_k = \lambda \theta_k \Rightarrow \sum_k n_k = N = \lambda \sum_k \theta_k = \lambda$$

$$\Rightarrow \hat{\theta}_{k,MLE} = \frac{n_k}{N} \quad \text{or} \quad \hat{\theta}_{k,MLE} = \frac{1}{N} \sum_n x_{n,k}$$

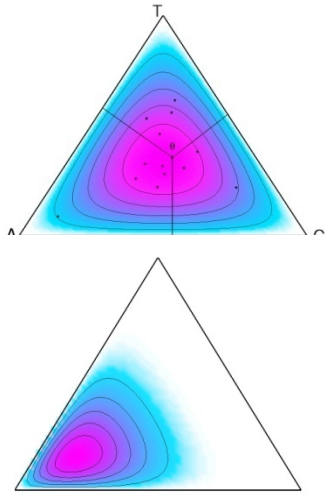
Frequency as sample mean

- Sufficient statistics
  - The counts, $\vec{n} = (n_1, \cdots, n_K), n_k = \sum_n x_{n,k}$, are sufficient statistics of data D

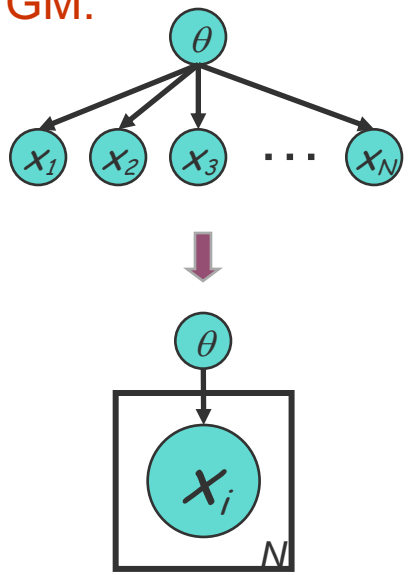
Bayesian estimation:

- Dirichlet distribution:

$$P(\theta)=\frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)}\prod_k \theta_k^{\alpha_k-1}=C(\alpha)\prod_k \theta_k^{\alpha_k-1}$$



GM:



- Posterior distribution of θ:

$$P(\theta \mid x_1,\ldots,x_N)=\frac{p(x_1,\ldots,x_N \mid \theta)p(\theta)}{p(x_1,\ldots,x_N)} \propto \prod_k \theta_k^{n_k}\prod_k \theta_k^{\alpha_k-1}=\prod_k \theta_k^{\alpha_k+n_k-1}$$

  - Notice the isomorphism of the posterior to the prior,
  - such a prior is called a conjugate prior

- Posterior mean estimation:

$$\theta_k=\int \theta_k p(\theta \mid D)d\theta = C\int \theta_k \prod_k \theta_k^{\alpha_k+n_k-1}d\theta=\frac{n_k+\alpha_k}{N+|\alpha|}$$

Dirichlet parameters can be understood as pseudo-counts

More on Dirichlet Prior:

- Where is the normalize constant $\mathcal{C}(\alpha)$ come from?

$$\frac{1}{C(\alpha)} = \int \cdots \int \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1} d\theta_1 \cdots d\theta_K = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma\left(\sum_k \alpha_k\right)}$$

  - Integration by parts
  - $\Gamma(\alpha)$ is the gamma function: $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$
  - For inregers, $\Gamma(n+1) = n!$
- Marginal likelihood:

$$p(\{x_1, \ldots, x_N\} \mid \vec{\alpha}) = p(\vec{n} \mid \vec{\alpha}) = \int p(\vec{n} \mid \vec{\theta}) p(\vec{\theta} \mid \vec{\alpha}) d\vec{\theta} = \frac{C(\vec{\alpha})}{C(\vec{n} + \vec{\alpha})}$$

- Posterior in closed-form:

$$P(\vec{\theta} \mid \{x_1, \ldots, x_N\}, \vec{\alpha}) = \frac{p(\vec{n} \mid \theta) p(\theta \mid \vec{\alpha})}{p(\vec{n} \mid \vec{\alpha})} = C(\vec{n} + \vec{\alpha}) \prod_k \theta_k^{\alpha_k + n_k - 1} = \mathrm{Dir}(\vec{n} + \vec{\alpha})$$

- Posterior predictive rate:

$$p(x_{N+1} = i \mid \{x_1, \ldots, x_N\}, \vec{\alpha}) = \int C(\vec{n} + \vec{\alpha}) \prod_k \theta_k^{\alpha_k + n_k - 1} \times \theta_i^{\alpha_k + n_k} d\vec{\theta} = \frac{C(\vec{n} + \vec{\alpha})}{C(\vec{n} + \vec{\alpha} + x_N)} = \frac{n_i + \alpha_i}{|\vec{n}| + |\vec{\alpha}|}$$

Sequential Bayesian updating

- Start with Dirichlet prior $P(\vec{\theta} \mid \vec{\alpha}) = \mathrm{Dir}(\vec{\theta} : \vec{\alpha})$
- Observe N' samples with sufficient statistics $\vec{n}'$. Posterior becomes:

$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}')$$

- Observe another N'' samples with sufficient statistics $\vec{n}''$. Posterior becomes:

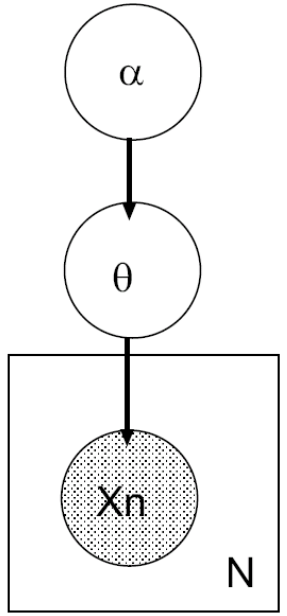
$$P(\vec{\theta} \mid \vec{\alpha}, \vec{n}', \vec{n}'') = \mathrm{Dir}(\vec{\theta} : \vec{\alpha} + \vec{n}' + \vec{n}'')$$

- So sequentially absorbing data in any order is equivalent to batch update.

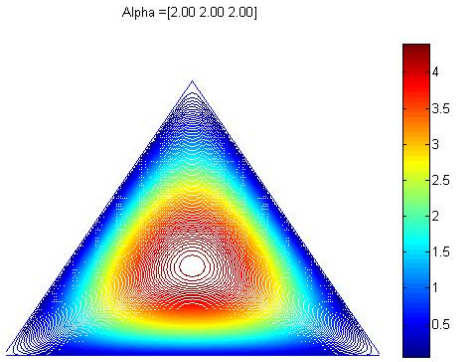
Hierarchical Bayesian Models

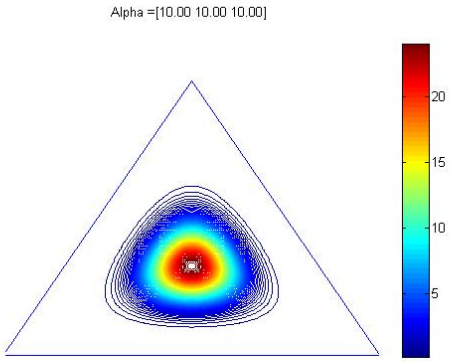
- θ are the parameters for the likelihood $p(x|\theta)$
- α are the parameters for the prior $p(\theta|\alpha)$.
- We can have hyper-hyper-parameters, etc.
- We stop when the choice of hyper-parameters makes no difference to the marginal likelihood; typically make hyper-parameters constants.
- Where do we get the prior?
  - Intelligent guesses
  - Empirical Bayes (Type-II maximum likelihood)
    → computing point estimates of α :

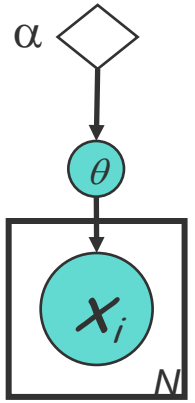
$$\hat{\vec{\alpha}}_{MLE} = \arg\max_{\vec{\alpha}} = p(\vec{n} \mid \vec{\alpha})$$

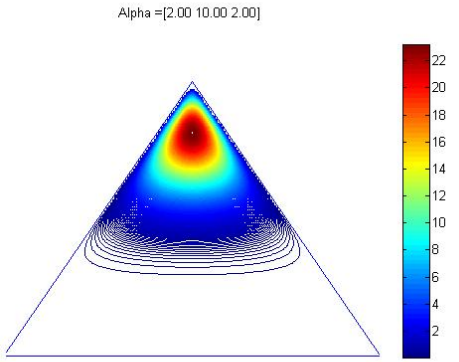


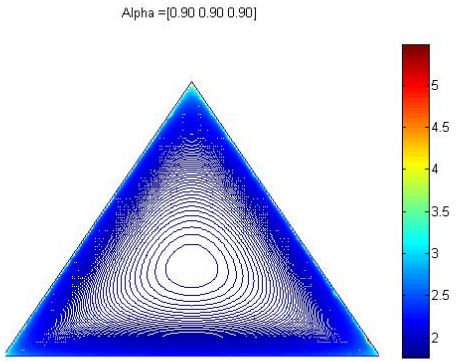
Limitation of Dirichlet Prior:











The Logistic Normal Prior

$$\theta \sim LN_K(\mu, \Sigma)$$

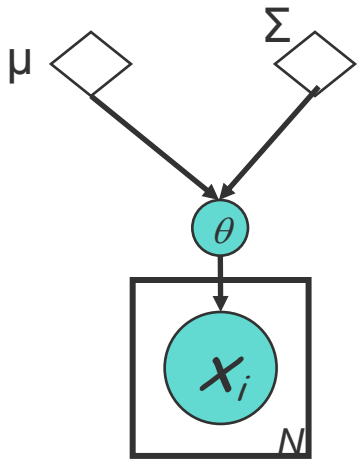
$$\gamma \sim N_{K-1}(\mu, \Sigma) \qquad \gamma_K = 0$$

$$\theta_i = \exp\left\{\gamma_i - \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)\right\}$$

$$C(\gamma) = \log\left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)$$

Problem

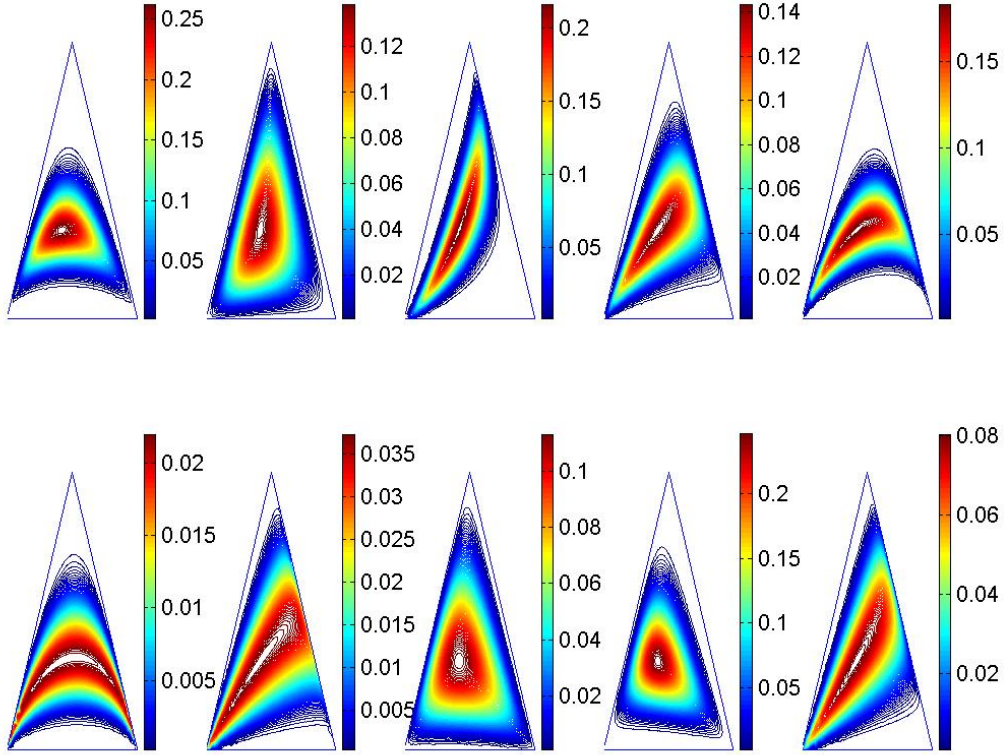
- Log Partition Function
- Normalization Constant



- **Pro: co-variance structure**
- **Con: non-conjugate (we will discuss how to solve this later)**

Logistic Normal Densities

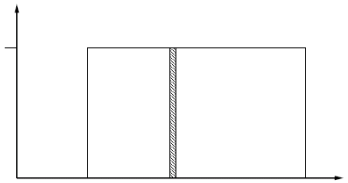
Logistic Normal



Continuous Distributions

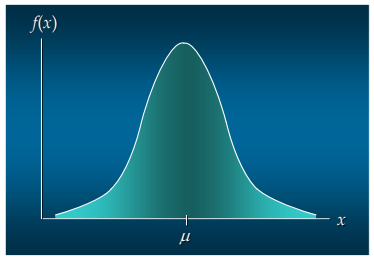
- Uniform Probability Density Function

$$p(x) = 1/(b-a) \quad \text{for } a \le x \le b$$
$$= 0 \quad \text{elsewhere}$$



- Normal (Gaussian) Probability Density Function

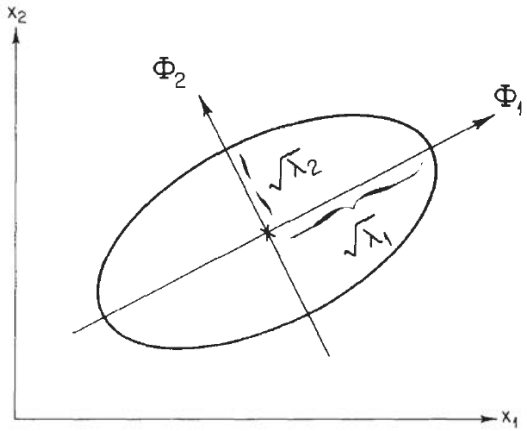
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$



  - The distribution is symmetric, and is often illustrated as a bell-shaped curve.
  - Two parameters, μ (mean) and σ (standard deviation), determine the location and shape of the distribution.
  - The highest point on the normal curve is at the mean, which is also the median and mode.
  - The mean can be any numerical value: negative, zero, or positive.

- Multivariate Gaussian

$$p(X;\vec{\mu},\Sigma) = \frac{1}{\left(\sqrt{2\pi}\right)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(X-\vec{\mu})^T\Sigma^{-1}(X-\vec{\mu})\right\}$$



MLE for a multivariate-Gaussian

- It can be shown that the MLE for μ *and* Σ is

$$\mu_{MLE} = \frac{1}{N}\sum_n (x_n)$$

$$\Sigma_{MLE} = \frac{1}{N}\sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \frac{1}{N} S$$

$$x_n = \begin{pmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,K} \end{pmatrix}$$

$$X = \begin{pmatrix} --- x_1^T --- \\ --- x_2^T --- \\ \vdots \\ --- x_N^T --- \end{pmatrix}$$

where the scatter matrix is

$$S = \sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum_n x_n x_{n\,n}^T\right) - N\mu_{ML}\mu_{ML}^T$$

- The sufficient statistics are $\Sigma_n x_n$ and $\Sigma_n x_n x_n^T$.
- Note that $X^T X = \Sigma_n x_n x_n^T$ may not be full rank (eg. if $N < D$), in which case Σ_{ML} is not invertible

Bayesian parameter estimation for a Gaussian

- There are various reasons to pursue a Bayesian approach
  - We would like to update our estimates sequentially over time.
  - We may have prior knowledge about the expected magnitude of the parameters.
  - The MLE for Σ may not be full rank if we don't have enough data.
- We will restrict our attention to conjugate priors.
- We will consider various cases, in order of increasing complexity:
  - Known σ, unknown μ
  - Known μ, unknown σ
  - Unknown μ and σ

Bayesian estimation: unknown μ, known σ

- Normal Prior:

$$P(\mu) = (2\pi\tau^2)^{-1/2} \exp\{-(\mu-\mu_0)^2/2\tau^2\}$$

- Joint probability:

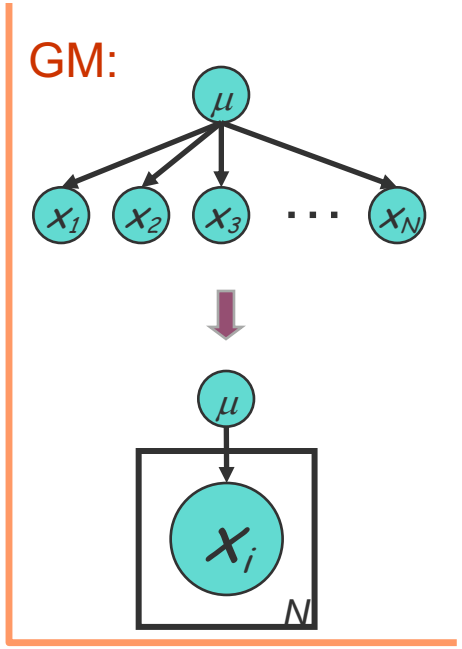
$$P(x,\mu) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\}$$
$$\times (2\pi\tau^2)^{-1/2} \exp\{-(\mu-\mu_0)^2/2\tau^2\}$$

- Posterior:

$$P(\mu \mid x) = (2\pi\tilde{\sigma}^2)^{-1/2} \exp\{-(\mu-\tilde{\mu})^2/2\tilde{\sigma}^2\}$$

where $\tilde{\mu} = \frac{N/\sigma^2}{N/\sigma^2+1/\tau^2}\bar{x} + \frac{1/\tau^2}{N/\sigma^2+1/\tau^2}\mu_0$, and $\tilde{\sigma}^2 = \left(\frac{N}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}$

Sample mean

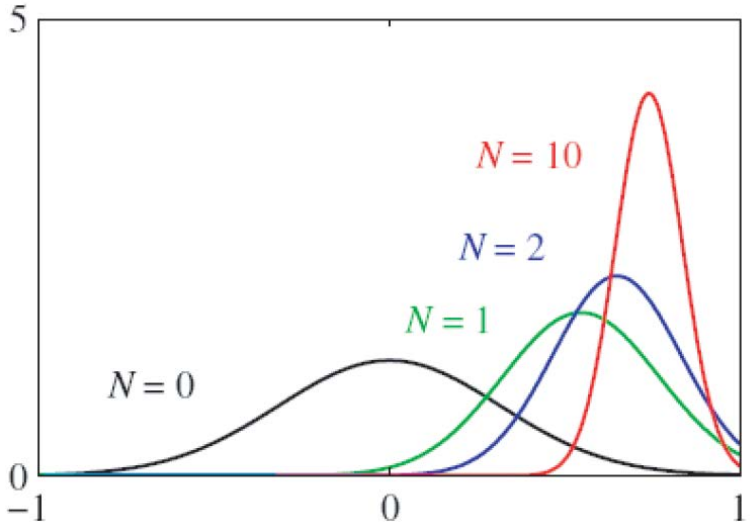


Bayesian estimation: unknown μ, known σ

$$\mu_N = \frac{N/\sigma^2}{N/\sigma^2 + 1/\sigma_0^2}\bar{x} + \frac{1/\sigma_0^2}{N/\sigma^2 + 1/\sigma_0^2}\mu_0\,, \qquad \tilde{\sigma}^2 = \left(\frac{N}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1}$$

- The posterior mean is a convex combination of the prior and the MLE, with weights proportional to the relative noise levels.
- The precision of the posterior $1/\sigma^2_N$ is the precision of the prior $1/\sigma^2_0$ plus one contribution of data precision $1/\sigma^2$ for each observed data point.

- Sequentially updating the mean
  - $\mu* = 0.8$ (unknown), $(\sigma^2)* = 0.1$ (known)
  - Effect of single data point $\mu_1 = \mu_0 + (x - \mu_0)\frac{\sigma_0^2}{\sigma^2 + \sigma_0^2} = x - (x - \mu_0)\frac{\sigma_0^2}{\sigma^2 + \sigma_0^2}$
  - Uninformative (vague/ flat) prior, $\sigma^2_0 \to \infty$ $\mu_N \to \mu_0$



Other scenarios

- Known μ, unknown $\lambda = 1/\sigma_2$
  - The conjugate prior for λ is a Gamma with shape a_0 and rate (inverse scale) b_0

$$p(\lambda|a,b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda)$$

  - The conjugate prior f

$$IG(\sigma^2|a,b) = \frac{1}{\Gamma(a)} b^a (\sigma^2)^{-(a+1)} \exp(-b/(\sigma^2))$$

- Unknown μ and unknown σ_2
  - The conjugate prior is Normal-Inverse-Gamma

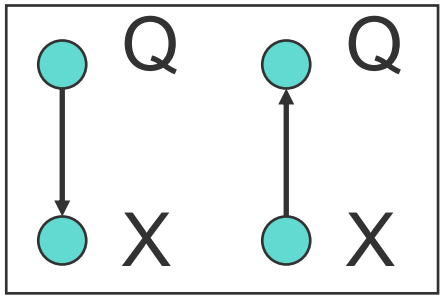
$$\begin{aligned} P(\mu,\sigma^2) &= P(\mu|\sigma^2)P(\sigma^2) \\ &= \mathcal{N}(\mu|m,\sigma^2 V)\ IG(\sigma^2|a,b) \end{aligned}$$

  - Semi conjugate prior
- Multivariate case:
  - The conjugate prior is Normal-Inverse-Wishart

$$\begin{aligned} P(\mu,\Sigma) &= P(\mu|\Sigma)P(\Sigma) \\ &= \mathcal{N}(\mu|\mu_0,\frac{1}{\kappa_0}\Sigma)\ \mathcal{IW}(\Sigma|\Lambda_0^{-1},\nu_0) \end{aligned}$$

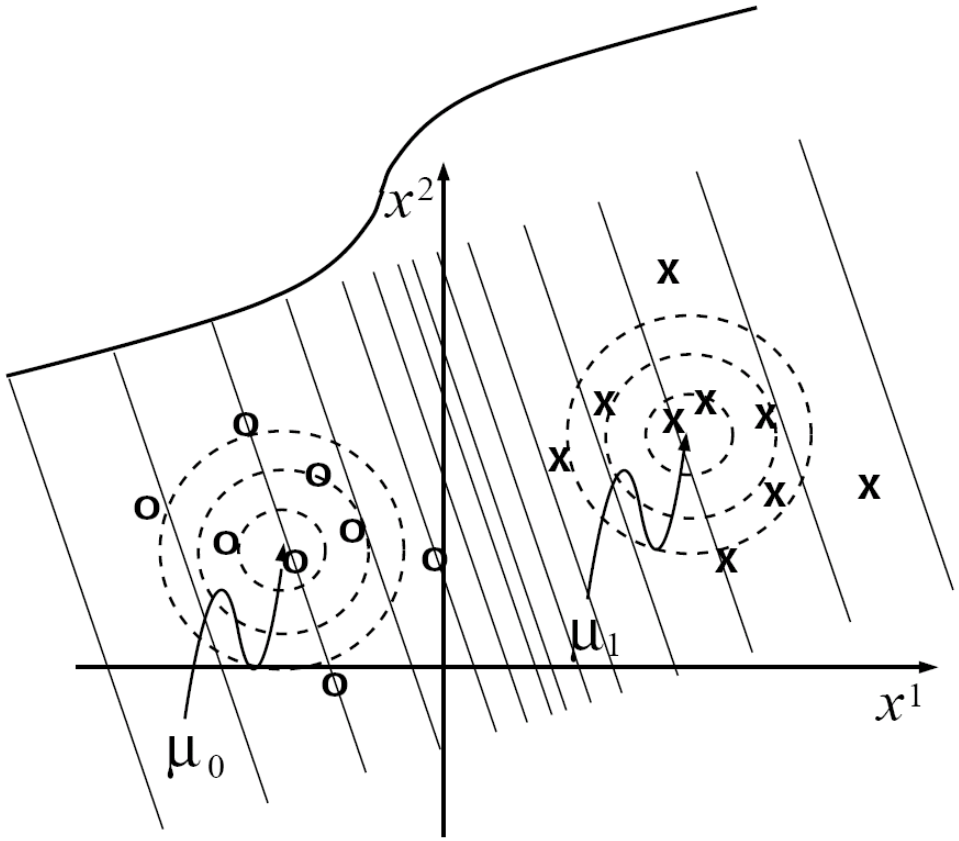
II: Two node fully observed BNs

- Conditional mixtures
- Linear/Logistic Regression
-
- Classification
  - Generative and discriminative approaches



Classification:

- Goal: Wish to learn f: X → Y
- Generative:
  - Modeling the joint distribution of all data
- Discriminative:
  - Modeling only points at the boundary

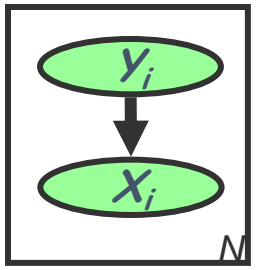


Conditional Gaussian

- The data:

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_N, y_N)\}$$

GM:



- Both nodes are observed:
  - y is a class indicator vector

$$p(y_n) = \text{multi}(y_n : \pi) = \prod_k \pi_k^{y_{n,k}}$$

  - x is a conditional Gaussian variable with a class-specific mean

$$p(x_n \mid y_{n,k} = 1, \mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(x_n - \mu_k)^2 \right\}$$

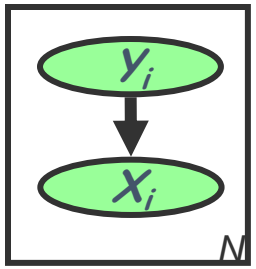
$$p(x \mid y, \mu, \sigma) = \prod_n \left(\prod_k N(x_n : \mu_k, \sigma)^{y_{n,k}} \right)$$

MLE of conditional Gaussian

- Data log-likelihood

$$\ell(\boldsymbol{\theta}; D) = \log \prod_n p(x_n, y_n) = \log \prod_n p(y_n \mid \pi) p(x_n \mid y_n, \mu, \sigma)$$

GM:



- MLE

$$\hat{\pi}_{k,MLE} = arg\max_{\pi} \ell(\boldsymbol{\theta}; D), \qquad \hat{\pi}_{k,MLE} = \sum_n y_{n,k} \Big/ N = n_k / N$$

the fraction of samples of class *m*

$$\hat{\mu}_{k,MLE} = \arg\max \ell(\boldsymbol{\theta}; D), \qquad \hat{\mu}_{k,MLE} = \frac{\sum_n y_{n,k} x_n}{\sum_n y_{n,k}} = \frac{\sum_n y_{n,k} x_n}{n_k}$$

the average of samples of class *m*

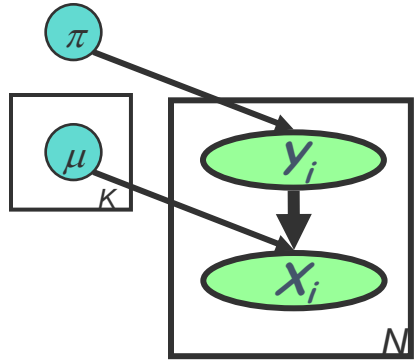
Bsyesian estimation of conditional Gaussian

- Prior:

$$P(\vec{\pi} \mid \vec{\alpha}) = \mathrm{Dir}(\vec{\pi} : \vec{\alpha})$$

$$P(\mu_k \mid \nu) = \mathrm{Normal}(\mu_k : \nu, \tau)$$

GM:



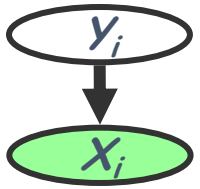
- Posterior mean (Bayesian est.)

$$\pi_{k,Bayes} = \frac{N}{N+|\alpha|}\hat{\pi}_{k,ML} + \frac{|\alpha|}{N+|\alpha|}\frac{\alpha_k}{|\alpha|} = \frac{n_k+\alpha_k}{N+|\alpha|}$$

$$\mu_{k,Bayes} = \frac{n_k/\sigma^2}{n_k/\sigma^2+1/\tau^2}\hat{\mu}_{k,ML} + \frac{1/\tau^2}{n_k/\sigma^2+1/\tau^2}\nu, \quad \text{and} \quad \sigma^2_{Bayes} = \left(\frac{N}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}$$

Classification

- Gaussian Discriminative Analysis:
  - The joint probability of a datum and it label is:



$$p(x_n, y_{n,k}=1 \mid \mu, \sigma) = p(y_{n,k}=1) \times p(x_n \mid y_{n,k}=1, \mu, \sigma)$$
$$= \pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x_n - \mu_k)^2\right\}$$

  - Given a datum x_n, we predict its label using the conditional probability of the label given the datum:

$$p(y_{n,k}=1 \mid x_n, \mu, \sigma) = \frac{\pi_k \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x_n - \mu_k)^2\right\}}{\sum_{k'} \pi_{k'} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x_n - \mu_{k'})^2\right\}}$$

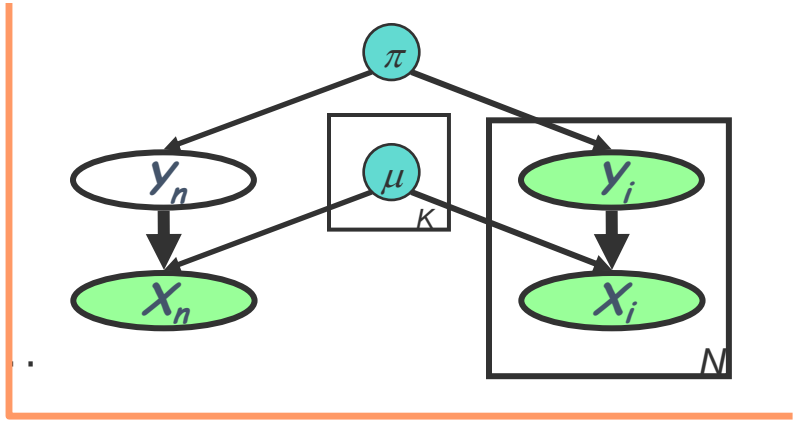
- This is basic inference
  - introduce evidence, and then normalize

Transductive classification

- Given X_n, what is its corresponding Y_n when we know the answer for a set of training data?

- Frequentist prediction:
  - we fit π, μ and σ from data first, and then …

GM:



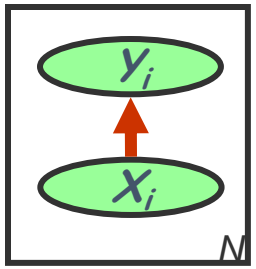
$$p(y_{n,k}=1 \mid x_n,\mu,\sigma,\pi)=\frac{p(y_{n,k}=1,x_n \mid \mu,\sigma,\pi)}{p(x_n \mid \mu,\sigma,\pi)}=\frac{\pi_k N(x_n,\mid \mu_k,\sigma)}{\sum_{k'}\pi_{k'}N(x_n,\mid \mu_{k'},\sigma)}$$

- Bayesian:
  - we compute the posterior dist. of the parameters first …

Linear Regression

- The data:

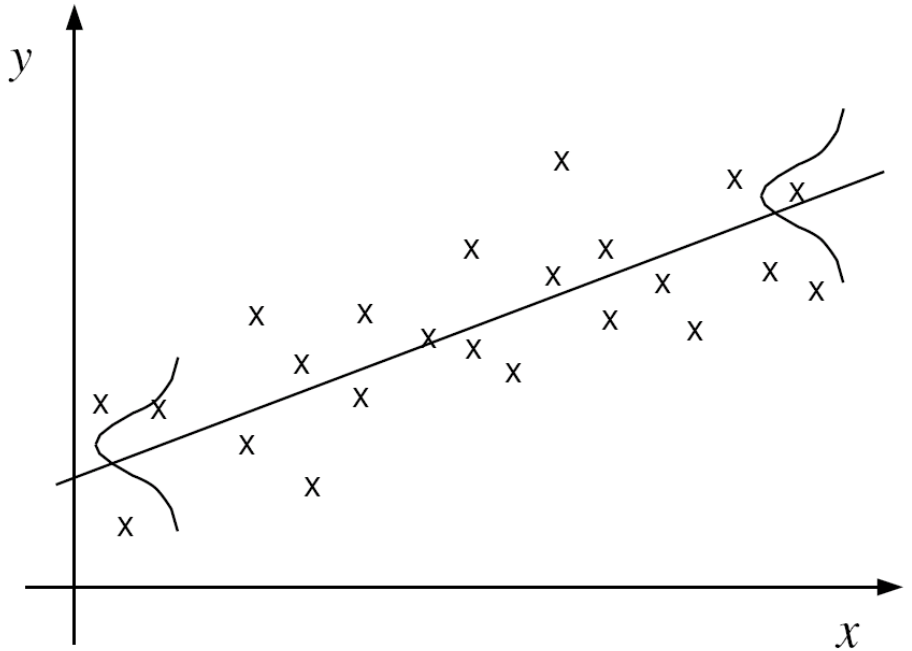
$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_N, y_N)\}$$



- Both nodes are observed:
  - X is an input vector
  - Y is a response vector

    (we first consider y as a generic continuous response vector, then we consider the special case of classification where y is a discrete indicator)

- A regression scheme can be used to model $p(y|x)$ directly, rather than $p(x,y)$



A discriminative probabilistic model

- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

where ε is an error term of unmodeled effects or random noise

- Now assume that ε follows a Gaussian $N(0,\sigma)$, then we have:

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

- By independence assumption:

$$L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i-\theta^T\mathbf{x}_i)^2}{2\sigma^2}\right)$$

Linear regression

- Hence the log-likelihood is:

$$l(\theta) = n \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta^T \mathbf{x}_i)^2$$

- Do you recognize the last term?

Yes it is:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (\mathbf{x}_i{}^T \theta - y_i)^2$$

- It is same as the MSE!

A recap:

- LMS update rule

$$\theta^{t+1} = \theta^t + \alpha(y_n - \mathbf{x}_n{}^T\theta^t)\mathbf{x}_n$$

  - Pros: on-line, low per-step cost
  - Cons: coordinate, maybe slow-converging
- Steepest descent

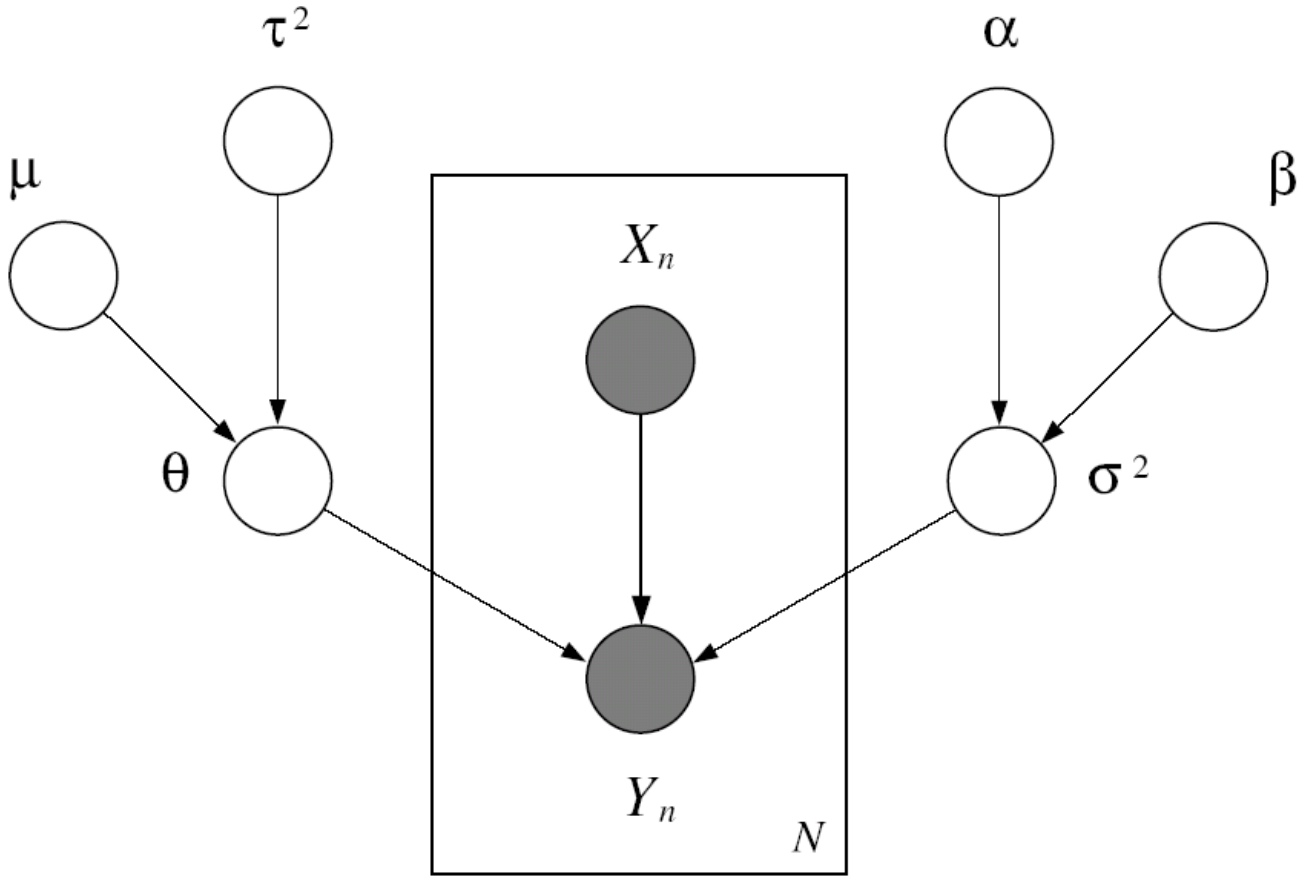
$$\theta^{t+1} = \theta^t + \alpha\sum_{i=1}^{n}(y_n - \mathbf{x}_n{}^T\theta^t)\mathbf{x}_n$$

  - Pros: fast-converging, easy to implement
  - Cons: a batch,
- Normal equations

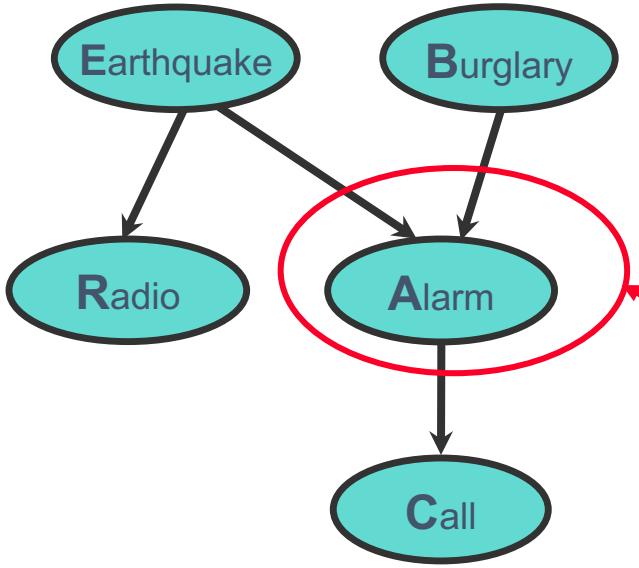
$$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$

  - Pros: a single-shot algorithm! Easiest to implement.
  - Cons: need to compute pseudo-inverse (XTX)-1, expensive, numerical issues (e.g., matrix is singular ..)

Bayesian linear regression



III: How to define parameter prior for general BN?



Factorization: $p(\mathbf{X}=\mathbf{x}) = \prod_{i=1}^{M} p(x_i \mid \mathbf{x}_{\pi_i})$

Local Distributions
defined by, e.g., multinomial parameters:

$$p(x_i^k \mid \mathbf{x}_{\pi_i}^j) = \theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j}$$

Assumptions (Geiger & Heckerman 97,99):

- Complete Model Equivalence
- Global Parameter Independence
- Local Parameter Independence
- Likelihood and Prior Modularity

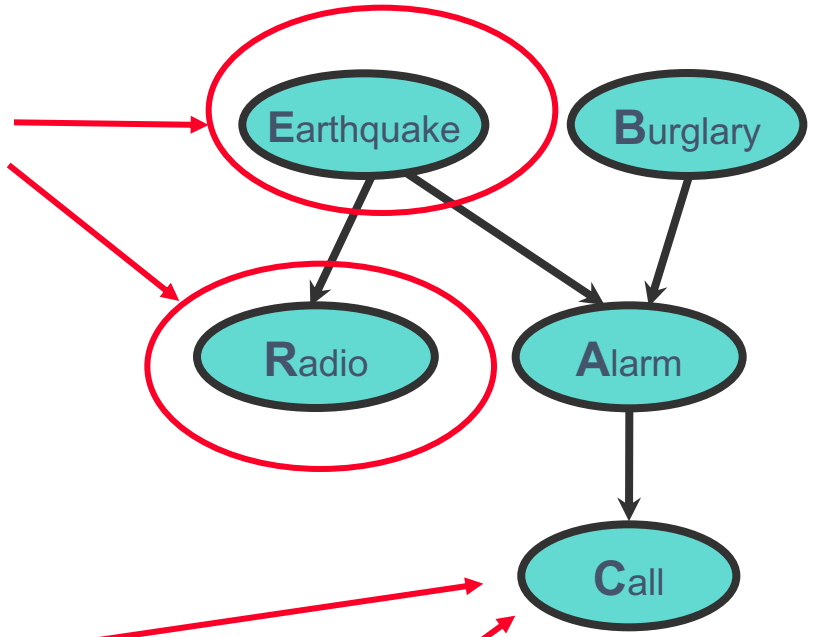
$$p(\theta \mid G)\ ?$$

Global & Local Parameter Independence

- Global Parameter Independence

  For every DAG model:

$$p(\theta_m \mid G) = \prod_{i=1}^{M} p(\theta_i \mid G)$$



- Local Parameter Independence

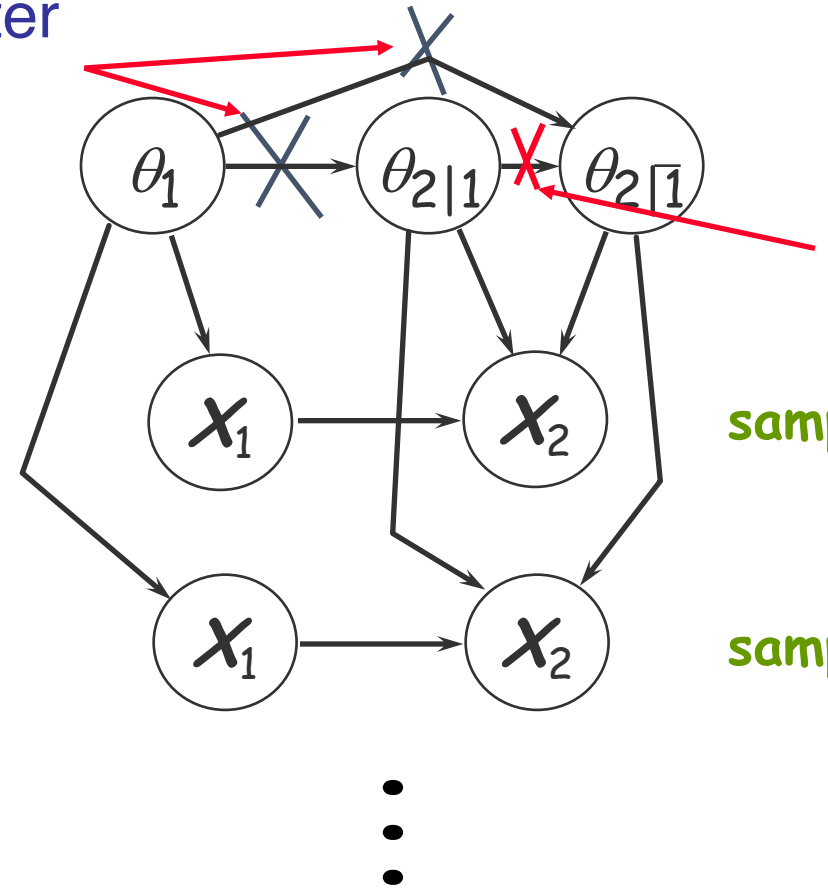
  For every node:

$$p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j} \mid G)$$

$P(\theta_{Call|Alarm=YES})$ **independent of** $P(\theta_{Call|Alarm=NO})$

Parameter Independence, Graphical View

Global Parameter Independence



Local Parameter Independence

sample 1

sample 2

Provided all variables are observed in all cases, we can perform Bayesian update each parameter independently !!!

Which PDFs Satisfy Our Assumptions? (Geiger & Heckerman 97,99)

- **Discrete DAG Models:**

$$x_i \mid \pi_{x_i}^j \sim \text{Multi}(\theta)$$

Dirichlet prior:

$$P(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} = C(\alpha) \prod_k \theta_k^{\alpha_k - 1}$$

- **Gaussian DAG Models:**

$$x_i \mid \pi_{x_i}^j \sim \text{Normal}(\mu, \Sigma)$$

Normal prior:

$$p(\mu \mid \nu, \Psi) = \frac{1}{(2\pi)^{n/2} \mid \Psi \mid^{1/2}} \exp\left\{ -\frac{1}{2} (\mu - \nu)' \Psi^{-1} (\mu - \nu) \right\}$$

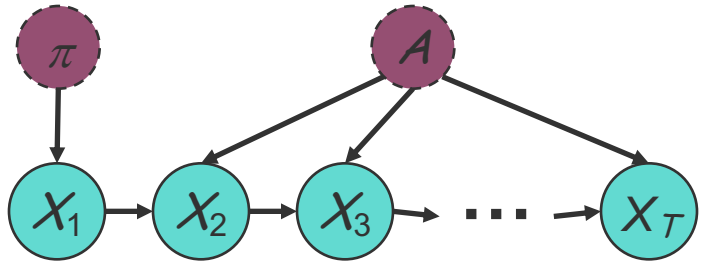
Normal-Wishart prior:

$$p(\mu \mid \nu, \alpha_\mu, \mathbf{W}) = \text{Normal}\left(\nu, (\alpha_\mu \mathbf{W})^{-1}\right),$$

$$p(\mathrm{W} \mid \alpha_w, \mathrm{T}) = c(n, \alpha_w) |\mathrm{T}|^{\alpha_w/2} |\mathrm{W}|^{(\alpha_w - n - 1)/2} \exp\left\{ \frac{1}{2} \mathrm{tr}\{\mathrm{TW}\} \right\},$$

where $\mathbf{W} = \Sigma^{-1}$.

Parameter sharing



- Consider a time-invariant (stationary) 1st-order Markov model
  - Initial state probability vector: $\pi_k \overset{\text{def}}{=} p(X_1^k = 1)$
  - State transition probability matrix: $A_{ij} \overset{\text{def}}{=} p(X_t^j = 1 \mid X_{t-1}^i = 1)$
- The joint: $p(X_{1:T} \mid \theta) = p(x_1 \mid \pi) \prod_{t=2}^{T} \prod_{t=2} p(X_t \mid X_{t-1})$
- The log-likelihood: $\ell(\theta; D) = \sum_n \log p(x_{n,1} \mid \pi) + \sum_n \sum_{t=2}^{T} \log p(x_{n,t} \mid x_{n,t-1}, A)$
- Again, we optimize each parameter separately
  - π is a multinomial frequency vector, and we've seen it before
  - What about A?

Learning a Markov chain transition matrix

- A is a stochastic matrix: $\sum_j A_{ij} = 1$
- Each row of A is multinomial distribution.
- So MLE of A_{ij} is the fraction of transitions from i to j

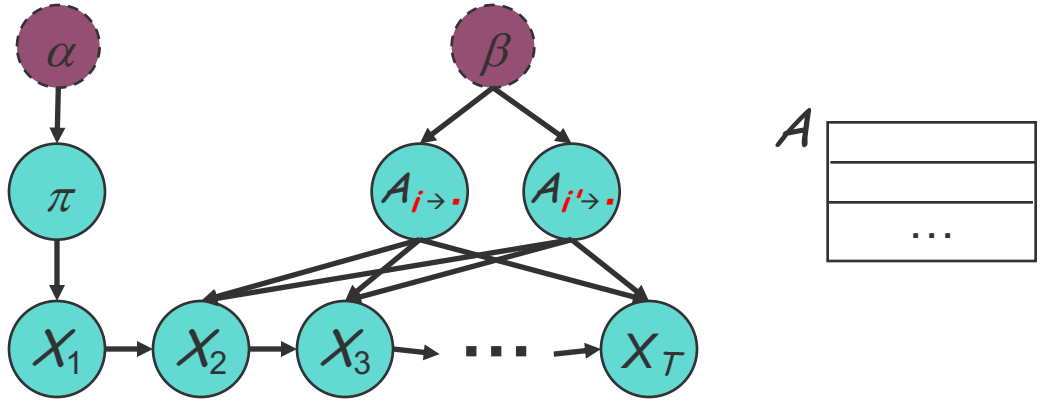
$$A_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^{T} x_{n,t-1}^{i} x_{n,t}^{j}}{\sum_n \sum_{t=2}^{T} x_{n,t-1}^{i}}$$

- Application:
  - if the states X_t represent words, this is called a *bigram language model*
- Sparse data problem:
  - If $i \to j$ did not occur in data, we will have $A_{ij} = 0$, then any future sequence with word pair $i \to j$ will have zero probability.
  - A standard hack: *backoff smoothing* or *deleted interpolation*

$$\tilde{A}_{i\to\bullet} = \lambda \eta_t + (1-\lambda) A_{i\to\bullet}^{ML}$$

Bayesian language model

- Global and local parameter independence



  - The posterior of $A_{i\to\cdot}$ and $A_{i'\to\cdot}$ is factorized despite v-structure on X_t, because X_{t-1} acts like a multiplexer
  - Assign a Dirichlet prior β_i to each row of the transition matrix:

$$A_{ij}^{Bayes} \stackrel{\text{def}}{=} p(j \mid i, D, \beta_i) = \frac{\#(i \to j) + \beta_{i,k}}{\#(i \to \bullet) + |\beta_i|} = \lambda_i \beta'_{i,k} + (1 - \lambda_i) A_{ij}^{ML}, \text{ where } \lambda_i = \frac{|\beta_i|}{|\beta_i| + \#(i \to \bullet)}$$

    - We could consider more realistic priors, e.g., mixtures of Dirichlets to account for types of words (adjectives, verbs, etc.)

IV: More on EM

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?

$$A_{ij} = \sum_{n,t} \left\langle y^i_{n,t-1} y^j_{n,t} \right\rangle \qquad B_{ik} = \sum_{n,t} \left\langle y^i_{n,t} \right\rangle x^k_{n,t}$$

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

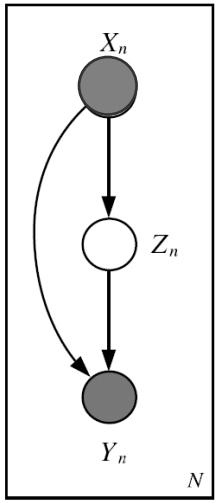
$$ESS_i \mathrel{+}= \left\langle SS_i(x_{n,i}, x_{n,\pi_i}) \right\rangle_{p(x_{n,H} | x_{n,-H})}$$

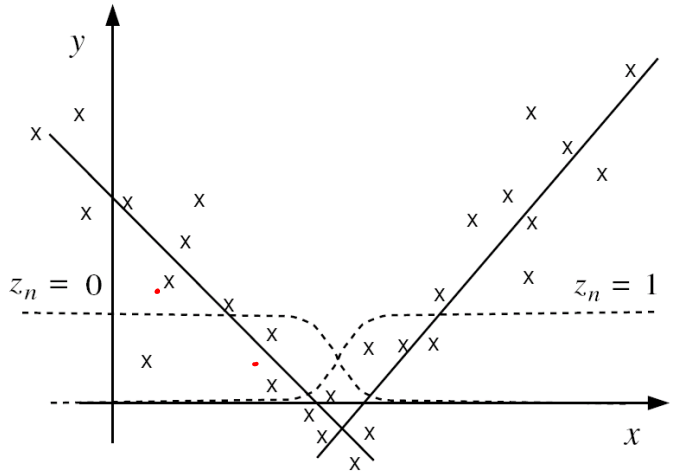
% M-step

for each node i

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

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:

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

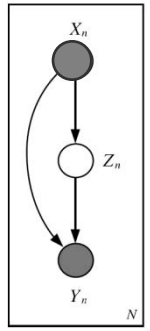
  - Each expert can be a linear regression model: $P(y|x, z^k = 1) = \mathcal{N}(y; \theta_k^T x, \sigma_k^2)$
  - The posterior expert responsibilities are

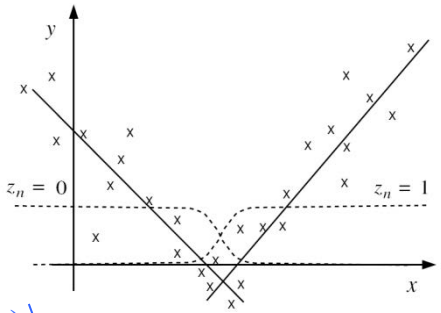
$$P(z^k = 1|x, y, \theta) = \frac{p(z^k = 1|x) p_k(y|x, \theta_k, \sigma_k^2)}{\sum_j p(z^j = 1|x) p_j(y|x, \theta_j, \sigma_j^2)}$$

EM for conditional mixture model

- Model:

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





- The objective function

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

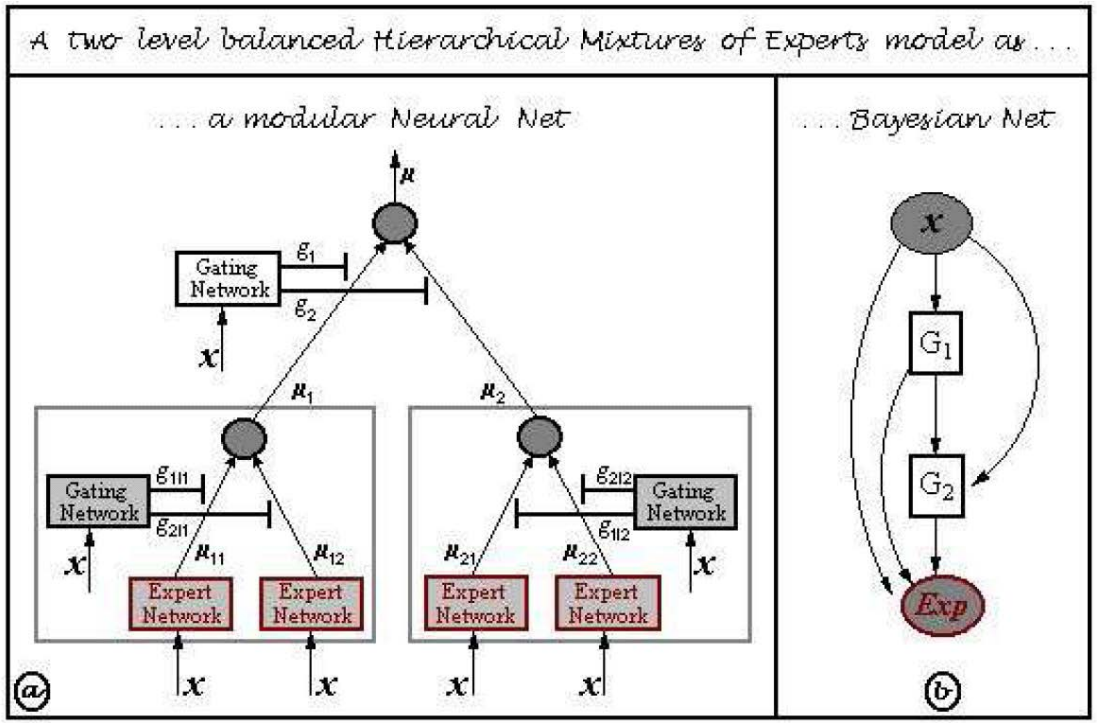
$$= \sum_n \sum_k \left\langle z_n^k \right\rangle \log\left(\text{softmax}(\xi_k^T x_n)\right) - \frac{1}{2} \sum_n \sum_k \left\langle z_n^k \right\rangle \left(\frac{(y_n - \theta_k^T x_n)}{\sigma_k^2} + \log \sigma_k^2 + C \right)$$

- EM:
  - E-step:
  - M-step:

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

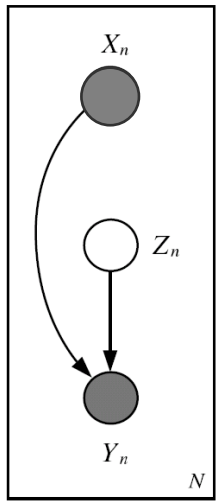
    - using the normal equation for standard LR $\theta = (X^T X)^{-1} X^T y$, but with the data re-weighted by τ (homework)
    - IRLS and/or weighted IRLS algorithm to update $\{\xi_k, \theta_k, \sigma_k\}$ based on data pair (x_n, y_n), with weights $\tau_n^{k(t)}$ (homework?)

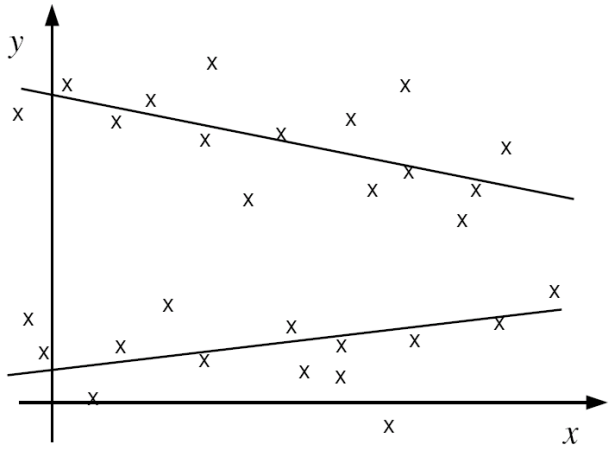
Hierarchical mixture of experts



- 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:

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

  - 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