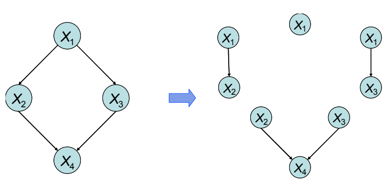
School of Computer Science
Carnegie Mellon

Probabilistic Graphical Models

Generalized linear models

Eric Xing

Lecture 6, February 3, 2014



Reading: KF-chap 17

© Eric Xing @ CMU, 2005-2014

Parameterizing graphical models

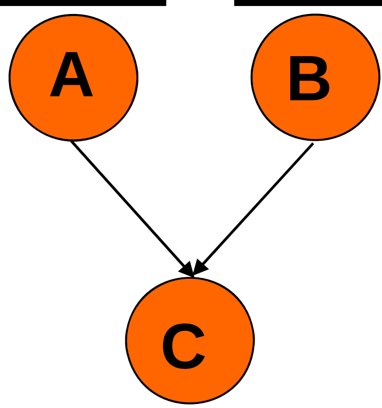
- Bayesian network:

$$P(\mathbf{X}) = \prod_{i=1:d} P(X_i \mid \mathbf{X}_{\pi_i})$$

Discrete

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

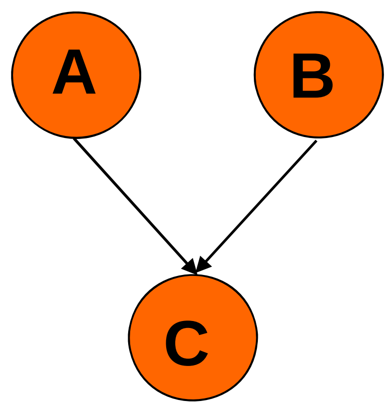


	a^0b^0	a^0b^1	a^1b^0	a^1b^1
c^0	0.45	1	0.9	0.7
c^1	0.55	0	0.1	0.3

Or

Continuous

$A \sim N(\mu_a, \Sigma_a)$ $B \sim N(\mu_b, \Sigma_b)$



$C \sim N(A+B, \Sigma_c)$

Or

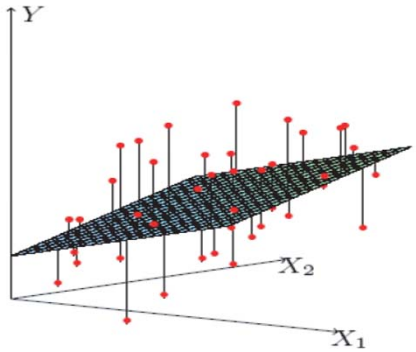
Hybrid

?

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

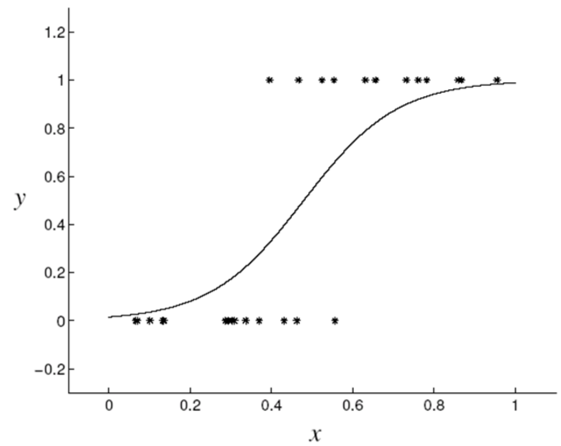
Recall: Logistic Regression (sigmoid classifier, perceptron, etc.)

- The condition distribution: a Bernoulli

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

where μ is a logistic function

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



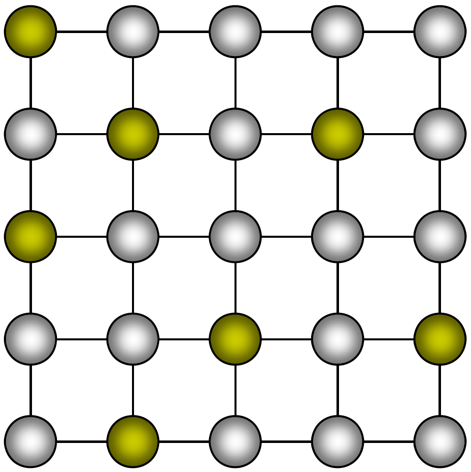
- We can used the brute-force gradient method as in LR

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

Parameterizing graphical models

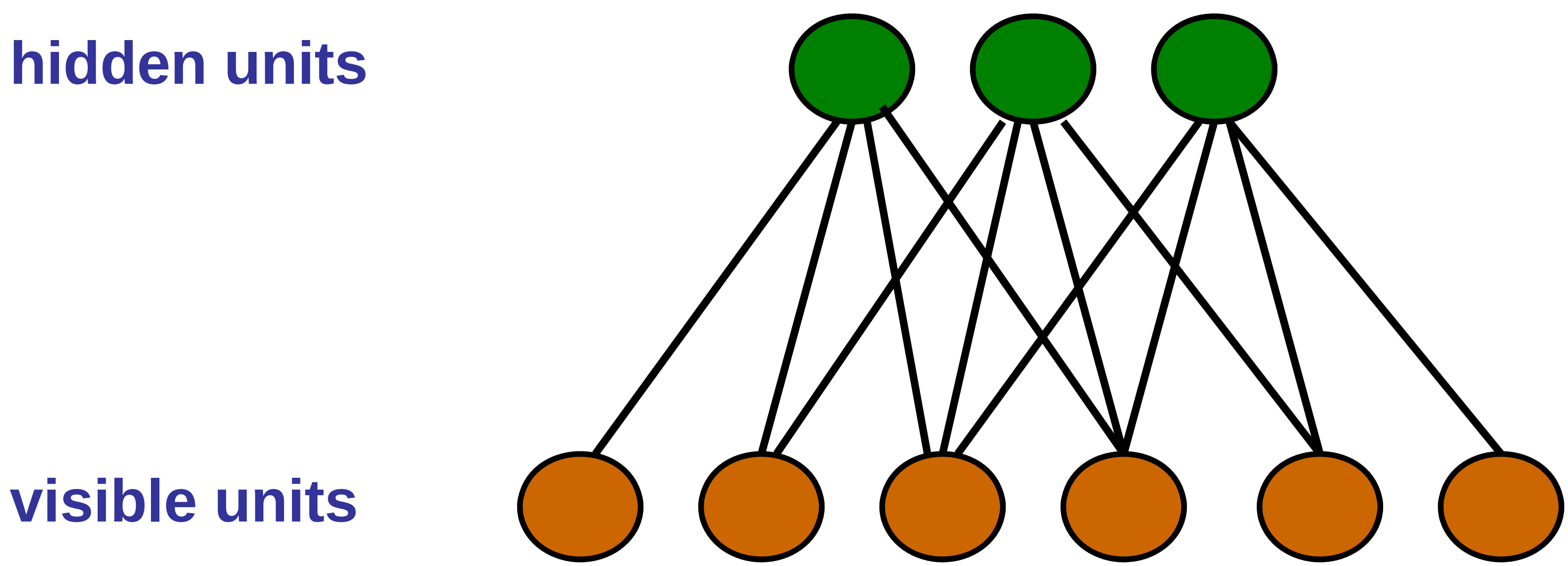
- Markov random fields

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



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

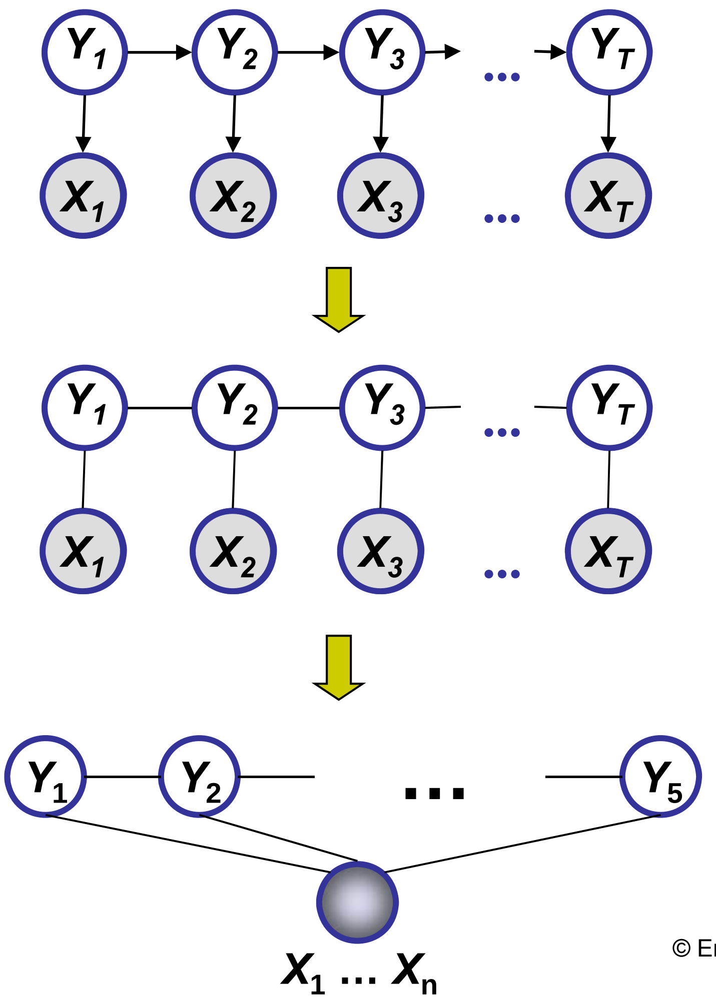
Restricted Boltzmann Machines



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

© Eric Xing @ CMU, 2005-2014

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

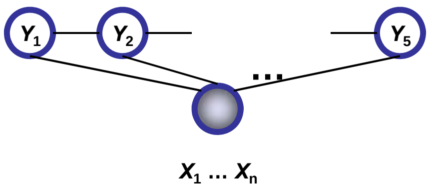
- When labeling X_i future observations are taken into account

© Eric Xing @ CMU, 2005-2014

Conditional Distribution

- If the graph $G = (V, E)$ of $\mathbf{Y}$ is a tree, the conditional distribution over the label sequence $\mathbf{Y} = \mathbf{y}$, given $\mathbf{X} = \mathbf{x}$, by the Hammersley Clifford theorem of random fields is:

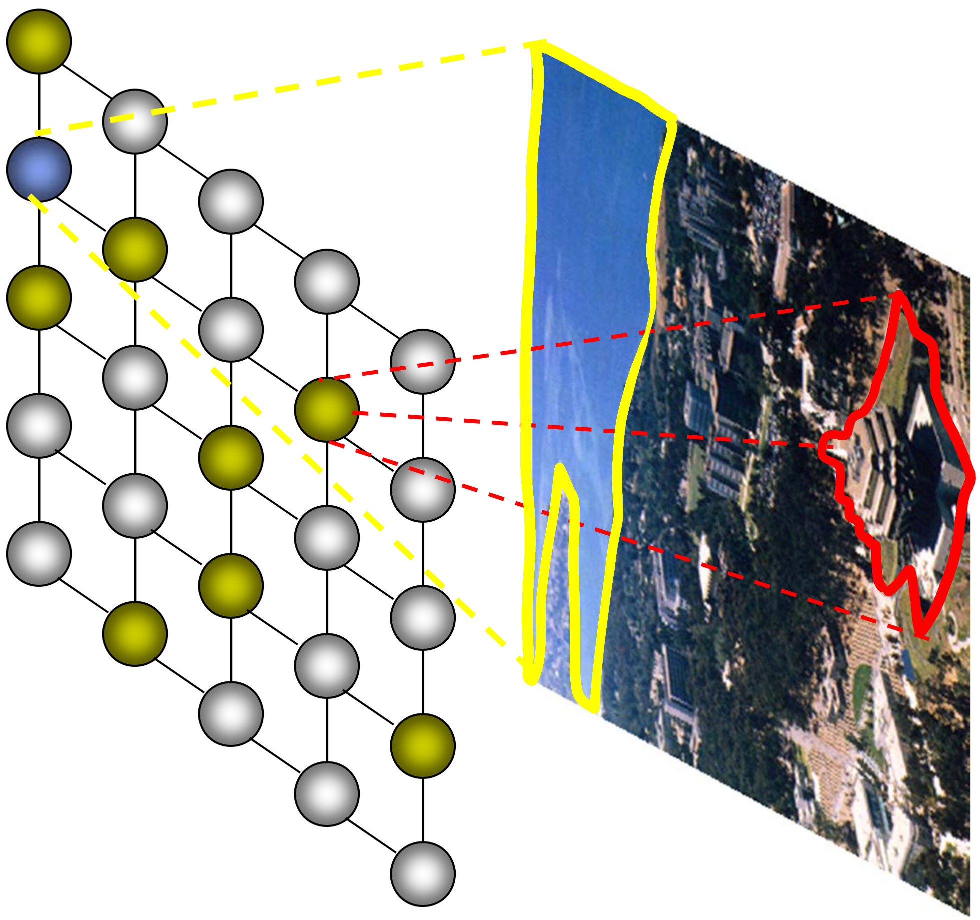
$$p_\theta(\mathrm{y} \mid \mathrm{x}) = \frac{1}{Z(\mathrm{x})} \exp\left(\sum_{e \in E, k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v \in V, k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$



  - x is a data sequence
  - y is a label sequence
  - *v* is a vertex from vertex set V = set of label random variables
  - *e* is an edge from edge set E over V
  - f_k and g_k are given and fixed. g_k is a Boolean vertex feature; f_k is a Boolean edge feature
  - *k* is the index number of the features
  - $\theta = (\lambda_1, \lambda_2, \cdots, \lambda_n; \mu_1, \mu_2, \cdots, \mu_n); \lambda_k$ and μ_k are parameters to be estimated
  - $\mathrm{y}|_e$ is the set of components of y defined by edge *e*
  - $\mathrm{y}|_v$ is the set of components of y defined by vertex *v*

© Eric Xing @ CMU, 2005-2014

2-D Conditional Random Fields



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

- Allow arbitrary dependencies on input
- Clique dependencies on labels
- Use approximate inference for general graphs

© Eric Xing @ CMU, 2005-2014

Exponential family, a basic building block

- For a numeric random variable *X*

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

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

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

Example: Multivariate Gaussian Distribution

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

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

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

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

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

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

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

- 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 $\mathbf{x} \sim \mathrm{multi}(\mathbf{x} \mid \pi)$,

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

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

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

- 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

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

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

© Eric Xing @ CMU, 2005-2014

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.

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

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

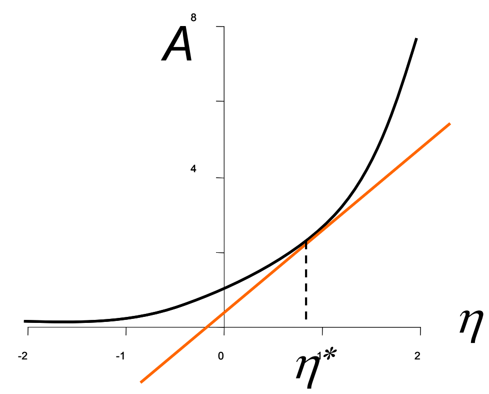
$$\dots$$

- 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

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



- $A(\eta)$ is convex since

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

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

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

  - 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

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

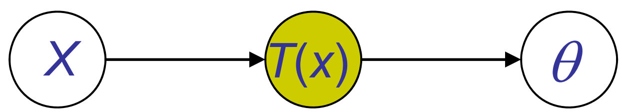
- Take derivatives and set to zero:

$$\frac{\partial \ell}{\partial \eta} = \sum_n T(x_n) - N\frac{\partial A(\eta)}{\partial \eta} = 0$$
$$\Rightarrow \quad \frac{\partial A(\eta)}{\partial \eta} = \frac{1}{N}\sum_n T(x_n)$$
$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_n T(x_n)$$

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

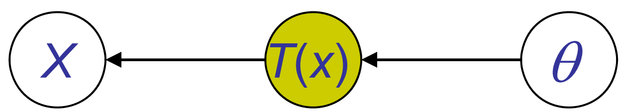
Sufficiency

- 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



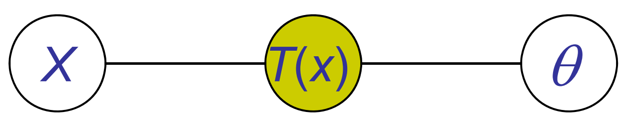
$$p(\theta \mid T(x), x) = p(\theta \mid T(x))$$

  - Frequentist view



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

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



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

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

© Eric Xing @ CMU, 2005-2014

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:

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

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

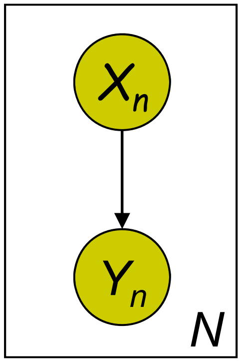
© Eric Xing @ CMU, 2005-2014

Bayesian est.

Generalized Linear Models (GLIMs)

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

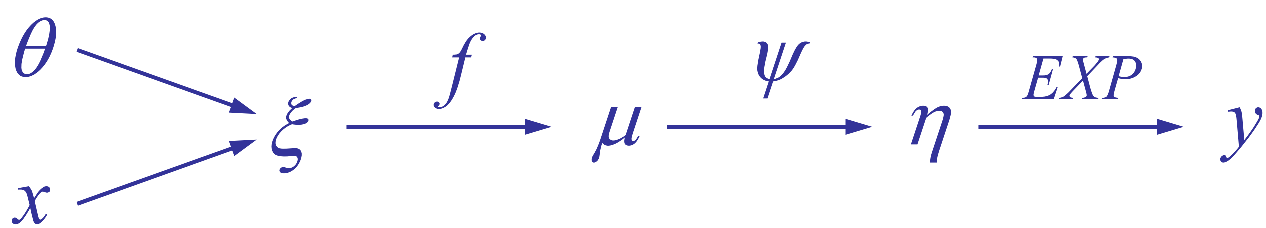
    model $\mathrm{E}_p(Y)=\mu=f(\theta^T X)$
    - What is $p()$? the cond. dist. of Y.
    - What is $f()$? the response function.



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

© Eric Xing @ CMU, 2005-2014

GLIM, cont.



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

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

- 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: $f = \psi^{-1}(\cdot)$
    - In this case $\theta^T x$ directly corresponds to canonical parameter η.

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	$\mu = -\eta^{-1}$

© Eric Xing @ CMU, 2005-2014

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 = least mean squares (LMS) algorithm:

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

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

© Eric Xing @ CMU, 2005-2014

Batch learning for canonical GLIMs

- The Hessian matrix

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1 & -- \\ -- & \mathbf{x}_2 & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n & -- \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

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

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

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

$$= -X^T W X$$

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

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

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

© Eric Xing @ CMU, 2005-2014

Recall LMS

- Cost function in matrix form:

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

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1 & -- \\ -- & \mathbf{x}_2 & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n & -- \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- To minimize $J(\theta)$, take derivative and set to zero:

$$\nabla_\theta J = \frac{1}{2}\nabla_\theta \mathrm{tr}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T \vec{y}\right)$$
$$= \frac{1}{2}\left(\nabla_\theta \mathrm{tr}\theta^T X^T X\theta - 2\nabla_\theta \mathrm{tr}\vec{y}^T X\theta + \nabla_\theta \mathrm{tr}\vec{y}^T \vec{y}\right)$$
$$= \frac{1}{2}\left(X^T X\theta + X^T X\theta - 2X^T \vec{y}\right)$$
$$= X^T X\theta - X^T \vec{y} = 0$$

$$\Rightarrow \quad X^T X\theta = X^T \vec{y}$$

The normal equations

$$\Downarrow$$

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

© Eric Xing @ CMU, 2005-2014

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

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

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

- Now:

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

-

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

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

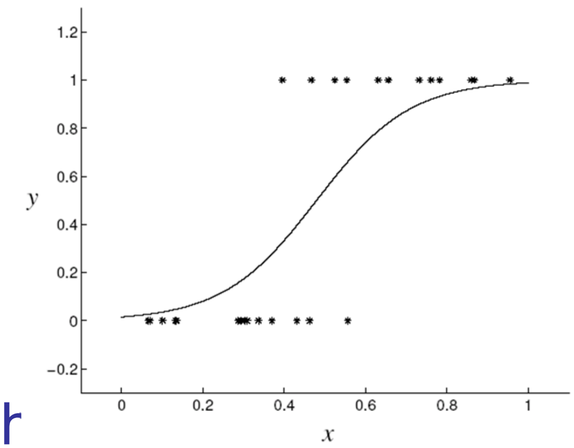
Example 1: logistic regression (sigmoid classifier)

- The condition distribution: a Bernoulli

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

where μ is a logistic function

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



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

- IRLS

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

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

© Eric Xing @ CMU, 2005-2014

Logistic regression: practical issues

- It is very common to use *regularized* maximum likelihood.

$$p(y=\pm 1 | x,\theta) = \frac{1}{1+e^{-y\theta^T x}} = \sigma(y\theta^T x)$$

$$p(\theta) \sim \text{Normal}(0, \lambda^{-1} I)$$

$$l(\theta) = \sum_n \log\left(\sigma(y_n \theta^T x_n)\right) - \frac{\lambda}{2}\theta^T\theta$$

  - IRLS takes $O(Nd^3)$ per iteration, where N = number of training cases and d = dimension of input x.
  - Quasi-Newton methods, that approximate the Hessian, work faster.
  - Conjugate gradient takes $O(Nd)$ per iteration, and usually works best in practice.
  - Stochastic gradient descent can also be used if N is large c.f. perceptron rule:

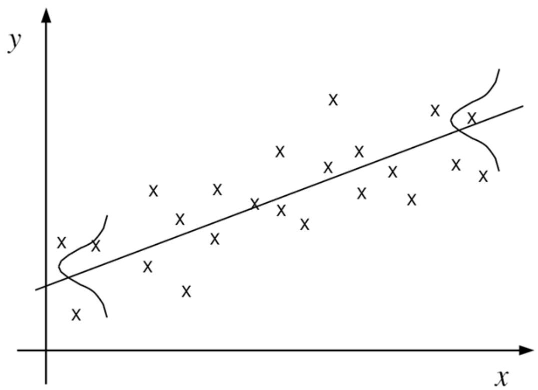
$$\nabla_\theta \ell = \left(1 - \sigma(y_n \theta^T x_n)\right) y_n x_n - \lambda\theta$$

© Eric Xing @ CMU, 2005-2014

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

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 $\frac{d\mu}{d\eta} = 1$, $W = I$

$$\Rightarrow \quad \theta^{t+1} = \left(X^T W^t X\right)^{-1} X^T W^t z^t = \left(X^T X\right)^{-1} X^T\left(X\theta^t + (y-\mu^t)\right) = \theta^t + \left(X^T X\right)^{-1} X^T(y-\mu^t) \quad \overset{t\to\infty}{\Rightarrow} \quad \theta = (X^T X)^{-1} X^T Y$$

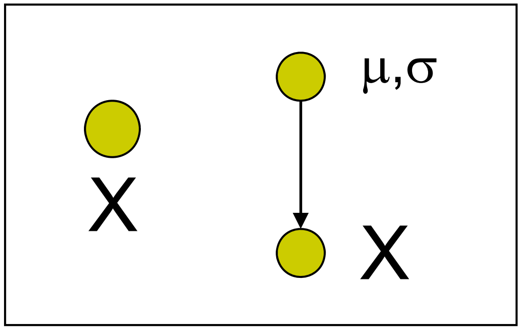
Steepest descent

Normal equation

Simple GMs are the building blocks of complex BNs

Density estimation

Parametric and nonparametric methods

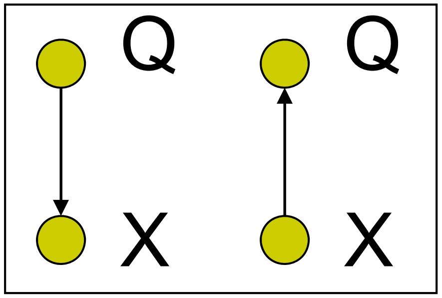


Regression

Linear, conditional mixture, nonparametric

Classification

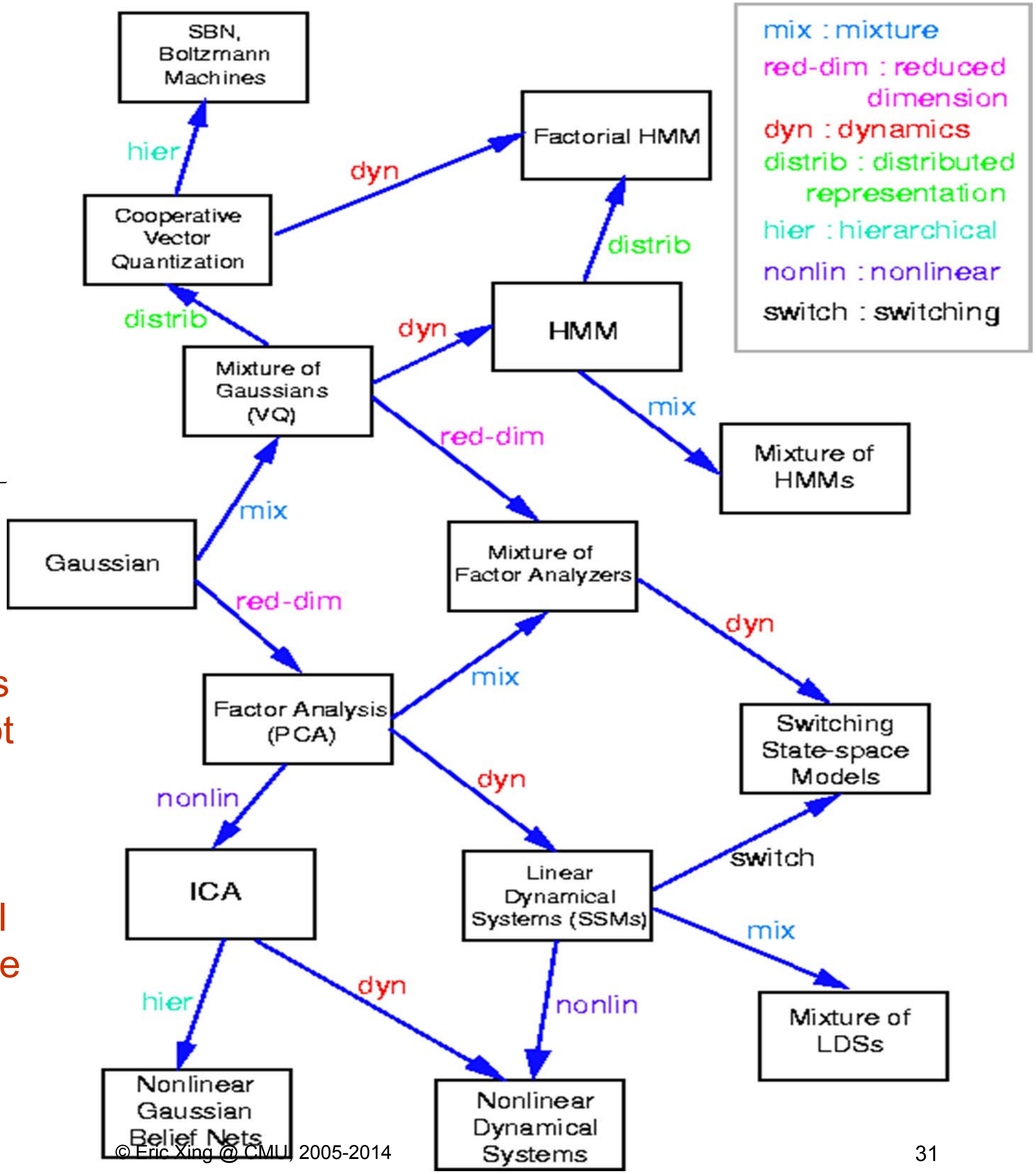
Generative and discriminative approach



© Eric Xing @ CMU, 2005-2014

School of Computer Science
Carnegie Mellon

An (incomplete) genealogy of graphical models



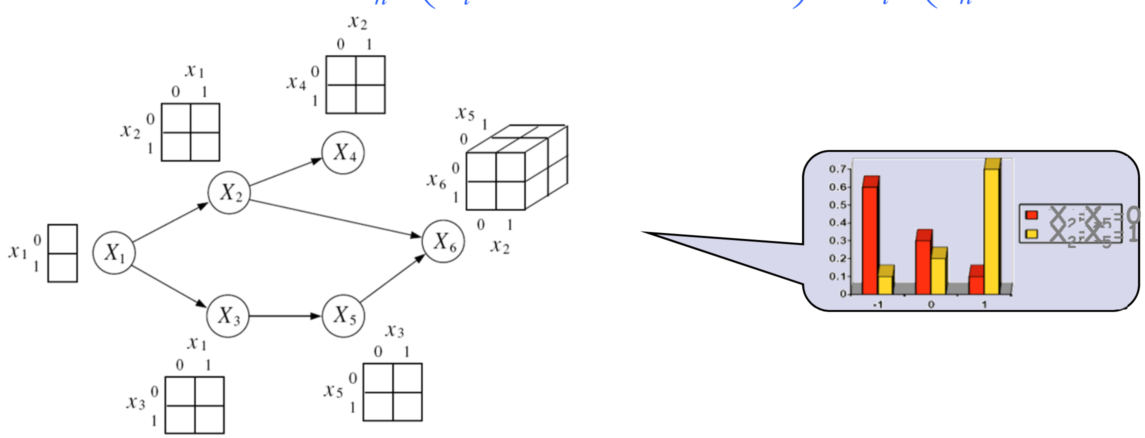
The structures of most GMs (e.g., all listed here), are not learned from data, but designed by human.

But such designs are useful and indeed favored because thereby human knowledge are put into good use …

MLE for general BNs

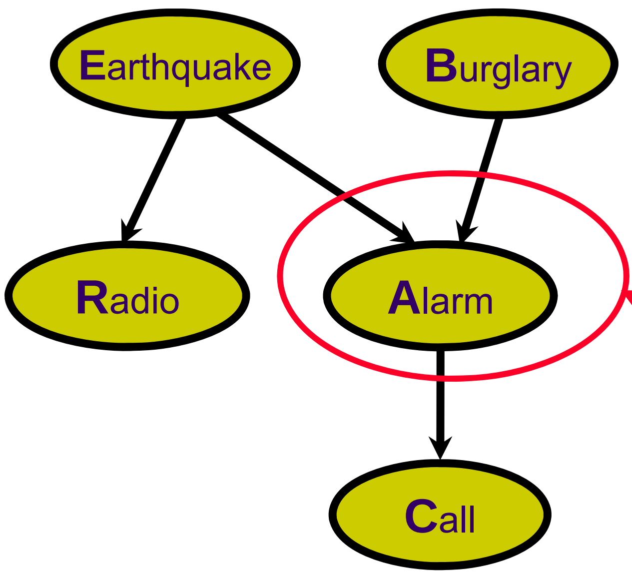
- If we assume the parameters for each CPD (a GLIM) are globally independent, and all nodes are fully observed, then the log-likelihood function decomposes into a sum of local terms, one per node:

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



- Therefore, MLE-based parameter estimation of GM reduces to local est. of each GLIM

How to define parameter prior?



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:

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

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

Assumptions (Geiger & Heckerman 97,99):

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

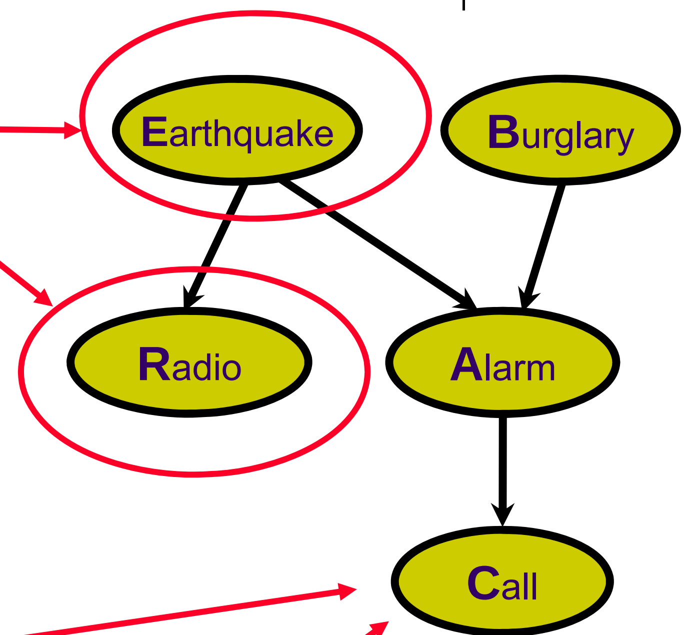
© Eric Xing @ CMU, 2005-2014

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:

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

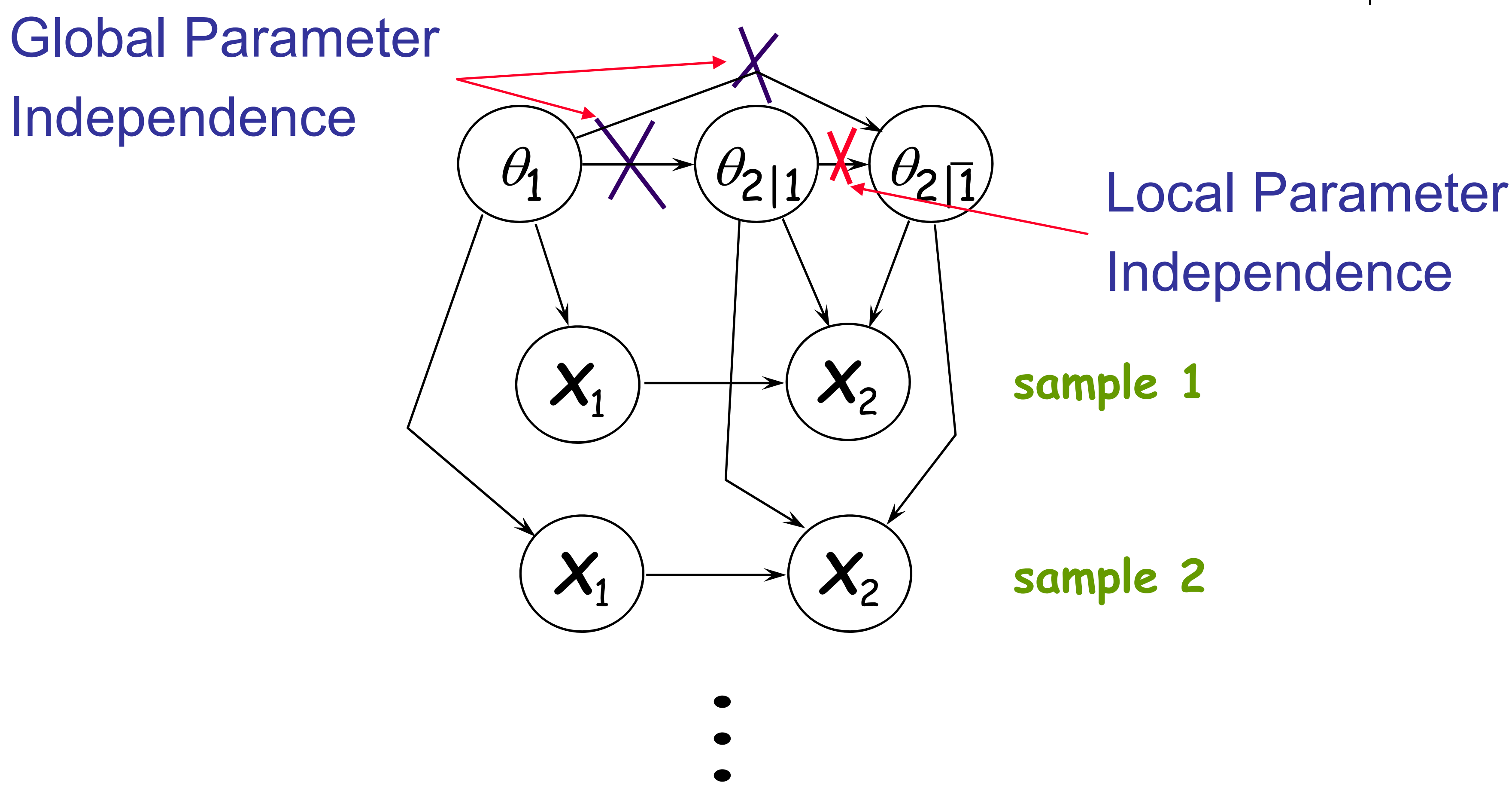
$$P(\theta_{Call|Alarm=YES})$$

independent of

$$P(\theta_{Call|Alarm=NO})$$

© Eric Xing @ CMU, 2005-2014

Parameter Independence, Graphical View



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:** $x_i \mid \pi_{x_i}^{j} \sim \mathrm{Multi}(\theta)$

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

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:

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

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

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

© Eric Xing @ CMU, 2005-2014

Summary: Parameterizing GM

- For exponential family dist., MLE amounts to moment matching
- GLIM:
  - Natural response
  - Iteratively Reweighted Least Squares as a general algorithm
- GLIMs are building blocks of most GMs in practical use
- Parameter independence and appropriate priors

© Eric Xing @ CMU, 2005-2014