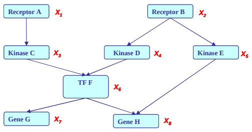
School of Computer Science
Carnegie Mellon

Learning generalized linear models and tabular CPT of structured full BN

Probabilistic Graphical Models (10-708)

Lecture 9, Oct 15, 2007



Eric Xing

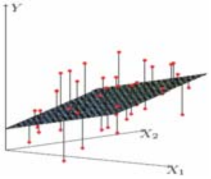
Reading: J-Chap. 7,8.

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 n



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

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

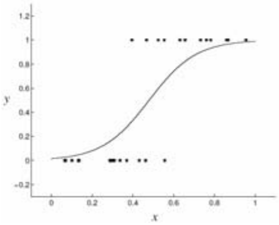
Logistic Regression (sigmoid classifier)

- The condition distribution: a Bernoulli

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

where μ is a logistic function

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



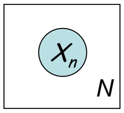
- We can used the brute-force gradient method as in LR

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

Exponential family

- For a numeric random variable X

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



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,...

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

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

- Exponential family representation

Natural parameter

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

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

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

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

  - 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)

Multinomial distribution

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

$$p(x|\pi) = \pi_1^{x^1}\pi_2^{x^2}\cdots\pi_K^{x^K} = \exp\left\{\sum_k x^k \ln \pi_k\right\}$$

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

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

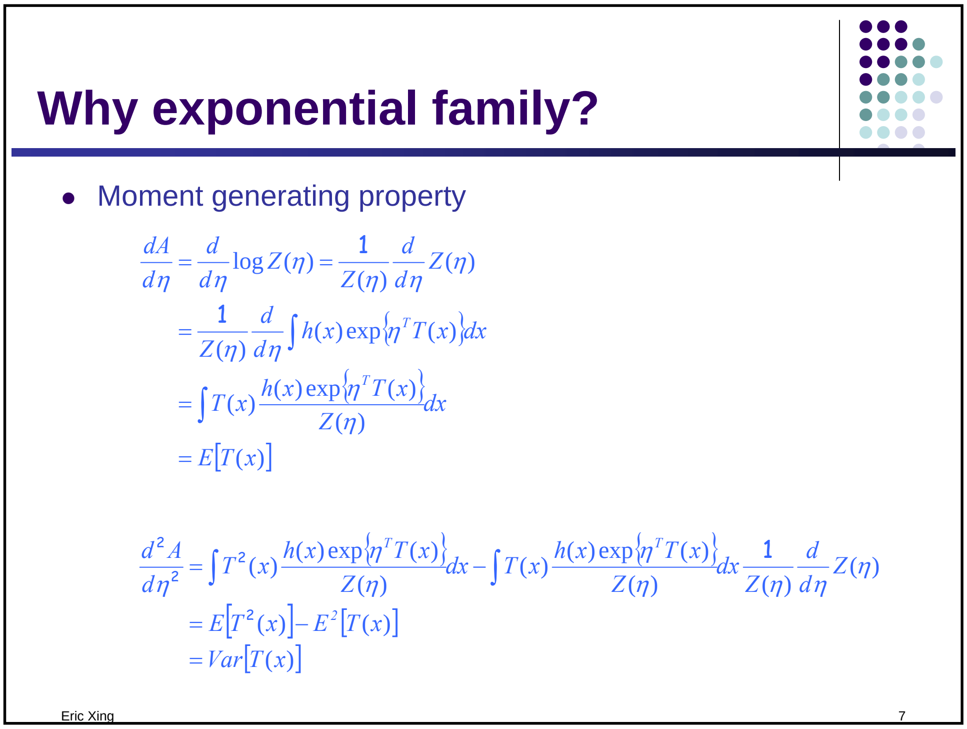
- Exponential family representation

$$\eta = \left[\ln\left({}^{\pi_k}\!/_{\pi_K}\right); 0\right]$$

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

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

$$h(x) = 1$$

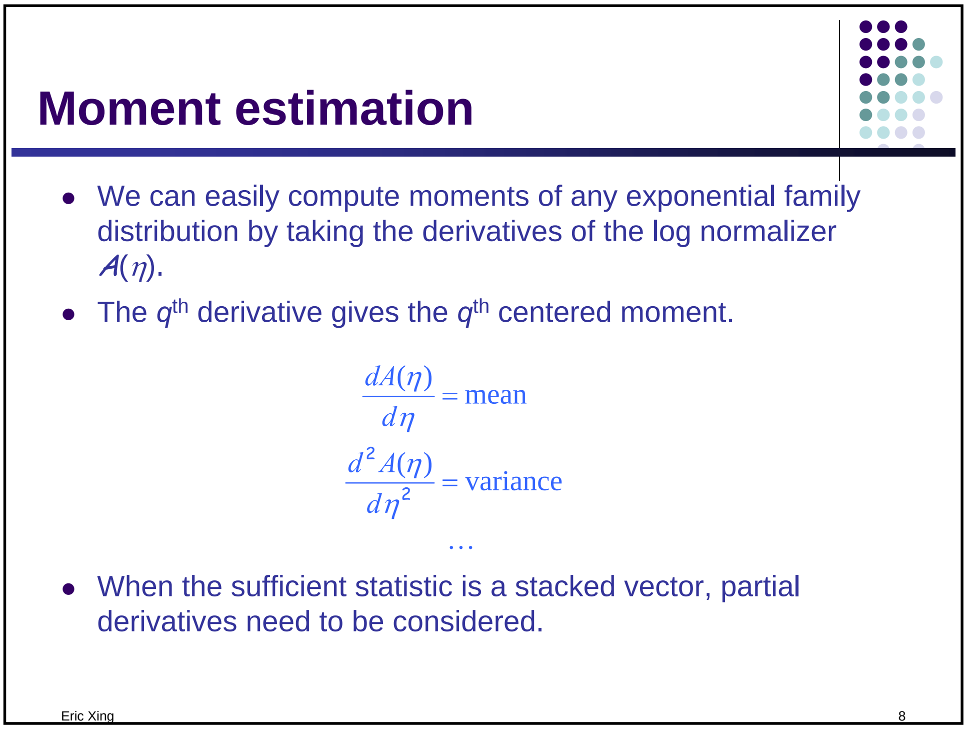


Why exponential family?

- Moment generating property

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

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



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.

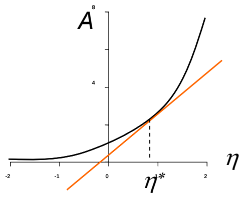
$$\frac{dA(\eta)}{d\eta} = \text{mean}$$
$$\frac{d^2 A(\eta)}{d\eta^2} = \text{variance}$$
$$\dots$$

- 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

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



- $A(h)$ 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

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

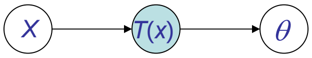
- Take derivatives and set to zero:

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

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

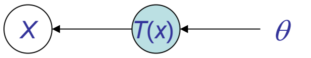
Sufficiency

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



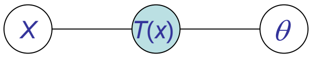
$$p(\theta \mid T(x), x) = p(\theta \mid T(x))$$

  - 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:

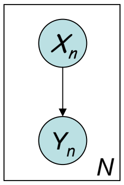
$$\eta = \log\lambda$$
$$T(x) = x$$
$$A(\eta) = \lambda = e^{\eta}$$
$$h(x) = \frac{1}{x!}$$

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

Generalized Linear Models (GLIMs)

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

    model $\mathrm{E}_p(Y)=\mu=f(\theta^{\mathrm{T}}X)$
    - 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.

GLIM, cont.

$$\begin{tikzcd}
\theta \arrow[dr] & & & \\
 & \xi \arrow[r, "f"] & \mu \arrow[r, "\psi"] & \eta \\
x \arrow[ur] & & &
\end{tikzcd}$$

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

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

- 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 $\theta^{\mathrm{T}}x$ directly corresponds to canonical parameter η.

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

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

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

Batch learning for canonical GLIMs

- The Hessian matrix

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

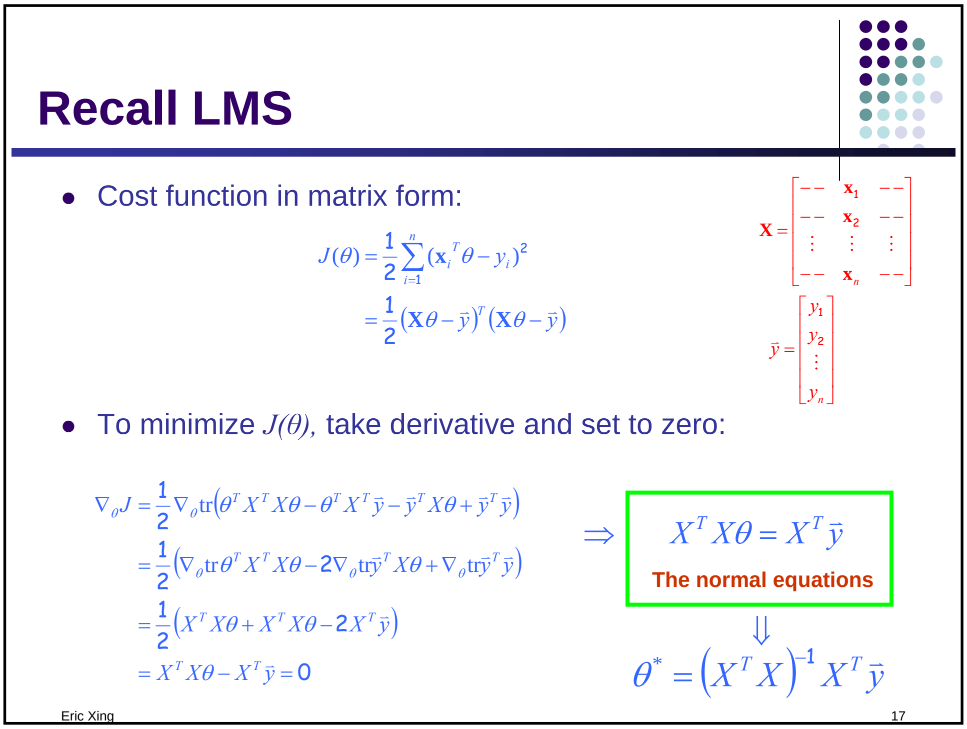
$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\begin{aligned}H = \frac{d^2\ell}{d\theta d\theta^T} &= \frac{d}{d\theta^T}\sum_n (y_n - \mu_n)x_n = \sum_n x_n \frac{d\mu_n}{d\theta^T}\\ &= -\sum_n x_n \frac{d\mu_n}{d\eta_n}\frac{d\eta_n}{d\theta^T}\\ &= -\sum_n x_n \frac{d\mu_n}{d\eta_n} x_n^T \quad \text{since } \eta_n = \theta^T x_n\\ &= -X^T W X\end{aligned}$$

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

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

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



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(θ)*, take derivative and set to zero:

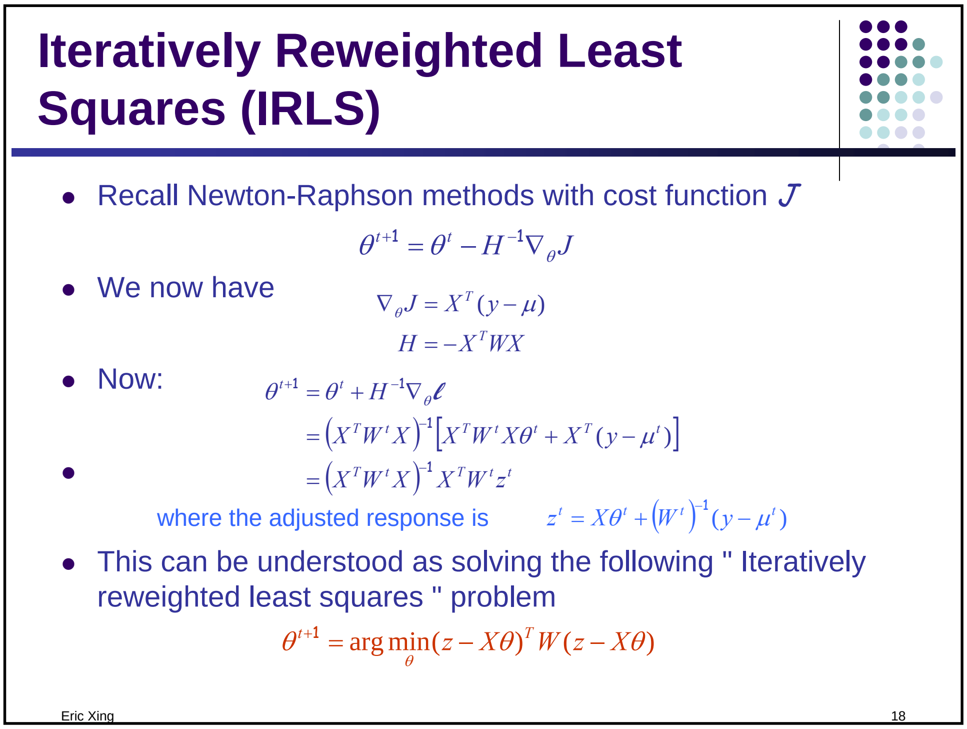
$$\nabla_\theta J = \frac{1}{2}\nabla_\theta \mathrm{tr}\left(\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T\vec{y}\right)$$
$$= \frac{1}{2}\left(\nabla_\theta \mathrm{tr}\theta^T X^T X\theta - 2\nabla_\theta \mathrm{tr}\vec{y}^T X\theta + \nabla_\theta \mathrm{tr}\vec{y}^T\vec{y}\right)$$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

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

The normal equations

$$\Downarrow$$

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



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

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

- Now:

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

-

$$= \left(X^T W^t X\right)^{-1} X^T W^t z^t$$

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

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

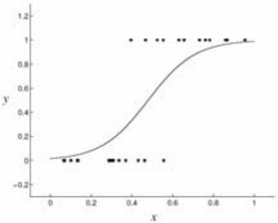
Example 1: logistic regression (sigmoid classifier)

- 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: $E[y \mid x] = \mu = \dfrac{1}{1+e^{-\eta(x)}}$
  - and canonical response function $\eta = \xi = \theta^T x$
- IRLS

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

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

Logistic regression: practical issues

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

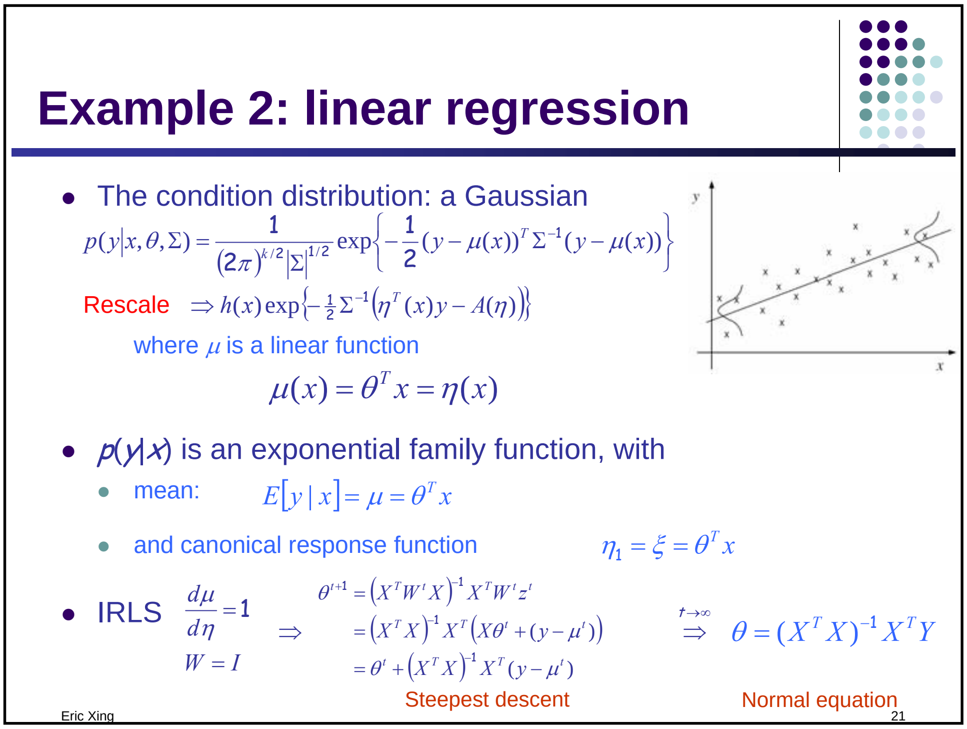
$$p(y=\pm 1 | x, \theta) = \frac{1}{1+e^{-y\theta^T x}} = \sigma(y\theta^T x)$$

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

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

  - 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:

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



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

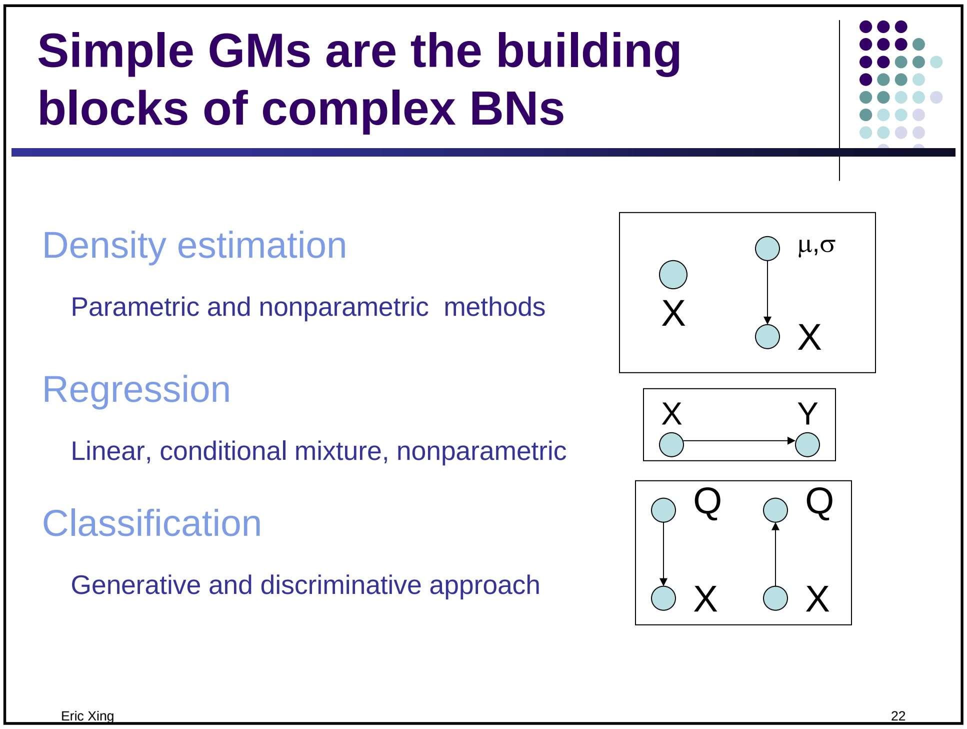
$$\mu(x)=\theta^T x=\eta(x)$$

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

$$\Rightarrow \begin{aligned}\theta^{t+1}&=\left(X^TW^tX\right)^{-1}X^TW^tz^t\\&=\left(X^TX\right)^{-1}X^T\left(X\theta^t+(y-\mu^t)\right)\\&=\theta^t+\left(X^TX\right)^{-1}X^T(y-\mu^t)\end{aligned} \overset{t\to\infty}{\Rightarrow} \theta=(X^TX)^{-1}X^TY$$

Steepest descent

Normal equation



Simple GMs are the building blocks of complex BNs

Density estimation

Parametric and nonparametric methods

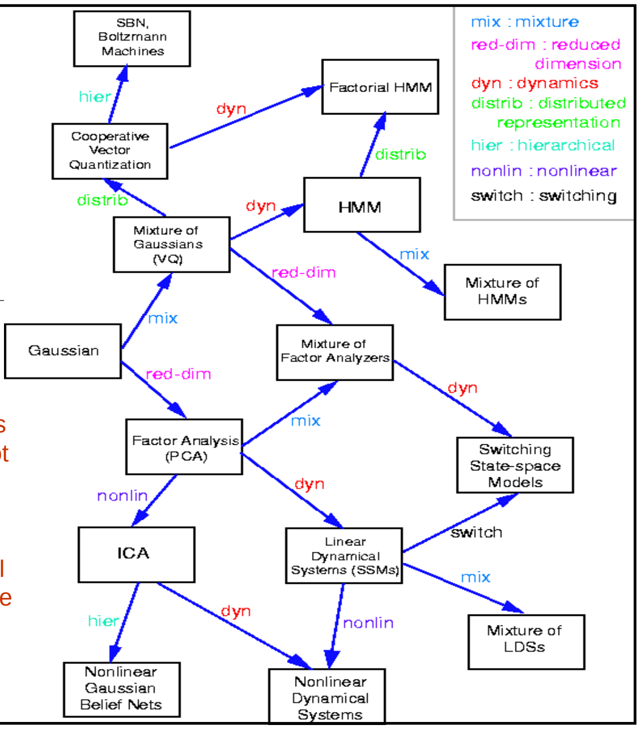
Regression

Linear, conditional mixture, nonparametric

Classification

Generative and discriminative approach

An (incomplete) genealogy of graphical models


SBN, Boltzmann Machines
Factorial HMM
Cooperative Vector Quantization
HMM
Mixture of Gaussians (VQ)
Mixture of HMMs
Gaussian
Mixture of Factor Analyzers
Factor Analysis (PCA)
Switching State-space Models
ICA
Linear Dynamical Systems (SSMs)
Mixture of LDSs
Nonlinear Gaussian Belief Nets
Nonlinear Dynamical Systems
hier
dyn
distrib
dyn
mix
red-dim
mix
red-dim
mix
dyn
nonlin
dyn
switch
mix
hier
dyn
nonlin
mix : mixture
red-dim : reduced dimension
dyn : dynamics
distrib : distributed representation
hier : hierarchical
nonlin : nonlinear
switch : switching

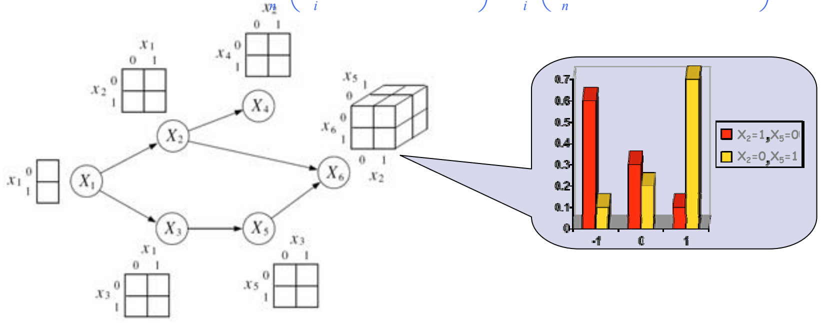
The structures of most GMs (e.g., all listed here), are not learned from data, but designed by human.

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

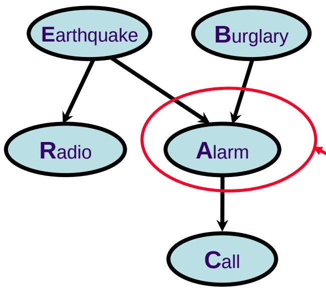
MLE for general BNs

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

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



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:

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

Assumptions (Geiger & Heckerman 97,99):

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

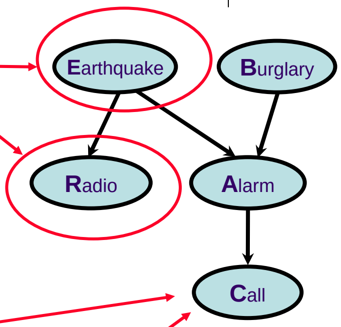
$p(\theta \mid G)$?

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:

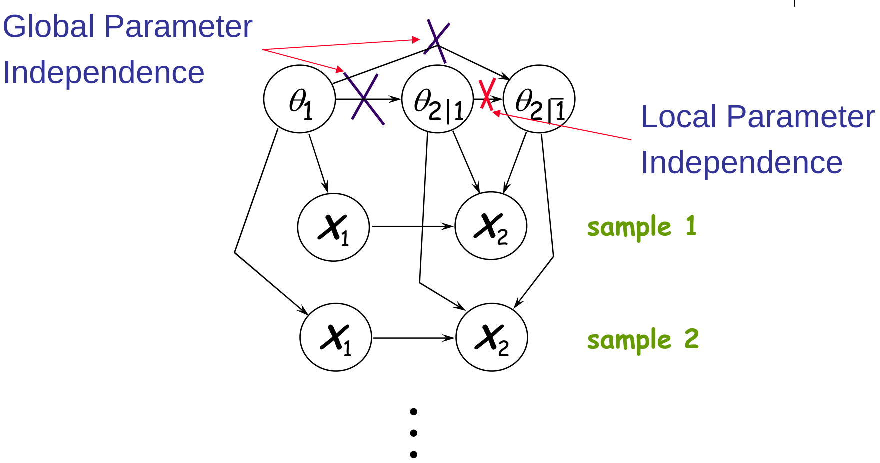
  $$p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j} \mid G)$$

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

independent of

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

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:** $x_i \mid \pi^j_{x_i} \sim \mathrm{Multi}(\theta)$

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

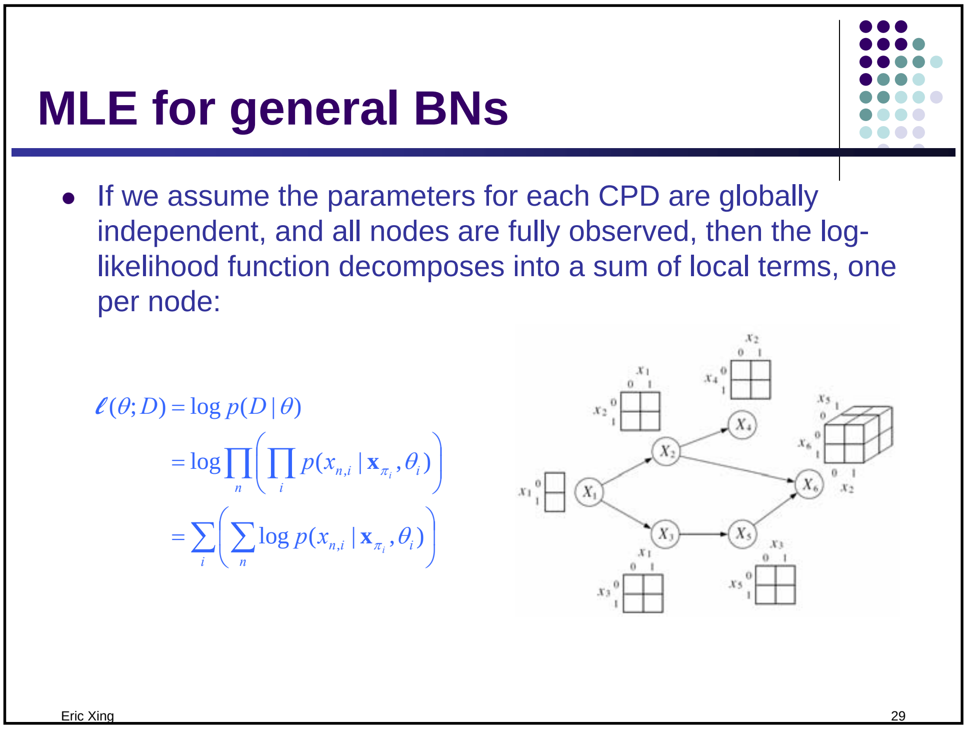
  Normal prior: $p(\mu \mid \nu, \Psi) = \frac{1}{(2\pi)^{n/2} |\Psi|^{1/2}} \exp\left\{ -\frac{1}{2} (\mu - \nu)' \Psi^{-1} (\mu - \nu) \right\}$

  Normal-Wishart prior:

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

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

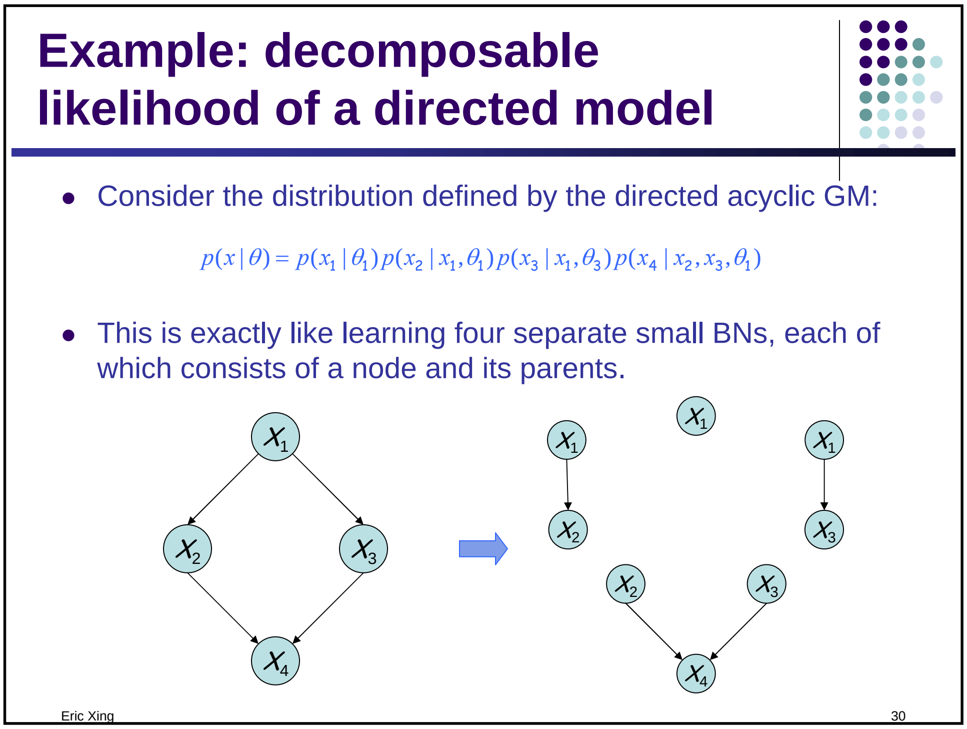
where $\mathbf{W} = \Sigma^{-1}$.



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:

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

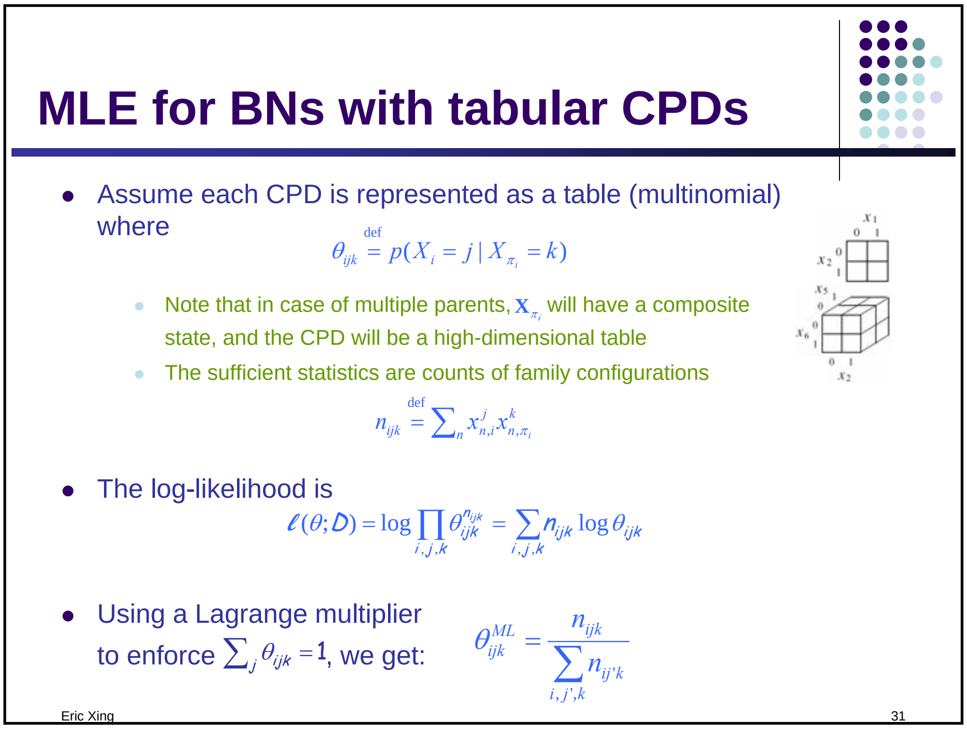


Example: decomposable likelihood of a directed model

- Consider the distribution defined by the directed acyclic GM:

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

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



x_1
0 1
x_2
0
1
x_5
1
0
x_6
0
1
0 1
x_2

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

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

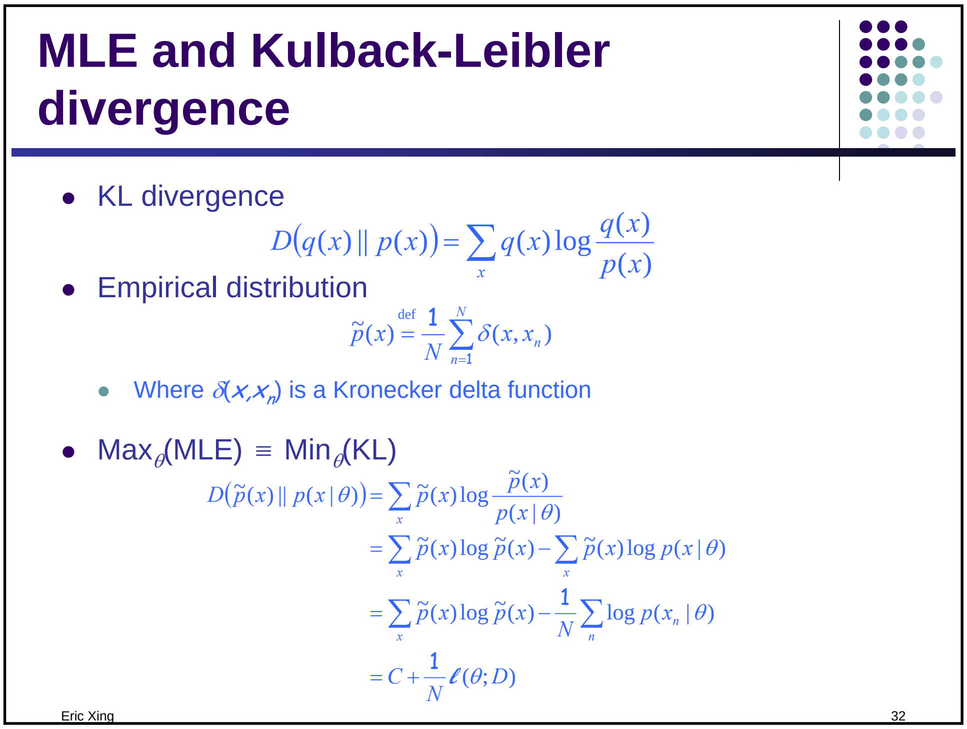
- The log-likelihood is

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

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

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

Eric Xing 31



MLE and Kulback-Leibler divergence

- KL divergence

$$D\big(q(x) \,\|\, p(x)\big) = \sum_x q(x) \log \frac{q(x)}{p(x)}$$

- Empirical distribution

$$\tilde{p}(x) \overset{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \delta(x, x_n)$$

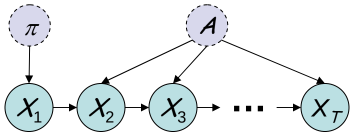
  - Where $\delta(x,x_n)$ is a Kronecker delta function

- Max$_\theta$(MLE) ≡ Min$_\theta$(KL)

$$\begin{aligned} D\big(\tilde{p}(x) \,\|\, p(x \mid \theta)\big) &= \sum_x \tilde{p}(x) \log \frac{\tilde{p}(x)}{p(x \mid \theta)} \\ &= \sum_x \tilde{p}(x) \log \tilde{p}(x) - \sum_x \tilde{p}(x) \log p(x \mid \theta) \\ &= \sum_x \tilde{p}(x) \log \tilde{p}(x) - \frac{1}{N} \sum_n \log p(x_n \mid \theta) \\ &= C + \frac{1}{N} \ell(\theta; D) \end{aligned}$$

Eric Xing 32

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*

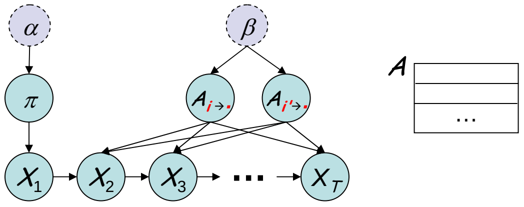
$$A_{ij}^{ML} = \frac{\#(i \to j)}{\#(i \to \bullet)} = \frac{\sum_n \sum_{t=2}^{T} x_{n,t-1}^i x_{n,t}^j}{\sum_n \sum_{t=2}^{T} x_{n,t-1}^i}$$

- Application:
  - if the states X_t represent words, this is called a *bigram language model*
- Sparse data problem:
  - If *i* → *j* did not occur in data, we will have A_{ij} =0, then any futher sequence with word pair *i* → *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:

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

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

Example: HMM: two scenarios

- **Supervised learning**: estimation when the "right answer" is known
  - **Examples:**

    GIVEN: a genomic region x = $x_1 \ldots x_{1,000,000}$ where we have good (experimental) annotations of the CpG islands

    GIVEN: the casino player allows us to observe him one evening, as he changes dice and produces 10,000 rolls

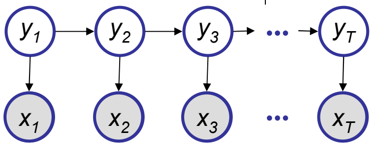
- **Unsupervised learning**: estimation when the "right answer" is unknown
  - **Examples:**

    GIVEN: the porcupine genome; we don't know how frequent are the CpG islands there, neither do we know their composition

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

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

Recall definition of HMM



- Transition probabilities between any two states

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

or $p(y_t \mid y_{t-1}^i = 1) \sim \text{Multinomial}\left(a_{i,1}, a_{i,2}, \ldots, a_{i,M}\right), \forall i \in \mathbb{I}.$

- Start probabilities

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

- Emission probabilities associated with each state

$$p(x_t \mid y_t^i = 1) \sim \text{Multinomial}\left(b_{i,1}, b_{i,2}, \ldots, b_{i,K}\right), \forall i \in \mathbb{I}.$$

or in general: $p(x_t \mid y_t^i = 1) \sim \mathrm{f}\left(\cdot \mid \theta_i\right), \forall i \in \mathbb{I}.$

Supervised ML estimation

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

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

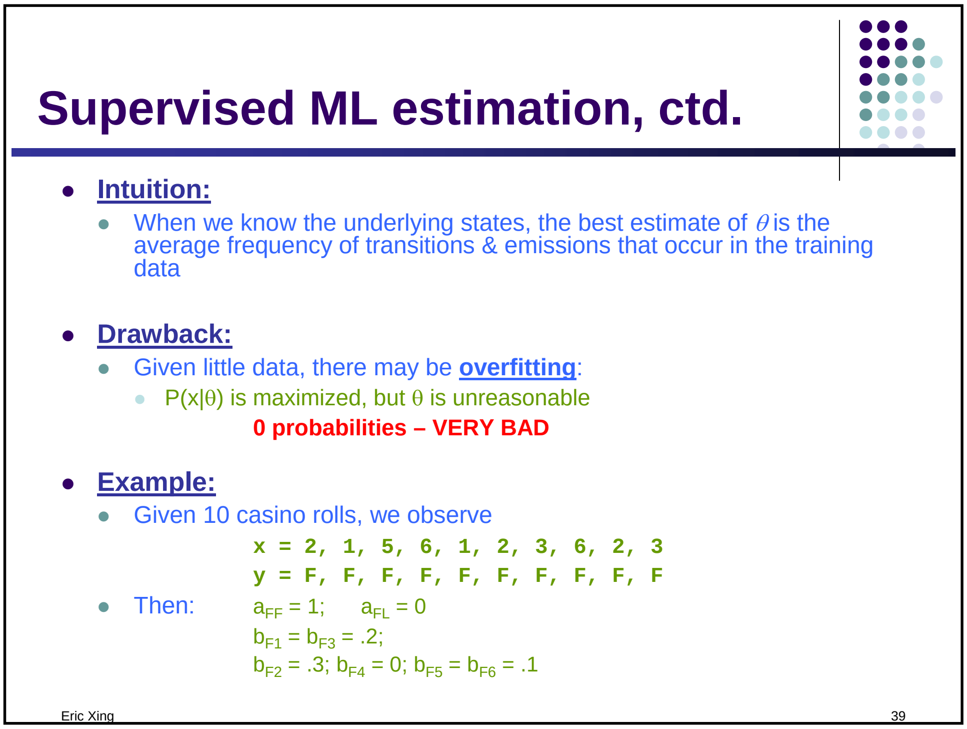
    B_{ik} = # times state i in $\mathbf{y}$ emits k in $\mathbf{x}$

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

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

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

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



Supervised ML estimation, ctd.

- **Intuition:**
  - When we know the underlying states, the best estimate of θ is the average frequency of transitions & emissions that occur in the training data

- **Drawback:**
  - Given little data, there may be **overfitting**:
    - P(x|θ) is maximized, but θ is unreasonable

      0 probabilities – VERY BAD

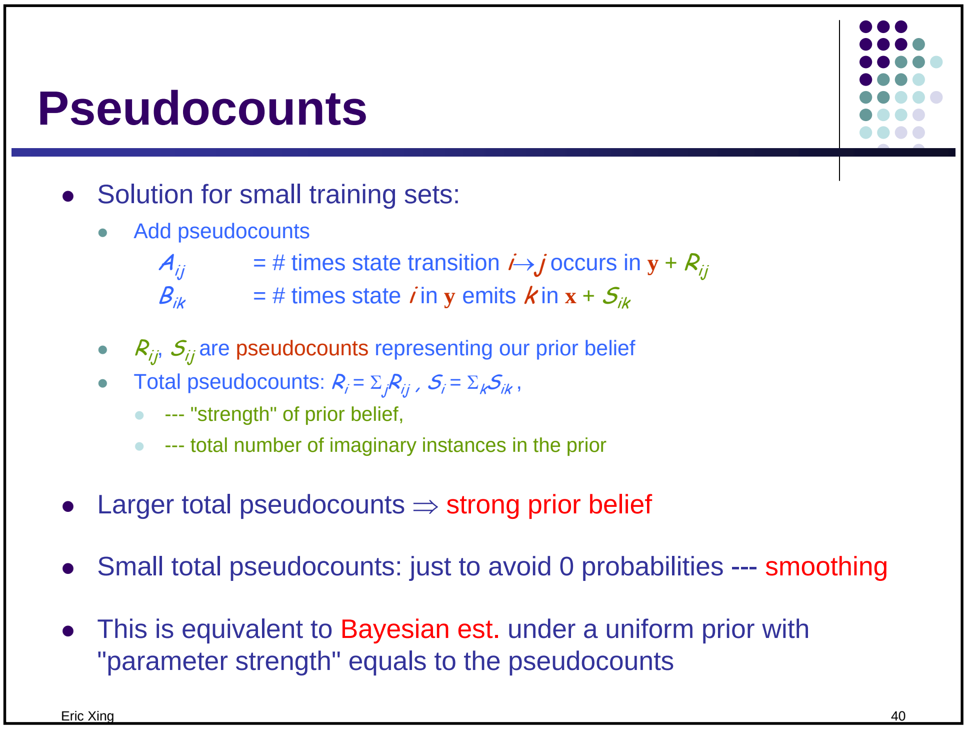
- **Example:**
  - Given 10 casino rolls, we observe

```
x = 2, 1, 5, 6, 1, 2, 3, 6, 2, 3
y = F, F, F, F, F, F, F, F, F, F
```

  - Then: $a_{FF} = 1$; $a_{FL} = 0$

    $b_{F1} = b_{F3} = .2$;

    $b_{F2} = .3$; $b_{F4} = 0$; $b_{F5} = b_{F6} = .1$



Pseudocounts

- Solution for small training sets:
  - Add pseudocounts

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

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

  - R_{ij}, S_{ij} are pseudocounts representing our prior belief
  - Total pseudocounts: $R_i = \Sigma_j R_{ij}$, $S_i = \Sigma_k S_{ik}$,
    - --- "strength" of prior belief,
    - --- total number of imaginary instances in the prior

- Larger total pseudocounts ⇒ strong prior belief

- Small total pseudocounts: just to avoid 0 probabilities --- smoothing

- This is equivalent to Bayesian est. under a uniform prior with "parameter strength" equals to the pseudocounts