CS839:
Probabilistic Graphical Models

Lecture 7: Learning Fully Observed BNs

Theo Rekatsinas

Exponential family: a basic building block

- For a numeric random variable X

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

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

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

Why exponential family?

- Moment generating property

$$\frac{dA}{d\eta} = \frac{d}{d\eta}\log Z(\eta) = \frac{1}{Z(\eta)}\frac{d}{d\eta}Z(\eta)$$

$$= \frac{1}{Z(\eta)}\frac{d}{d\eta}\int h(x)\exp\{\eta^T T(x)\}dx$$

$$= \int T(x)\frac{h(x)\exp\{\eta^T T(x)\}}{Z(\eta)}dx$$

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

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

$$= E[T^2(x)] - E^2[T(x)]$$

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

We can easily compute moments of any exponential family distribution by taking the derivatives of the log normalizer A(η)

MLE for Exponential Family

- For iid data the log-likelihood is

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

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

- We take the derivatives and set them to zero

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

- We perform **moment matching**

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

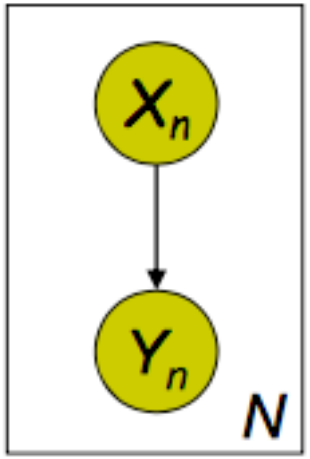
- We can infer the canonical parameters using

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

$$\hat{\eta}_{MLE} = \psi(\hat{\mu}_{MLE})$$

Generalized Linear Models

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

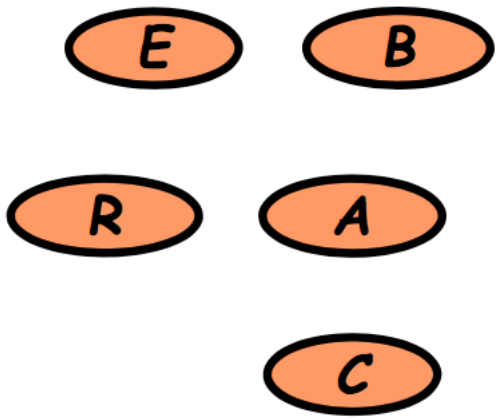


$$E_p(T) = \mu = f(\theta^T X)$$

- Generalized Linear Model
  - The observed input x is assumed to enter into the model via a linear combination of its elements $\xi = \theta^T \mathbf{x}$.
  - The conditional mean μ is represented as a function f(ξ) 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.

Learning in Graphical Models

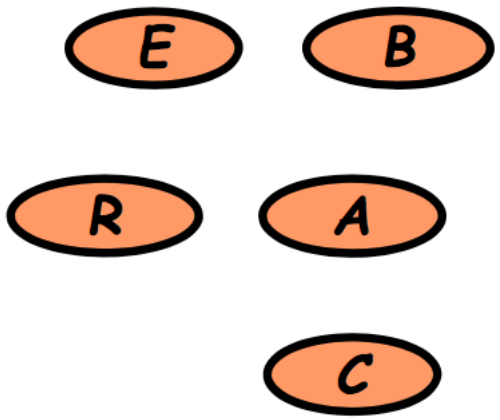
- Goal: Given a set of independent samples (**assignments** to random variables), find the **best** Bayesian Network (both DAG and CPDs)

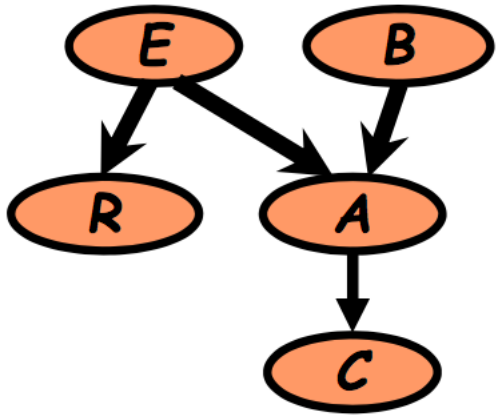


(B,E,A,C,R) = (T,F,F,T,F)
(B,E,A,C,R) = (T,F,T,T,F)
…
(B,E,A,C,R) = (F,T,T,T,F)

Learning in Graphical Models

- Goal: Given a set of independent samples (**assignments** to random variables), find the **best** Bayesian Network (both DAG and CPDs)



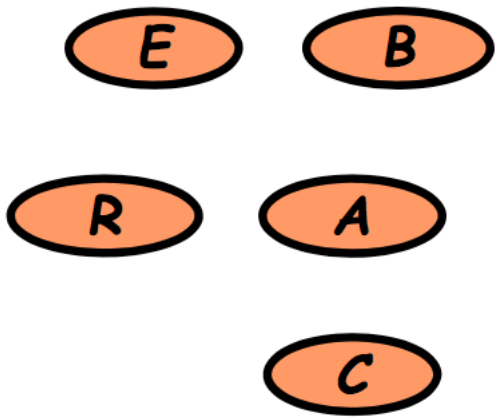


Structure learning

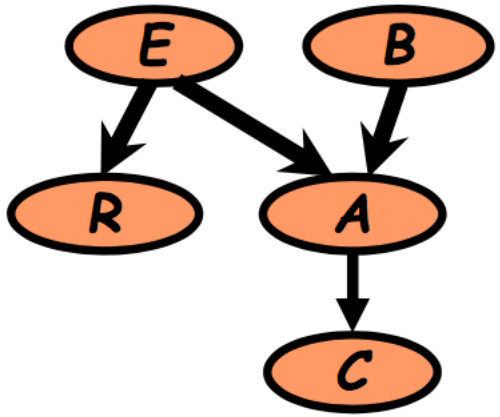
(B,E,A,C,R) = (T,F,F,T,F)
(B,E,A,C,R) = (T,F,T,T,F)
…
(B,E,A,C,R) = (F,T,T,T,F)

Learning in Graphical Models

- Goal: Given a set of independent samples (**assignments** to random variables), find the **best** 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)



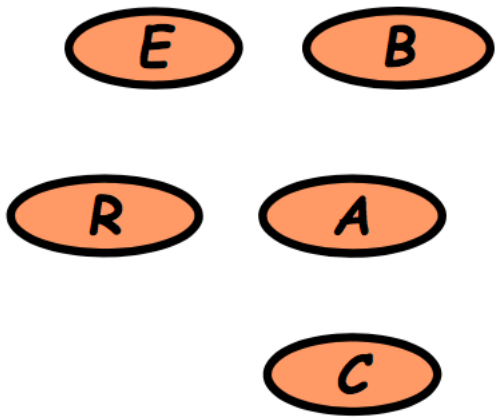
Structure learning

E	B	P(A | E,B)	
e	b	0.9	0.1
e	b	0.2	0.8
e	b	0.9	0.1
e	b	0.01	0.99

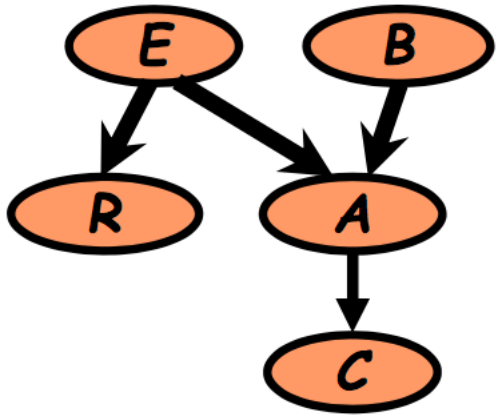
Parameter learning

Learning in Graphical Models

- Goal: Given a set of independent samples (**assignments** to random variables), find the **best** 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)



Structure learning

Later in the class

E	B	P(A | E,B)	
e	b	0.9	0.1
e	b	0.2	0.8
e	b	0.9	0.1
e	b	0.01	0.99

Parameter learning

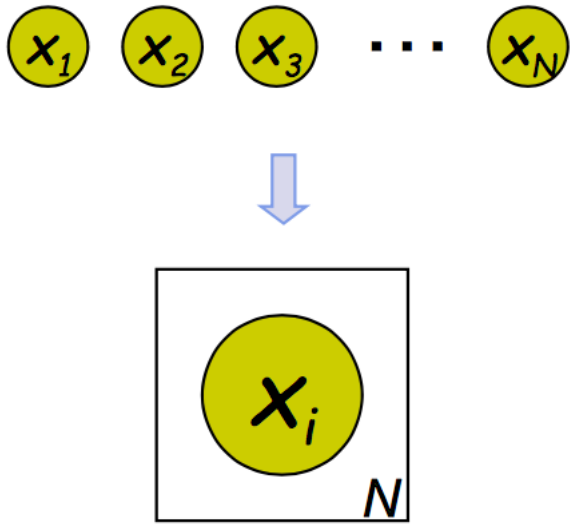
Parameter Estimation for Fully Observed GMs

- The data: D = (x1, x2, x3, … , xN)
- Assume the graph G is known and fixed
  - Expert design or structure learning
- Goal: estimate from a dataset of N **independent, identically distributed (iid)** training examples D
- Each training example corresponds to a vector of M values one per node random variable
  - Model should be completely observable: no missing values, no hidden variables

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

Density estimation

- A construction of an **estimate**, based on **observed data**, of an unobservable underlying probability density function



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

Discrete Distributions

- Bernoulli distribution: $P(x) = p^x(1-p)^{1-x}$

- Multinomial distribution: Mult(1, θ)

$$X = [X_1, X_2, X_3, X_4, X_5, X_6] \quad X_j = [0,1], \sum_{j \in [1, \cdots, 6]} X_j = 1$$

$$X_j = 1 \text{ with probability } \theta_j, \sum_{j \in [1, \cdots, 6]} theta_j = 1$$

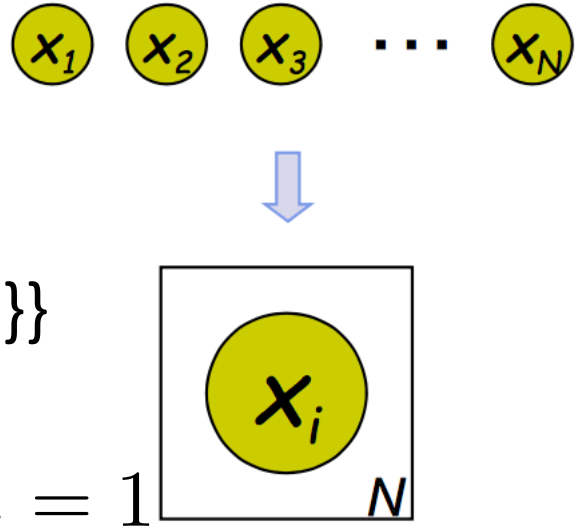
$$P(X_j = 1) = \theta_j$$

Discrete Distributions

- Multinomial distribution: Mult(n, θ)

$$n = [n_1, n_2, \ldots, n_k] \text{ where } \sum_j n_j = N$$

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



Example: multinomial model

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

$$x_n = [x_{n,1}, x_{n,2}, \cdots, x_{n,K}] \text{ where } x_{n,k} = 0, 1 \sum_{k=1}^{K} x_{n,k} = 1$$

- Model: $$X_{n,k} = 1 \text{ with probability } \theta_k \text{ and } \sum_{k \in \{1, \cdots, K\}} \theta_k = 1$$

- Likelihood of an observation: $$P(x_i) = P(\{x_{n,k} = 1, \text{ where } k \text{ is the index of the n-th roll}\})$$
$$= \theta_k = \theta_1^{x_{n,1}} \theta_2^{x_{n,2}} \cdots \theta_k^{x_{n,k}} = \prod_{k=1}^{K} \theta_k^{x_{n,k}}$$

- Likelihood of D: $$P(x_1, x_2, \ldots, x_N | \theta) = \prod_{n=1}^{N} P(x_n | \theta) = \prod_{k} \theta_k^{n_k}$$

MLE: constrained optimization

- Objective function: $l(\theta; D) = \log P(D|\theta) = \log \prod_k \theta_k^{n_k} = \sum_k n_k \log \theta_k$

- We need to maximize this subject to the constraint: $\sum_{k \in \{1, \cdots, K\}} \theta_k = 1$

- Lagrange multipliers: $\bar{l}(\theta; D) = \sum_k n_k \log \theta_k + \lambda(1 - \sum_k \theta_k)$

- Derivatives: $\frac{\partial \bar{l}}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0$

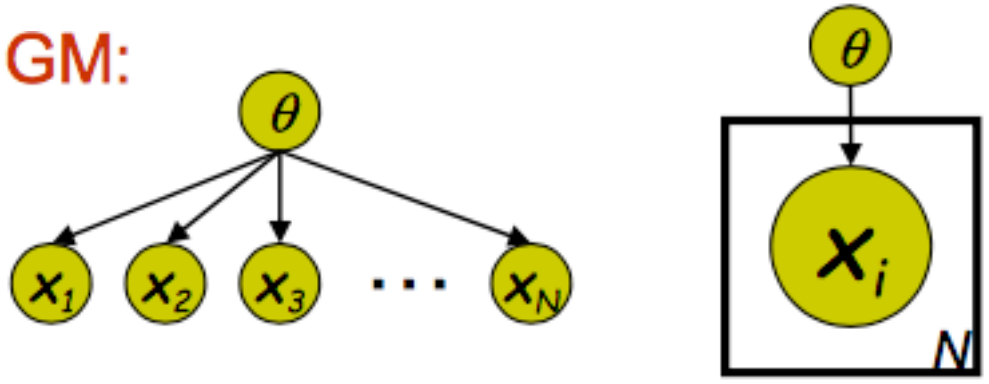
  $n_k = \lambda \theta_k \Rightarrow \sum_k n_k = \lambda \sum_k \theta_k \Rightarrow N = \lambda$

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

- Sufficient statistics?

Bayesian estimation

- I need a prior over parameters θ



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

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

  - Isomorphism of the posterior with the prior (**conjugate prior**)

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

Continuous Distributions

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

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

- 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

- You can show that the MLE for μ and Σ is

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

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

- What are the sufficient statistics?

MLE for a multivariate Gaussian

- You can show that the MLE for μ and Σ is

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

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

- What are the sufficient statistics?
  - Rewrite

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

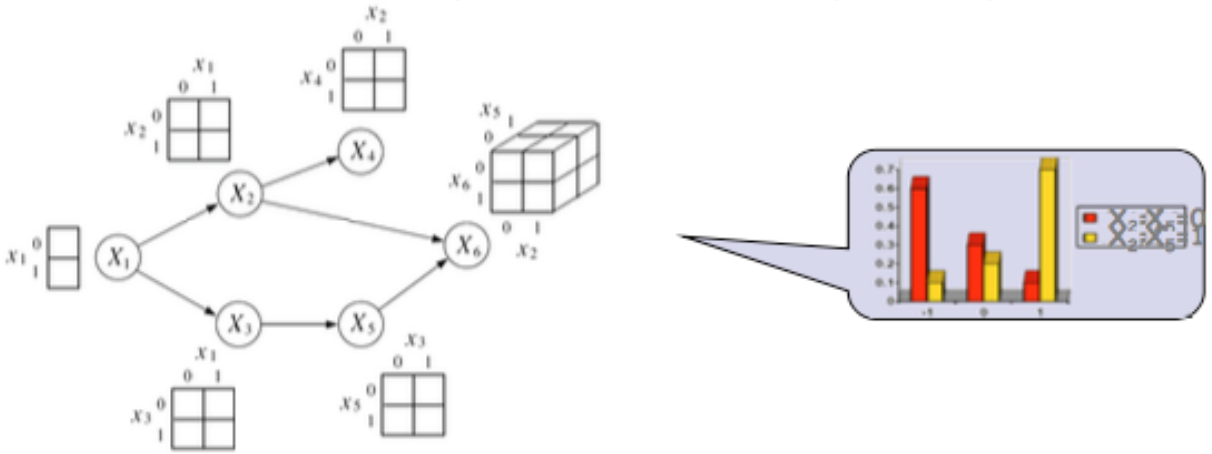
  - Sufficient statistics are: $\sum_n (x_n) \quad \left(\sum_n x_n x_n^T{}_n\right).$

- Similar for Bayesian estimation: Normal prior

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

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



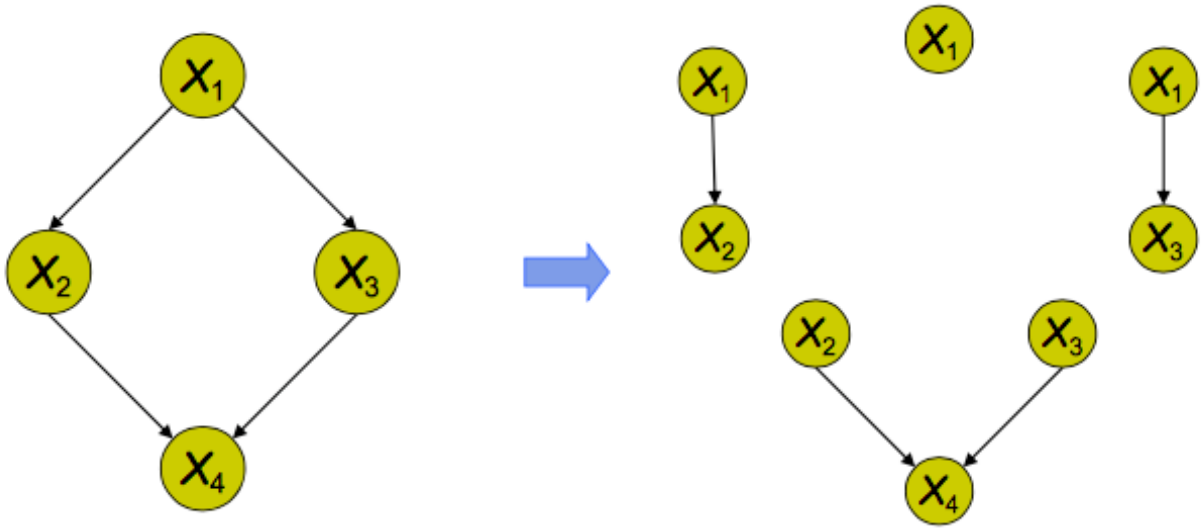
- MLE-based parameter estimation of GM reduces to local est. of each GLIM.

Decomposable likelihood of a BN

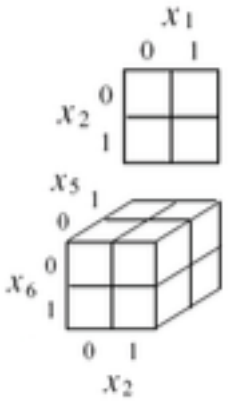
- Consider the GM:

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

- This is the same as learning four separate smaller BNs each of which consists of a node an its parents.



MLE for BNs with tabular CPDs



- Each CPD is represented as a table (multinomial) with

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

  - In case of multiple parents the CPD is a high-dimensional table
  - The sufficient statistics are counts of variable configurations

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

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

- And using a Lagrange multiplier to enforce that conditionals sum up to 1 we have:

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

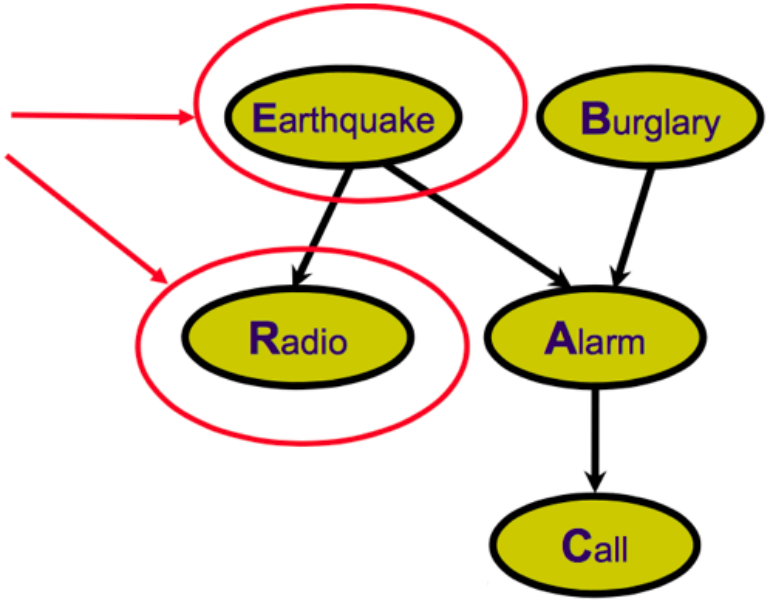
What about parameter priors?

- In a BN we have a collection of local distributions $p(x_i^k \mid \mathbf{x}_{\pi_i}^j) = \theta_{x_i^k \mid \mathbf{x}_{\pi_i}^j}$
- How can we define priors over the whole BN?
- We could write P(x1,x2,…xN;G,θ)P(θ|α)
  - Symbolically the same as before but θ is defined over a vector of random variables that follow different distributions.
  - We need θ to decompose to use local rules. Otherwise we cannot decompose the likelihood any more.
- We need certain rules on θ
  - Complete Model Equivalence
  - **Global Parameter Independence**
  - **Local Parameter Independence**
  - Likelihood and Prior Modularity

Global and 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

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

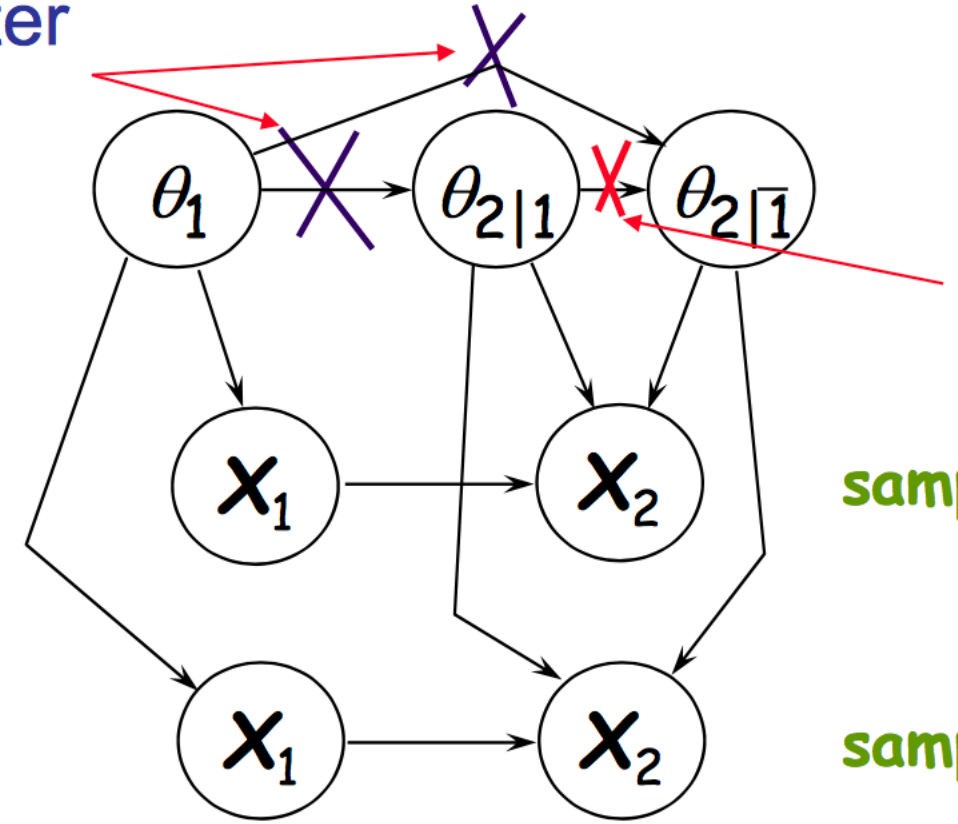
$$P(\theta_{Call \mid Alarm=YES})$$

independent of

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

Global and Local Parameter Independence

Global Parameter Independence



Local Parameter Independence

sample 1

sample 2

Which PDFs satisfy these assumptions?

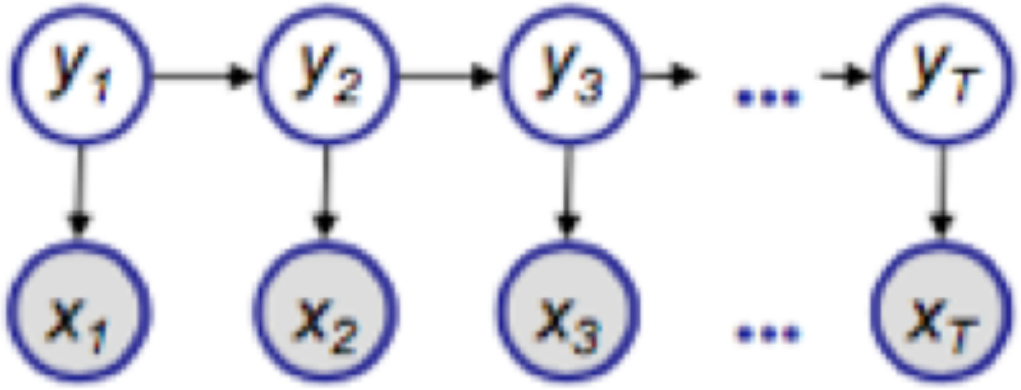
- Discrete DAG Models $x_i \mid \pi_{x_i}^j \sim \text{Multi}(\theta)$

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

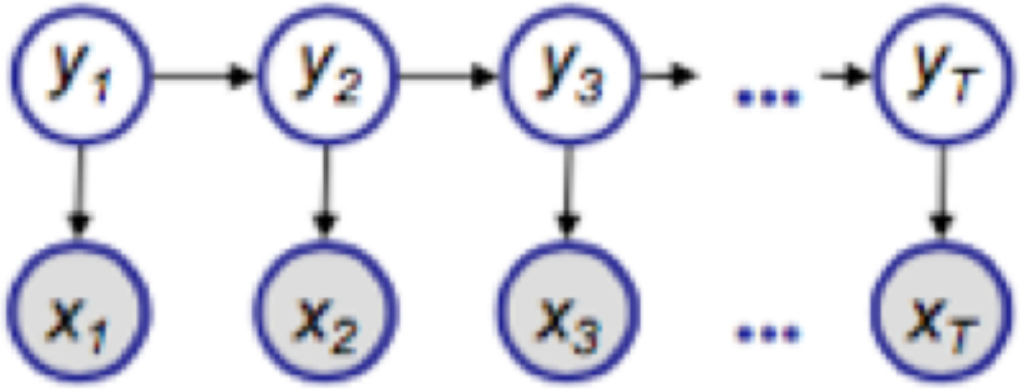
Normal prior: $$p(\mu \mid \nu, \Psi) = \frac{1}{(2\pi)^{n/2} \mid \Psi \mid^{1/2}} \exp\left\{ -\frac{1}{2} (\mu - \nu)' \Psi^{-1} (\mu - \nu) \right\}$$

Parameter sharing



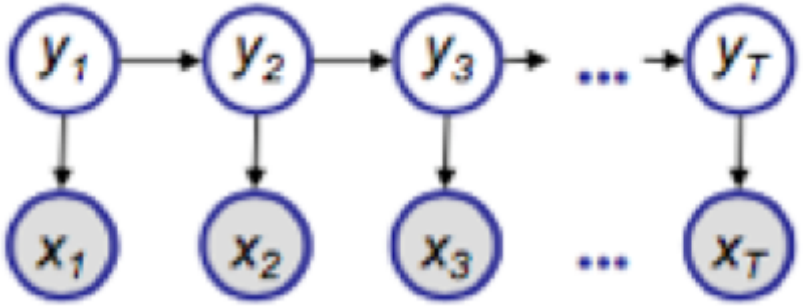
- Transition probabilities $P(X_t|X_{t-1})$ can be different: What is the parameterization cost?

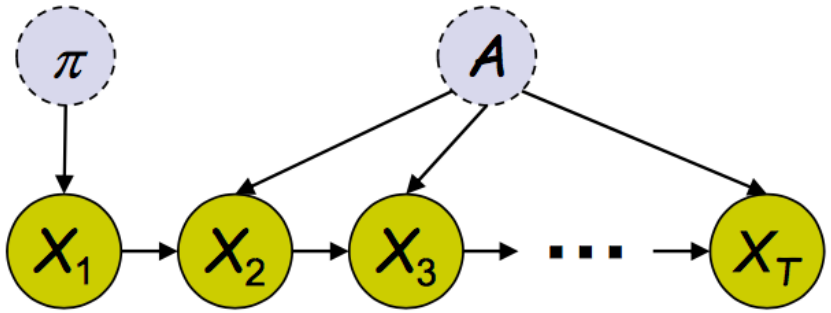
Parameter sharing



- Transition probabilities $P(X_t|X_{t-1})$ can be different: What is the parameterization cost?
  - I need a different local conditional distribution.
  - How can I learn the transition probabilities? I need iid data. So my training examples should correspond multiple rolls of the dice. I need to do alignment etc.
  - What do we do?

Parameter sharing





- We will make the assumption that for every transition we follow the same conditional
- Consider a time-invariant (stationary) 1st-order Markov model
  - Initial state probability vector
  - State transition probability matrix $\pi_k \stackrel{\text{def}}{=} p(X_1^k = 1)$ $A_{ij} \stackrel{\text{def}}{=} p(X_t^j = 1 \mid X_{t-1}^i = 1)$
- Now: $p(X_{1:T} \mid \theta) = p(x_1 \mid \pi) \prod_{t=2}^{T} \prod_{t=2} p(X_t \mid X_{t-1})$ optimize separately
  - π (multinomial)
  - What about A?

Learning a Markov chain transition matrix

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

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

- Sparse data problem:
  - If i to j did not occur in the data then A_{ij} is 0. Any future sequence with pair i to j will have zero probability
  - Backoff smoothing $\tilde{A}_{i\to\bullet} = \lambda \eta_t + (1-\lambda) A_{i\to\bullet}^{ML}$

Example: HMM supervised ML estimation

- Given x = x1, x2, …, xN for which the true state path y1, y2, …, yN is known:
  - Define

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

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

- The maximum likelihood parameters θ are:

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

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

- If x is continuous we can apply learning rules for a Gaussian

Supervised ML estimation

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

- **Drawback:** Given little data, we may **overfit** (remember zero probabilities)

- **Example:**
  - Given 10 rolls we have

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

  - Then

    $a_{FF} = 1$; $a_{FL} = 0$
    $b_{F1} = b_{F3} = .2$;
    $b_{F2} = .3$; $b_{F4} = 0$; $b_{F5} = b_{F6} = .1$

Pseudocounts

- Solution for small training sets:
  - Add pseudocounts

$$A_{ij} = \text{\# times state transition } i \rightarrow j \text{ occurs in } \mathbf{y} + R_{ij}$$
$$B_{ik} = \text{\# times state } i \text{ in } \mathbf{y} \text{ emits } k \text{ in } \mathbf{x} + S_{ik}$$

  - The pseudocounts represent our prior belief

- Large total pseudocounts => strong prior belief
- Small total pseudocounts => smoothing to avoid 0 probabilities
- Equivalent to Bayesian estimation under a uniform prior with parameter strength equal to the pseudocounts.

Summary

- 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: exp. Family
  - Typical discrete distribution
  - Typical continuous distribution
  - Conjugate priors

- Learning BN with more nodes:
  - Local operations