Probabilistic Graphical Models & Probabilistic AI

Ben Lengerich

Lecture 8: Parameter Learning in Fully-Observed BNs

February 20, 2025

Reading: See course homepage

A Follow-up on Project Ideas

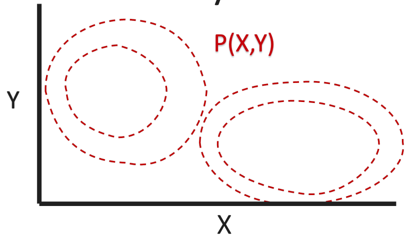
- What does “novel” mean?
  - Something is uniquely **yours**
- Questions? Please ask.

Today

- HW3 + Feedback
- Parameter Learning in Fully-Observed BNs

A Brief Recap of our Roadmap

Probability Distribution



Observations

X	Y
0	0
0.1	1
0.2	1

Prior Knowledge

Parametric Family P_θ

Regularizer / Prior

Disc / Gen Models - Lecture 3

Estimation $\hat{\theta}$

MLE / MAP - Lecture 2

A Brief Recap of our Roadmap

Local Structure

Prior Knowledge

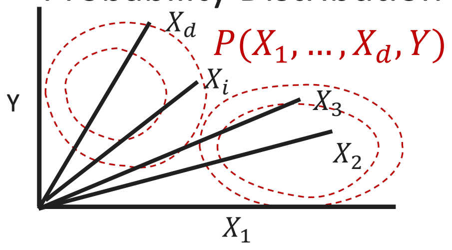
Graphs - Lectures 4,5

Parametric Family P_θ

Regularizer / Prior

Disc / Gen Models - Lecture 3

Probability Distribution



Observations

X	Y
0	0
0.1	1
0.2	1

Estimation → $\hat{\theta}$

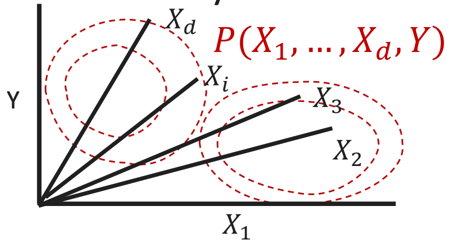
MLE / MAP - Lecture 2

Inference → $P_{\hat{\theta}}(\cdot)$

Exact - Lecture 6

A Brief Recap of our Roadmap

Probability Distribution



Observations

X	Y
0	0
0.1	1
0.2	1

Structure Learning - Lectures 10, 11

Local Structure

Architecture Design, Lectures 16-23

Prior Knowledge

Graphs - Lectures 4,5

Parametric Family P_θ

Regularizer / Prior

Disc / Gen Models - Lecture 3

Estimation $\hat{\theta}$

MLE / MAP - Lecture 2

Learning on Graphs - **Lectures 7 (8)** - 9, 12-13

Inference $P_{\hat{\theta}}(\cdot)$

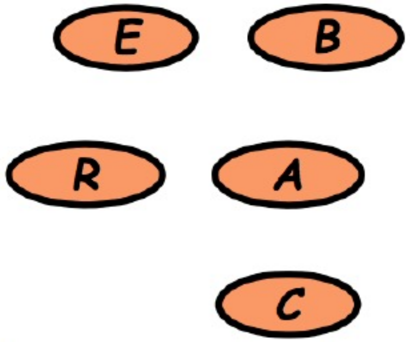
Exact - Lecture 6

Approximate - Lecture 14

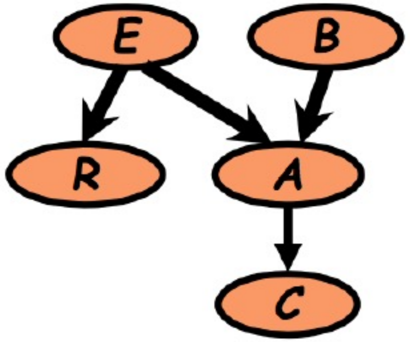
Parameter Learning in Fully-Observed Bayesian Networks

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	$P(A \mid E,B)$	
e	$\underline{b}$	0.9	0.1
$\underline{e}$	b	0.2	0.8
$\underline{e}$	$\underline{b}$	0.9	0.1
e	b	0.01	0.99

Parameter learning

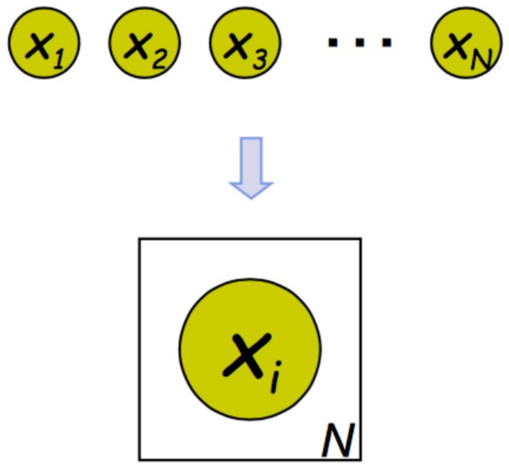
Parameter Estimation for Fully-Observed BNs

- 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

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

Simplest case: 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

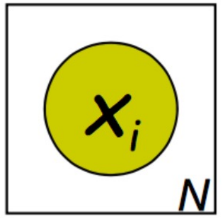
- Multinomial distribution: Mult(n, θ)

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

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

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

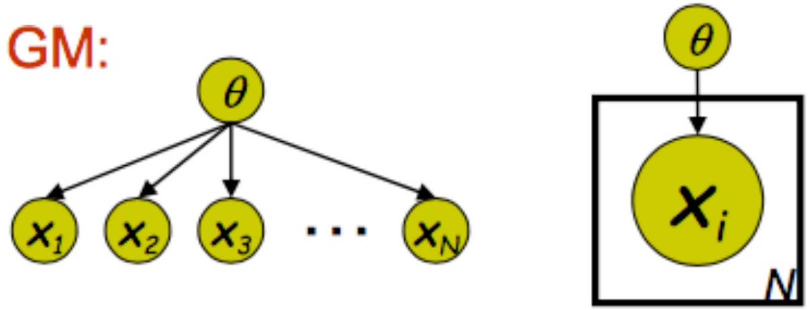
$$n_k = \lambda \theta_k \Rightarrow \sum_k n_k = \lambda \sum_k \theta_k \Rightarrow N = \lambda$$

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

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

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

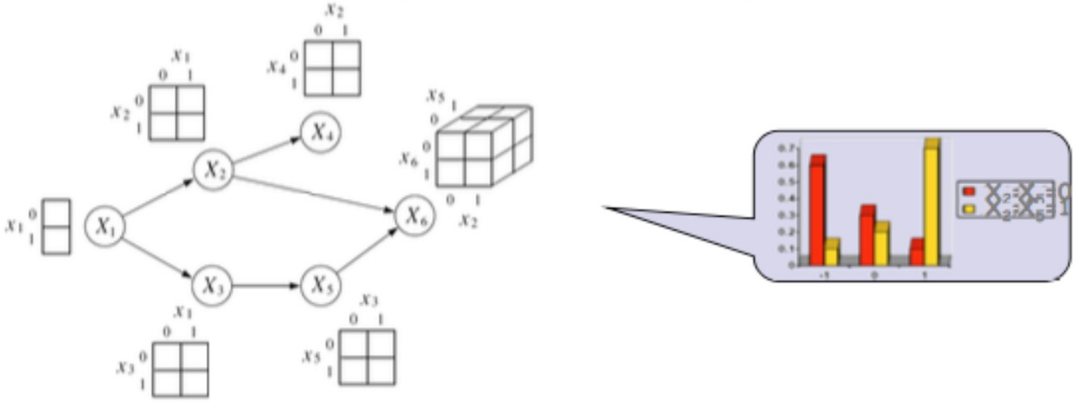
$$S = \sum_n (x_n - \mu_{ML})(x_n - \mu_{ML})^T = \left(\sum_n x_n x_n^T\right) - N\mu_{ML}\mu_{ML}^T$$

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

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

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



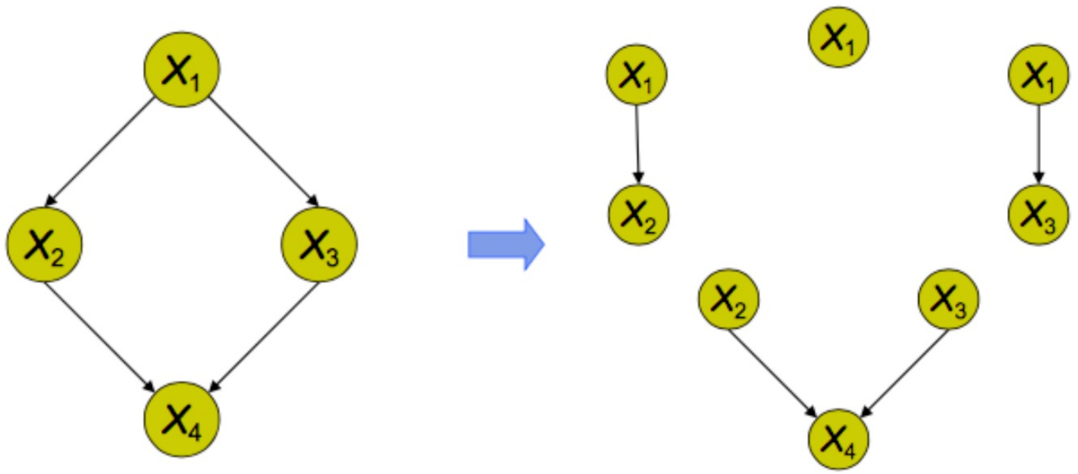
- MLE-based parameter estimation of GM reduces to local est. of each GLIM.

Decomposable likelihood of a BN

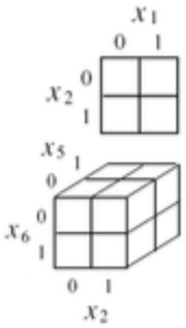
- Consider the GM:

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

- 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:

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

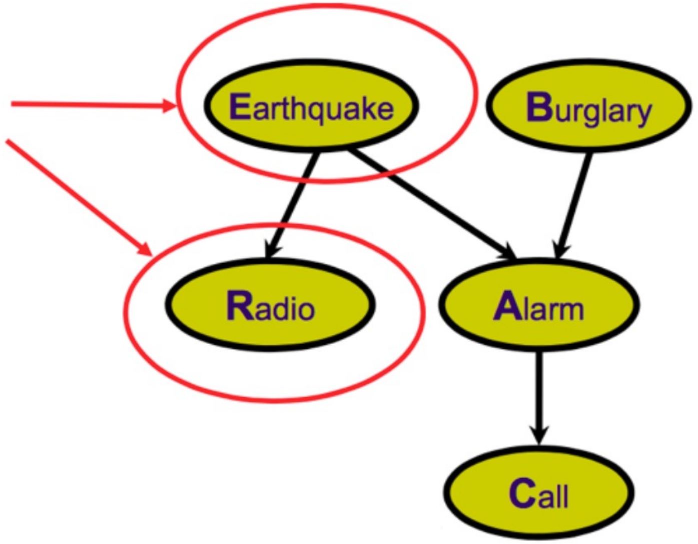
What about parameter priors?

- 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

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

Which PDFs satisfy these assumptions?

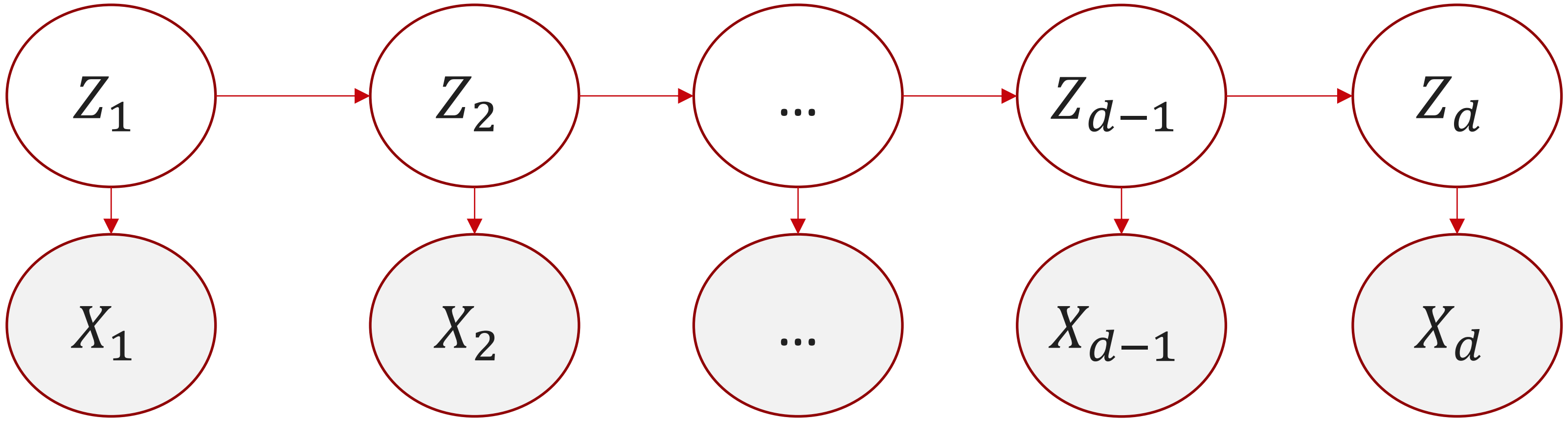
- Discrete DAG Models $x_i \mid \pi_{x_i}^{j} \sim \mathrm{Multi}(\theta)$

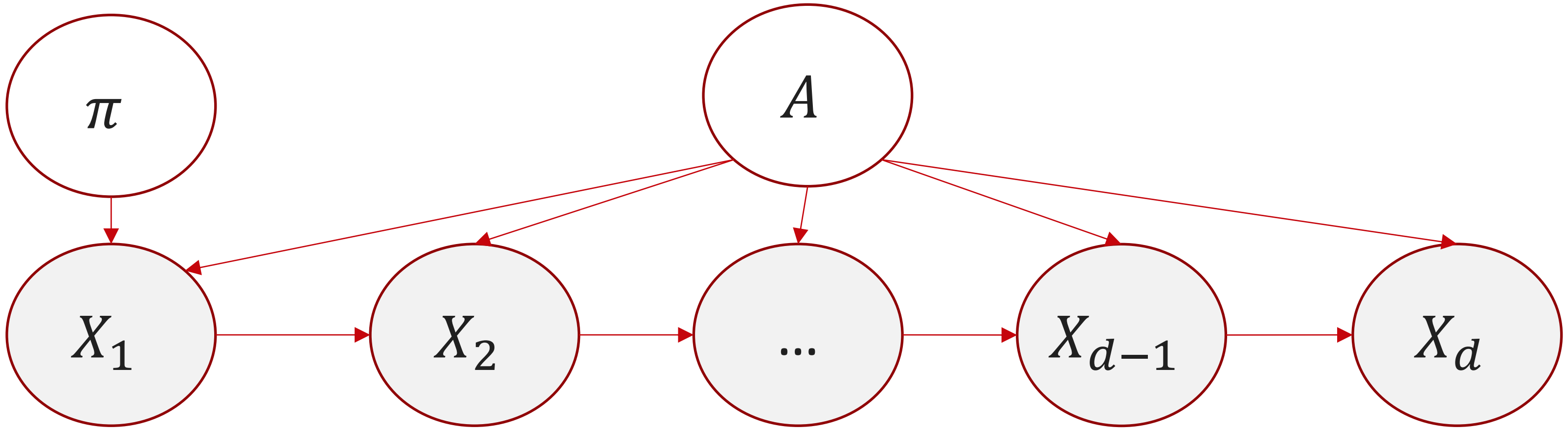
Dirichlet prior: $$P(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} = C(\alpha) \prod_k \theta_k^{\alpha_k - 1}$$

- Gaussian DAG Models $x_i \mid \pi_{x_i}^{j} \sim \mathrm{Normal}(\mu, \Sigma)$

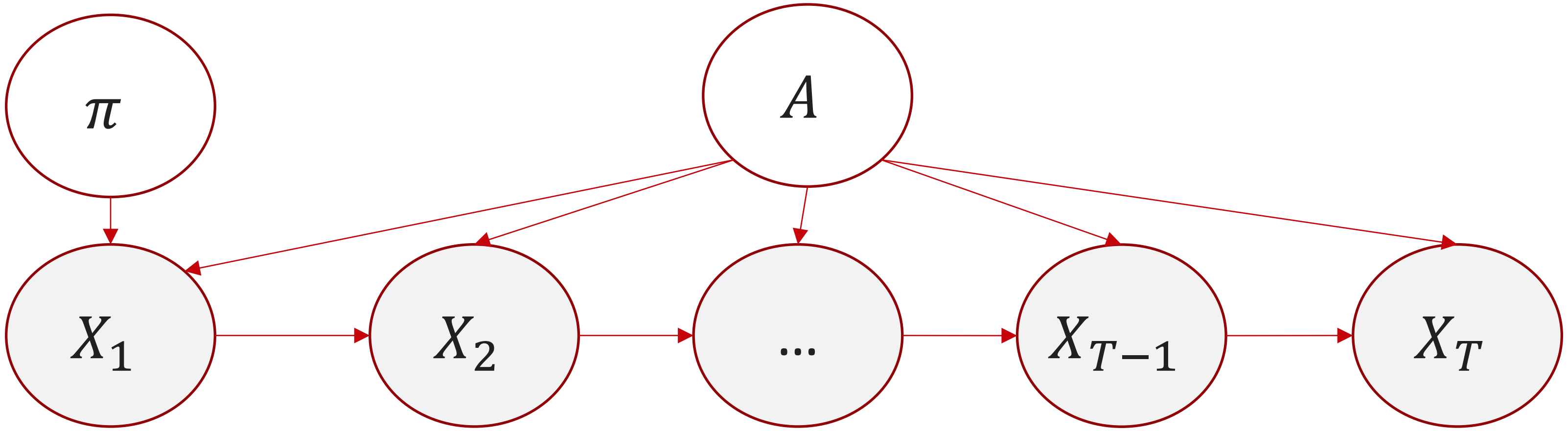
Normal prior: $$p(\mu \mid \nu, \Psi) = \frac{1}{(2\pi)^{n/2} |\Psi|^{1/2}} \exp\left\{ -\frac{1}{2} (\mu - \nu)' \Psi^{-1} (\mu - \nu) \right\}$$

Parameter Sharing





Parameter Sharing



- Now: $p(X_{1:T} \mid \theta) = p(x_1 \mid \pi) \prod_{t=2} \prod_{t=2} p(X_t \mid X_{t-1})$ optimize separately
  - π (multinomial)
  - What about A?

- 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

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

Key idea today

For fully-observed BNs, the log-likelihood function **decomposes** into a sum of local terms → **Learning is factored**

Questions?

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON