

CS 6347

Lecture 17

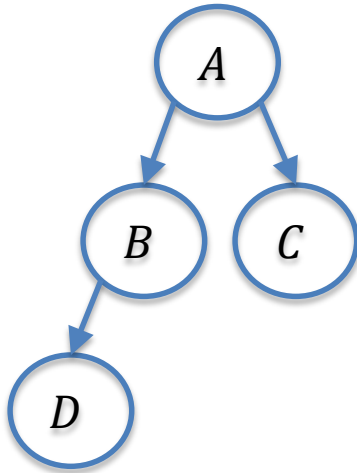
Introduction to Structure Learning

- We have been focusing on parameter learning:
 - E.g., given a graph structure, find the parameters that maximize the log-likelihood
- In practice, the structure of the graph may not be known and may need to be learned from the data
 - For Bayesian networks, we may be only given samples and asked to make predictions

BN Structure Learning



- Recall that for a fixed Bayesian network with fully observed data, the MLE of the conditional probability tables was given by the empirical probabilities

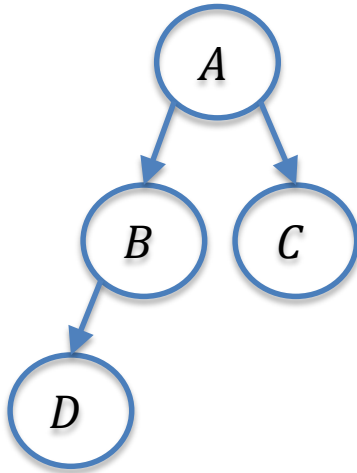


A	B	C	D
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	1
0	0	1	1

BN Structure Learning



- Recall that for a fixed Bayesian network with fully observed data, the MLE of the conditional probability tables was given by the empirical probabilities



A	B	C	D
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	1
0	0	1	1

A	P(A)
0	4/5
1	1/5

A	B	P(B A)
0	0	3/4
0	1	1/4
1	0	1
1	1	0

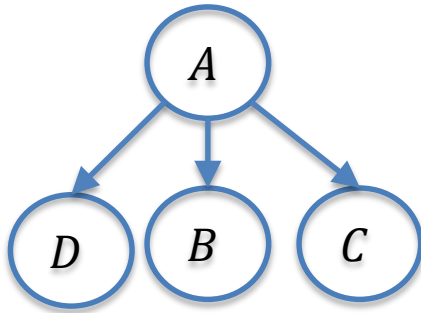
B	D	P(D B)
0	0	1/4
0	1	3/4
1	0	1
1	1	0

A	C	P(C A)
0	0	1/4
0	1	3/4
1	0	1
1	1	0

BN Structure Learning



- Recall that for a fixed Bayesian network with fully observed data, the MLE of the conditional probability tables was given by the empirical probabilities



A	B	C	D
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	1
0	0	1	1

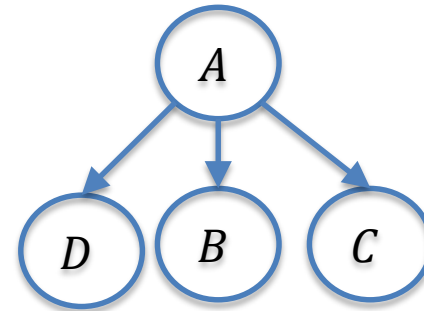
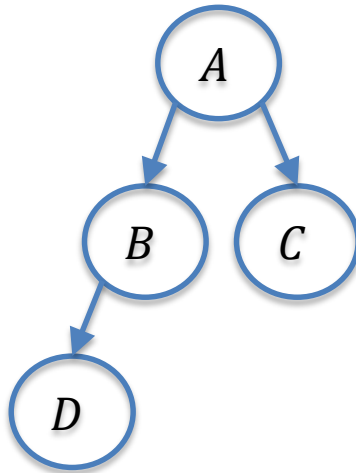
A	P(A)
0	4/5
1	1/5

A	B	P(B A)
0	0	3/4
0	1	1/4
1	0	1
1	1	0

A	D	P(D A)
0	0	1/2
0	1	1/2
1	0	0
1	1	1

A	C	P(C A)
0	0	1/4
0	1	3/4
1	0	1
1	1	0

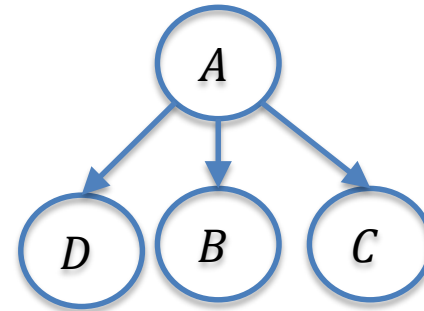
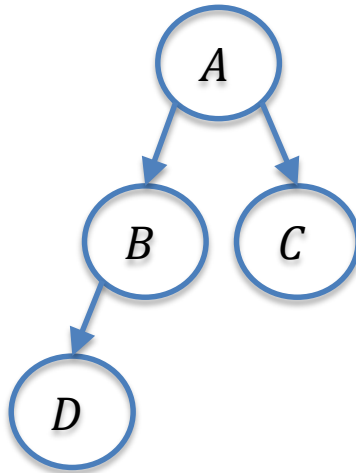
- Which model should be preferred?



BN Structure Learning



- Which model should be preferred?

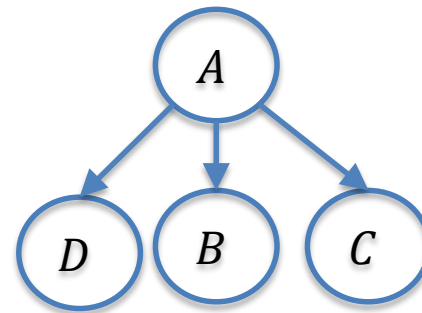
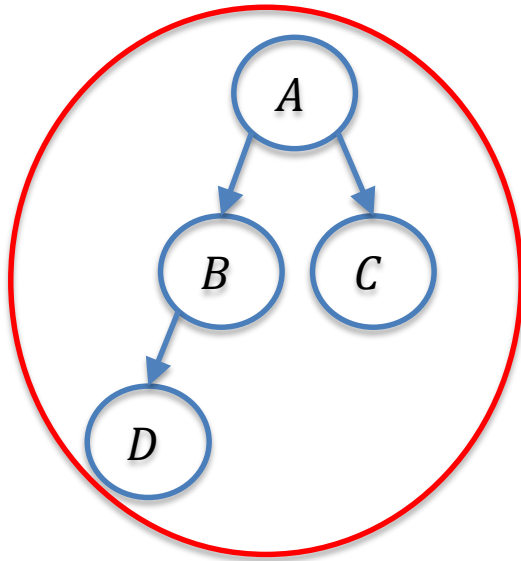


Which one has the highest log-likelihood given the data?

BN Structure Learning



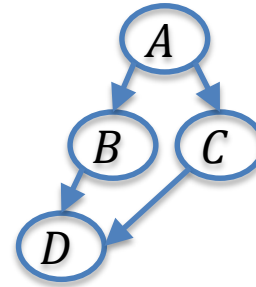
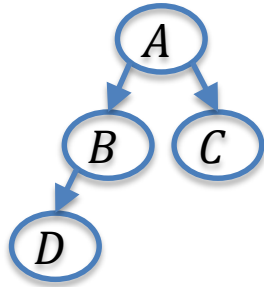
- Which model should be preferred?



Which one has the highest log-likelihood given the data?

- Determining the structure that maximizes the log-likelihood is not too difficult
 - A complete DAG always maximizes the log-likelihood
 - This almost certainly results in overfitting
- Alternative is to attempt to learn simple structures
 - Approach 1: Optimize the log-likelihood over simple graphs
 - Approach 2: Add a penalty term to the log-likelihood

Adding Edges Increases the MLE



Let p' be the empirical probability distribution

$$\begin{aligned}\frac{\ell_2 - \ell_1}{M} &= \frac{1}{M} \sum_m \log \frac{p'(x_D^m | x_B^m, x_C^m)}{p'(x_D^m | x_B^m)} \\ &= \sum_x p'(x_B, x_C, x_D) \log \frac{p'(x_D | x_B, x_C)}{p'(x_D | x_B)} \\ &= \sum_x p'(x_B, x_C, x_D) \log \frac{p'(x_B, x_C, x_D)}{p'(x_C | x_B) p'(x_D | x_B) p'(x_B)} \\ &= d(p'(x_B, x_C, x_D) || p'(x_C | x_B) p'(x_D | x_B) p'(x_B)) \geq 0\end{aligned}$$

- Suppose that we want to find the best tree-structured BN that represents a given joint probability distribution
 - Find the tree-structured BN that maximizes the likelihood
- Let's consider the log-likelihood of a fixed tree T
 - Assume that the edges are directed so that each node has exactly one parent

For a fixed tree:

$$\begin{aligned}\max_{\theta} \log l(\theta, T) &= \sum_{i \in V(T)} \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_{\text{parent}(i)}}} \\ &= \sum_{i \in V(T)} \left[\sum_{x_i} N_{x_i} \log N_{x_i} + \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_i} N_{x_{\text{parent}(i)}}} \right] \\ &= \left[\sum_{i \in V} \sum_{x_i} N_{x_i} \log N_{x_i} \right] + \left[\sum_{(i,j) \in E(T)} \sum_{x_i, x_j} N_{x_i, x_j} \log \frac{N_{x_i, x_j}}{N_{x_i} N_{x_j}} \right]\end{aligned}$$

For a fixed tree:

$$\begin{aligned}
 \max_{\theta} \log l(\theta, T) &= \sum_{i \in V(T)} \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_{\text{parent}(i)}}} \\
 &= \sum_{i \in V(T)} \left[\sum_{x_i} N_{x_i} \log N_{x_i} + \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_i} N_{x_{\text{parent}(i)}}} \right] \\
 &= \left[\sum_{i \in V} \sum_{x_i} N_{x_i} \log N_{x_i} \right] + \left[\sum_{(i,j) \in E(T)} \sum_{x_i, x_j} N_{x_i, x_j} \log \frac{N_{x_i, x_j}}{N_{x_i} N_{x_j}} \right]
 \end{aligned}$$

Doesn't depend on the selected tree!

For a fixed tree:

$$\begin{aligned}
 \max_{\theta} \log l(\theta, T) &= \sum_{i \in V(T)} \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_{\text{parent}(i)}}} \\
 &= \sum_{i \in V(T)} \left[\sum_{x_i} N_{x_i} \log N_{x_i} + \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_i} N_{x_{\text{parent}(i)}}} \right] \\
 &= \left[\sum_{i \in V} \sum_{x_i} N_{x_i} \log N_{x_i} \right] + \left[\sum_{(i,j) \in E(T)} \sum_{x_i, x_j} N_{x_i, x_j} \log \frac{N_{x_i, x_j}}{N_{x_i} N_{x_j}} \right]
 \end{aligned}$$

This is almost the (empirical) **mutual information**

For a fixed tree:

$$\begin{aligned}
 \max_{\theta} \log l(\theta, T) &= \sum_{i \in V(T)} \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_{\text{parent}(i)}}} \\
 &= \sum_{i \in V(T)} \left[\sum_{x_i} N_{x_i} \log N_{x_i} + \sum_{x_{\text{parent}(i)}} \sum_{x_i} N_{x_i, x_{\text{parent}(i)}} \log \frac{N_{x_i, x_{\text{parent}(i)}}}{N_{x_i} N_{x_{\text{parent}(i)}}} \right] \\
 &= \left[\sum_{i \in V} \sum_{x_i} N_{x_i} \log N_{x_i} \right] + \left[\sum_{(i,j) \in E(T)} \sum_{x_i, x_j} N_{x_i, x_j} \log \frac{N_{x_i, x_j}}{N_{x_i} N_{x_j}} \right] \\
 &= \left[\sum_{i \in V} \sum_{x_i} N_{x_i} \log N_{x_i} \right] + M \left[\sum_{(i,j) \in E(T)} I(x_i; x_j) \right] - |E(T)| M \log M
 \end{aligned}$$

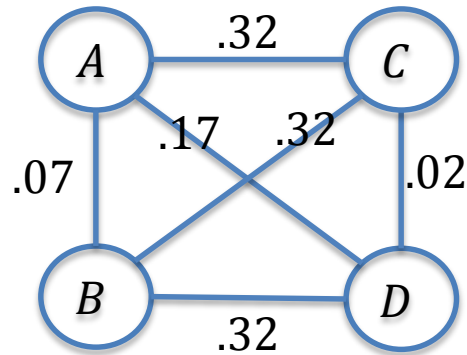
- To maximize the log-likelihood, it then suffices to choose the tree T that maximizes

$$\max_T \sum_{i,j} I(x_i; x_j)$$

- This problem can be solved by finding the maximum weight spanning tree in the complete graph with edge weight w_{ij} given by the mutual information over the edge (i, j)
 - Greedy algorithm works: at each step, pick the largest remaining edge that does not form a cycle when added to the already selected edges

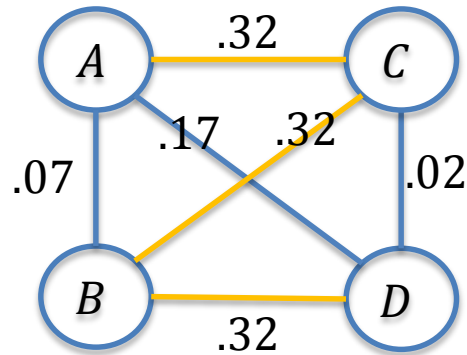
- To use this technique for learning, we simply compute the mutual information for each edge using the empirical probability distributions and then find the max-weight spanning tree
- As a result, we can learn tree-structured BNs in polynomial time
 - Can we generalize this to all DAGs?

Chow-Liu Trees: Example



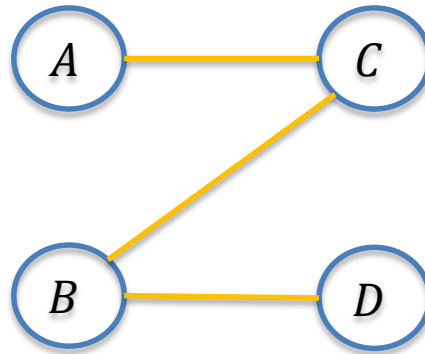
- Edge weights correspond to empirical mutual information for the earlier samples

Chow-Liu Trees: Example



- Edge weights correspond to empirical mutual information for the earlier samples

Chow-Liu Trees: Example



- Any directed tree (where each node has one parent) over these edges maximizes the log-likelihood
 - Why doesn't the direction matter?

Approach 2: Penalized Likelihood



- Add a penalty term to the log-likelihood that can depend on the number of samples and the chosen structure

$$\ell(G, \theta) = \sum_m \log p_G(x^m | \theta) - \eta(M) \text{Dim}(G)$$

- $\eta(M)$ is only a function of the number of samples
 - $\eta(M) = \text{constant}$ called the Akaike information criterion
 - $\eta(M) = \frac{\log(M)}{2}$ called the Bayesian information criterion
- $\text{Dim}(G)$ is the number of parameters needed to represent G