CS 6347

Lecture 19

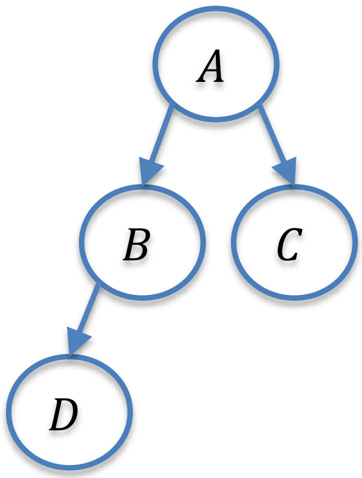
Introduction to Structure Learning

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

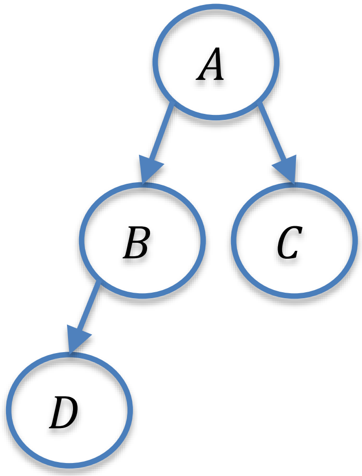
- Recall that for a fixed Bayesian network with fully observed data, the MLE of the conditional probability tables was given by the empirical probabilities



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

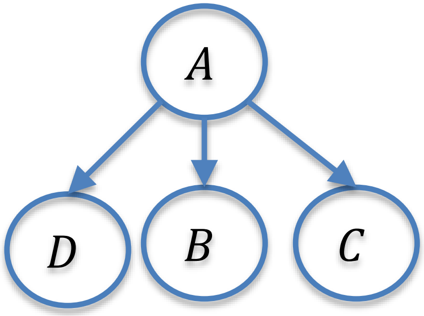
A	B	P(B|A)
0	0	3/4
0	1	1/4
1	0	1
1	1	0

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	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

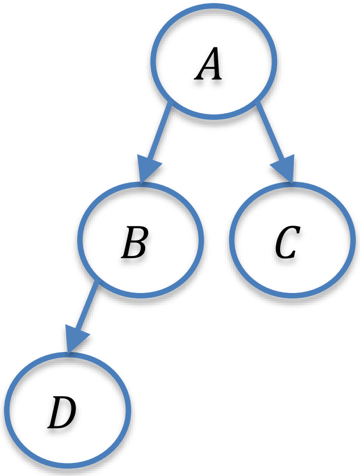
A	D	P(D|A)
0	0	1/2
0	1	1/2
1	0	0
1	1	1

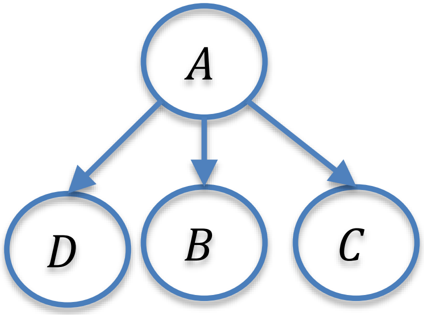
A	C	P(C|A)
0	0	1/4
0	1	3/4
1	0	1
1	1	0

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

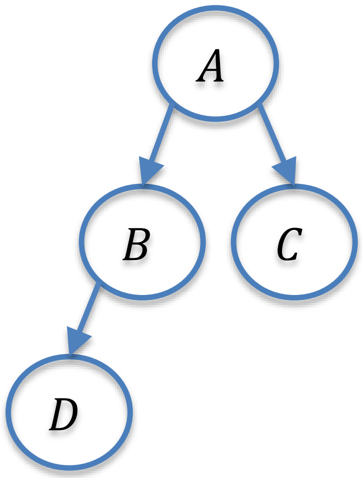
- Which model should be preferred?

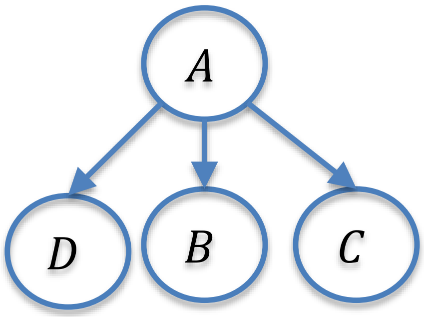




BN Structure Learning

- Which model should be preferred?

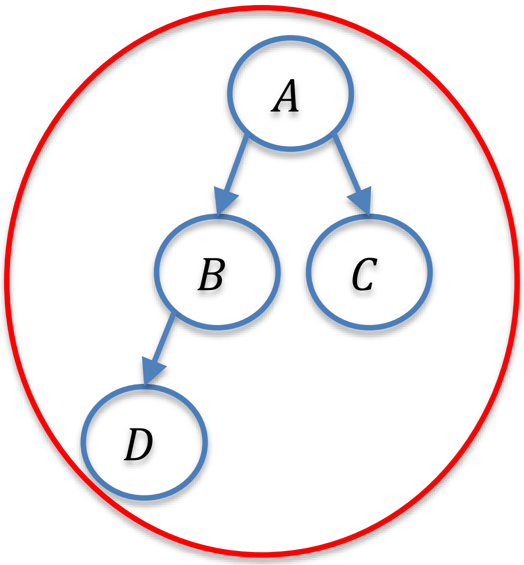


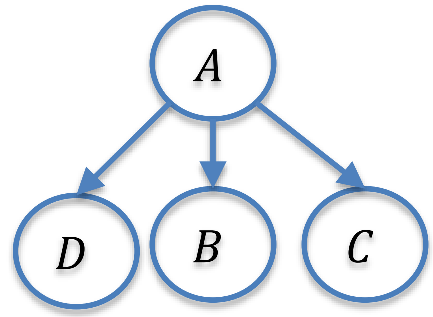


Which one has the highest log-likelihood given the data?

BN Structure Learning

- **Which model should be preferred?**



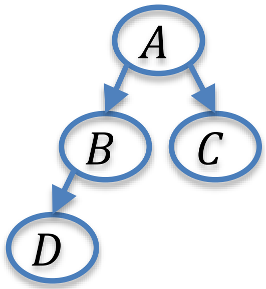


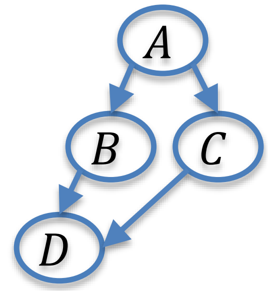
Which one has the highest log-likelihood given the data?

BN Structure Learning

- Determining the structure that maximizes the log-likelihood is not too difficult
  - A complete DAG always maximizes the log-likelihood
    - Proof: next slide
  - 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

$$\frac{\ell_2 - \ell_1}{M} = \frac{1}{M} \sum_m \log \frac{p'(x_D^m | x_B^m, x_C^m)}{p'(x_D^m | x_B^m)}$$

$$= \sum_x p'(x_B, x_C, x_D) \log \frac{p'(x_D | x_B, x_C)}{p'(x_D | x_B)}$$

$$= \sum_x p'(x_B, x_C, x_D) \log \frac{p'(x_B, x_C, x_D)}{p'(x_C | x_B) p'(x_D | x_B) p'(x_B)}$$

$$= d\big(p'(x_B, x_C, x_D) || p'(x_C | x_B) p'(x_D | x_B) p'(x_B)\big) \geq 0$$

Approach 1: Chow-Liu Trees

- Suppose that we want to find the best tree-structured BN that represents a given joint probability distribution
  - Minimize the KL-divergence between the true distribution and the one given by the BN
- First, let's consider the infinite data limit
  - We want to find the directed tree T that minimizes

$$d\left(p(x) || \prod_i p\left(x_i | x_{parent(i \in T)}\right)\right) = ?$$

Approach 1: Chow-Liu Trees

$$d\left(p(x)||\prod_i p(x_i|x_{parent(i\in T)})\right) = -H(p) + \sum_i H(p_i) - \sum_{(i,j)\in T} I(x_i; x_j)$$

- $I(x_i; x_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$ is called the mutual information
  - Measures the dependence between two random variables
- Minimizing the KL-divergence over all directed trees is then equivalent to finding

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

Approach 1: Chow-Liu Trees

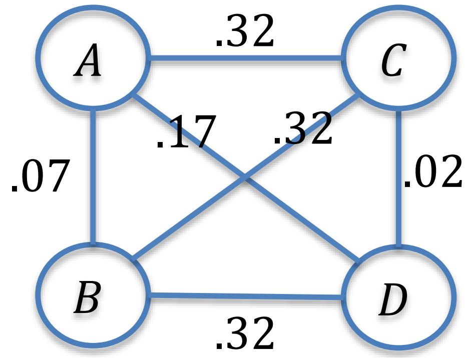
$$\max_T \sum_{(i,j)\in T} I(x_i; x_j)$$

- 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

Approach 1: Chow-Liu Trees

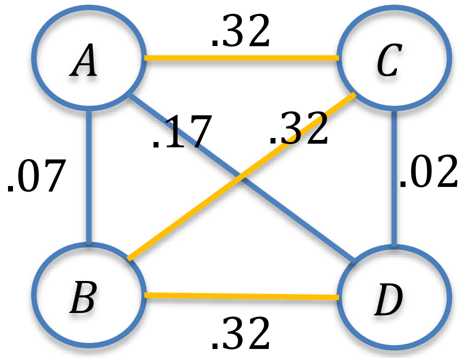
- To use this technique for learning, we simply compute the mutual information for each edge using the empirical probability distributions and then find the max-weight spanning tree
  - Why does this maximize the log-likelihood?
- As a result, we can learn tree-structured BNs in polynomial time
  - Can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

Chow-Liu Trees: Example



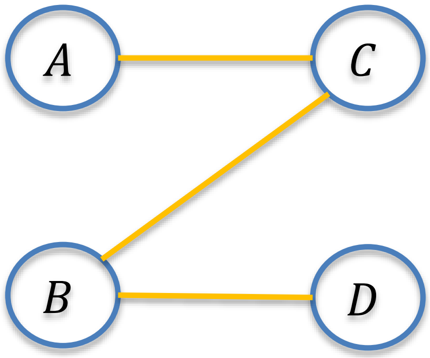
- Edge weights correspond to empirical mutual information for the earlier samples

Chow-Liu Trees: Example



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

Chow-Liu Trees: Example



- Any directed tree 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

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

- $\eta(M)$ is only a function of the number of data points
  - $\eta(M) = constant$ called the Akaike information criterion
  - $\eta(m) = \frac{\log(M)}{2}$ called the Bayesian information criterion

- $Dim(G)$ is the number of parameters needed to represent G