CMSC828G Principles of Data Mining Lecture #12

- Readings:
  - handouts
- Today's Lecture:
  - d-Separation, minimal I-Maps
  - Bayesian Networks
  - Markov Networks
- Upcoming Due Dates:
  - H2 due today
  - P2 due 3/14

Summary of Last Class

We defined the following concepts

- The Markov Independences of a DAG G
  - I (X_i , NonDesc(X_i) | Pa_i)
- G is an I-Map of a distribution P
  - If P satisfies the Markov independencies implied by G

We proved the factorization theorem

- if G is an I-Map of P, then

$$P(X_1,\ldots,X_n) = \prod_i P(X_i \mid Pa_i)$$

- slides courtesy of Nir Friedman, see references

Conditional Independencies

- Let *Markov(G)* be the set of Markov Independencies implied by *G*
- The factorization theorem shows

$$G \text{ is an I-Map of } P \Rightarrow \quad P(X_1,\ldots,X_n) = \prod_i P(X_i \mid Pa_i)$$

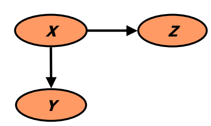
- We can also show the opposite:

Thm:

$$P(X_1,\ldots,X_n) = \prod_i P(X_i \mid Pa_i) \Rightarrow G \text{ is an I-Map of } P$$

Proof (Outline)

Example:



$$P(Z \mid X,Y) = \frac{P(X,Y,Z)}{P(X,Y)} = \frac{\cancel{P(X)}\cancel{P(Y \mid X)}P(Z \mid X)}{\cancel{P(X)}\cancel{P(Y \mid X)}}$$

$$= P(Z \mid X)$$

Implied Independencies

- Does a graph *G* imply additional independencies as a consequence of *Markov(G)*?
- We can define a **logic** of independence statements
- Some axioms:
  - $I(X ; Y \mid Z) \Rightarrow I(Y; X \mid Z)$
  - $I(X ; Y_1, Y_2 \mid Z) \Rightarrow I(X; Y_1 \mid Z)$

d-seperation

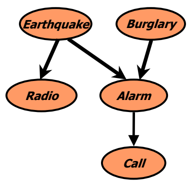
- A procedure *d-sep(X; Y | Z, G)* that given a DAG *G*, and sets *X, Y,* and *Z* returns either *yes* or *no*
- **Goal**:

  d-sep(X; Y | Z, G) = yes iff *I(X;Y|Z)* follows from *Markov(G)*

Paths

- **Intuition:** dependency must “flow” along paths in the graph
- A path is a sequence of neighboring variables

Examples:
- $R \leftarrow E \rightarrow A \leftarrow B$
- $C \leftarrow A \leftarrow E \rightarrow R$



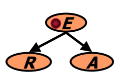
Paths

- We want to know when a path is
  - **active** -- creates dependency between end nodes
  - **blocked** -- cannot create dependency end nodes
- We want to classify situations in which paths are active.

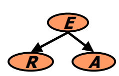
Path Blockage

Three cases:
- Common cause
-
-

Blocked



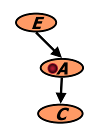
Active



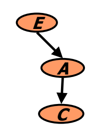
Path Blockage

Three cases:
- Common cause
- Intermediate cause
-

Blocked



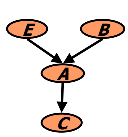
Active



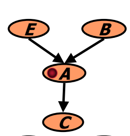
Path Blockage

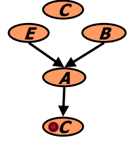
Three cases:
- Common cause
- Intermediate cause
- Common Effect

Blocked



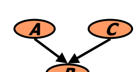
Active





Path Blockage -- General Case

A path is active, given evidence ***Z***, if
- Whenever we have the configuration



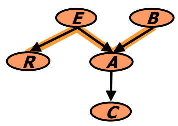
B or one of its descendents are in ***Z***

- No other nodes in the path are in ***Z***

A path is blocked, given evidence ***Z***, if it is not active.

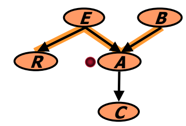
Example

- *d-sep(R,B)?*



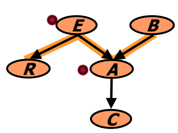
Example

- *d-sep(R,B) = yes*
- *d-sep(R,B|A)?*



Example

- *d-sep(R,B) = yes*
- *d-sep(R,B|A) = no*
- *d-sep(R,B|E,A)?*



d-Separation

- ***X*** is **d-separated** from ***Y***, given ***Z***, if all paths from a node in ***X*** to a node in ***Y*** are blocked, given ***Z***.
- Checking d-separation can be done efficiently (linear time in number of edges)
  - Bottom-up phase:
    Mark all nodes whose descendents are in ***Z***
  - ***X*** to ***Y*** phase:
    Traverse (BFS) all edges on paths from ***X*** to ***Y*** and check if they are blocked

Soundness

Thm:

- If
  - *G* is an I-Map of *P*
  - *d-sep(X; Y | Z, G) = yes*
- then
  - *P* satisfies *I(X; Y | Z)*

Informally,

- Any independence reported by d-separation is satisfied by underlying distribution

Completeness

Thm:

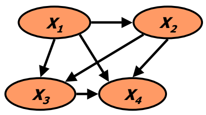
- If *d-sep(X; Y | Z, G) = no*
- then there is a distribution *P* such that
  - *G* is an I-Map of *P*
  - *P* does not satisfy *I(X; Y | Z)*

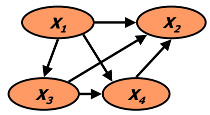
Informally,

- Any independence not reported by d-separation might be violated by the underlying distribution
- We cannot determine this by examining the graph structure alone

I-Maps revisited

- The fact that *G* is I-Map of *P* might not be that useful
- For example, **complete** DAGs
  - A DAG is *G* is complete is we cannot add an arc without creating a cycle





- These DAGs do not imply any independencies
- Thus, they are I-Maps of any distribution

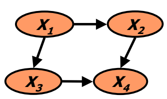
Minimal I-Maps

A DAG *G* is a **minimal I-Map** of *P* if

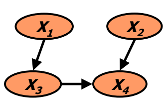
- *G* is an I-Map of *P*
- If $G' \subset G$, then G' is not an I-Map of *P*

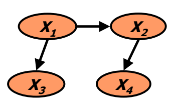
Removing any arc from *G* introduces (conditional) independencies that do not hold in *P*

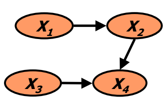
Minimal I-Map Example

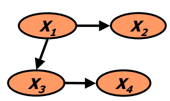
- If  is a minimal I-Map

- Then, these are **not** I-Maps:









Constructing minimal I-Maps

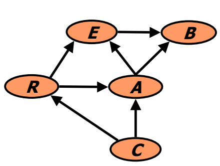
The factorization theorem suggests an algorithm

- Fix an ordering $X_1,\dots,X_n$
- For each *i*,
  - select Pa_i to be a minimal subset of $\{X_1,\dots,X_{i-1}\}$, such that $I(X_i\,;\,\{X_1,\dots,X_{i-1}\} - Pa_i \mid Pa_i)$
- Clearly, the resulting graph is a minimal I-Map.

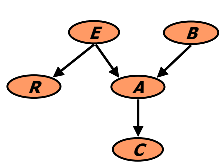
Non-uniqueness of minimal I-Map

- Unfortunately, there may be several minimal I-Maps for the same distribution
  - Applying I-Map construction procedure with different orders can lead to different structures

Order: *C, R, A, E, B*



Original I-Map



Choosing Ordering & Causality

- The choice of order can have drastic impact on the complexity of minimal I-Map
- Heuristic argument:
  construct I-Map using **causal** ordering among variables
- Justification?
  - It is often reasonable to assume that graphs of causal influence should satisfy the Markov properties.
  - We will revisit this issue in future classes

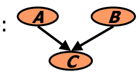
P-Maps

- A DAG G is P-Map (**perfect map**) of a distribution P if
  - $I(X; Y \mid Z)$ if and only if $d\text{-}sep(X; Y \mid Z, G) = yes$

Notes:

- A P-Map captures all the independencies in the distribution
- P-Maps are unique, up to DAG equivalence

P-Maps

- Unfortunately, some distributions do not have a P-Map
- Example: $P(A,B,C) = \begin{cases} 1/12 & \text{if } A \oplus B \oplus C = 0 \\ 1/6 & \text{if } A \oplus B \oplus C = 1 \end{cases}$
- A minimal I-Map: 

- This is not a P-Map since $I(A;C)$ but $d\text{-}sep(A;C) = no$

Bayesian Networks

- A Bayesian network specifies a probability distribution via two components:
  - A DAG G
  - A collection of conditional probability distributions $P(X_i|Pa_i)$
- The joint distribution P is defined by the factorization

$$P(X_1,\ldots,X_n) = \prod_i P(X_i \mid Pa_i)$$

- Additional requirement: G is a minimal I-Map of P

Summary

- We explored DAGs as a representation of conditional independencies:
  - Markov independencies of a DAG
  - Tight correspondence between *Markov(G)* and the factorization defined by G
  - d-separation, a sound & complete procedure for computing the consequences of the independencies
  - Notion of minimal I-Map
  - P-Maps
- This theory is the basis for defining Bayesian networks

References

- *Principles of Data Mining*, Hand, Mannila, Smyth. MIT Press, 2001.
- Nir Friedman's excellent lecture notes, http://www.cs.huji.ac.il/~pmai/