Lecture 5: d-separation. Bayes nets in practice

- Bayes ball revisited
- d-separation
- Constructing Bayes nets

Recall from last time

- A Bayesian network is a DAG G over variables $X_1, \ldots, X_n$, together with a distribution P that factorizes over G. P is specified as the set of conditional probability distributions (local probability models) associated with G's nodes.
- G is an *I-map (independence map)* for P. I.e., for any node X_i, we have:

$$X_i \perp\!\!\!\perp \text{Nondescendents}(X_i) | \text{Parents}(X_i))$$

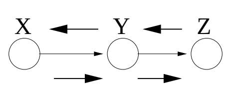
- What other independencies can be "read off" G?

Recall: Bayes ball algorithm

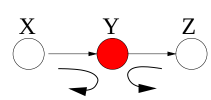
- Suppose we want to decide whether $X \perp\!\!\!\perp Z | Y$ for a general Bayes net with corresponding graph G.
- We shade all nodes in the evidence set, Y
- We put balls in all the nodes in X, and we let them bounce around the graph according to rules inspired by these three base cases
- Note that the balls can go in any direction along an edge!
- If any ball reaches any node in Z, then the conditional independence assertion is *not* true.

Base rules

- *Head-to-tail*

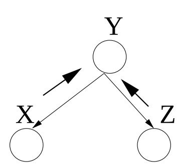


Y unknown, path unblocked

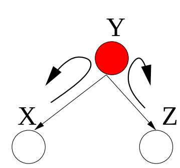


Y known, path blocked

- *Tail-to-tail*

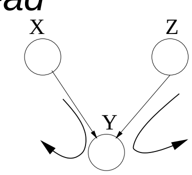


Y unknown, path unblocked

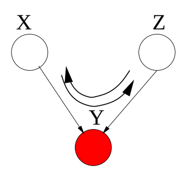


Y known, path blocked

- *Head-to-head*

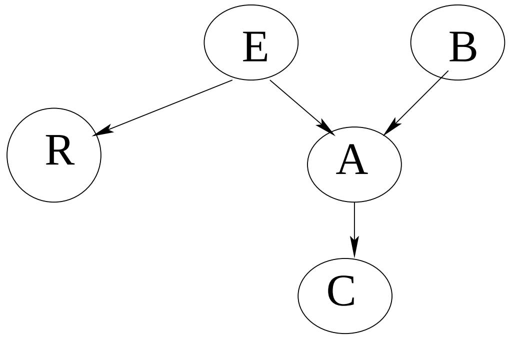


Y unknown, path BLOCKED



Y known, path UNBLOCKED

Example: The alarm network



Is $R \perp\!\!\!\perp C | A$?

d-separation

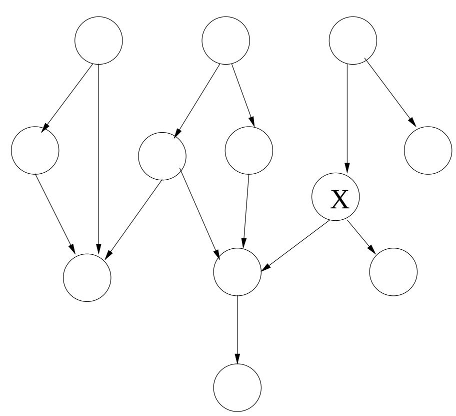
- Suppose we want to show that a conditional independence relation, $X \perp\!\!\!\perp Z | Y$, is implied by a DAG G in which X, Y, Z are non-intersecting sets of nodes.
- A path is said to be **blocked** if it includes a node such that:
  1. the arrows in the path do *not* meet head-to-head at the node, and the node is in the conditioning set Y (this covers the head-to-tail and tail-to-tail cases)
  2. the arrows do meet head-to-head and neither the node nor its descendents are in Y
- If, given the set of conditioning nodes Y, all paths from any node in X to any node in Z are blocked, then X is **d-separated** from Z given Y

Important results

- "Soundness": If a joint distribution P factorizes according to a DAG G, and if X, Y and Z are subsets of nodes such that Y d-separates X and Z in G, then P satisfies $X \perp\!\!\!\perp Z | Y$.
- "Completeness": if Y does **not** d-separate X and Z in DAG G, then there exists at least one distribution P which factorizes over G and in which $X \not\perp\!\!\!\perp Z | Y$

Isolating a node

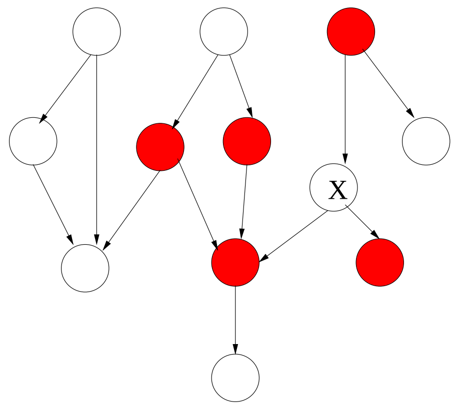
Suppose we want the smallest set of nodes U such that X is independent of all other nodes in the network given U: $X \perp\!\!\!\perp (\{X_1 \ldots X_n\} - \{X\} - U) \, | U$. What should U be?



Markov blanket

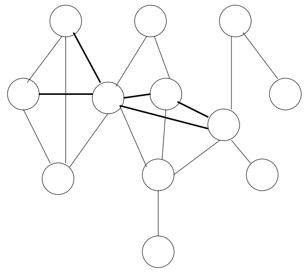
- Clearly, at least X's parents and children should be in U
- But this is not enough if there are v-structures; U sill also have to include X's "spouses" - i.e. the other parents of X's children

The set U consisting of X's parents, children and other parents of his children is called the **Markov blanket** of X.



Moral graphs

Given a DAG G, we define the **moral graph of G** to be an undirected graph U over the same set of vertices, such that the edge (X, Y) is in U if X is in Y's Markov blanket



- If G is an I-map of P, then U will also be an I-map of P
- But many independencies are lost when going to a moral graph
- Moral graphs will prove to be useful when we talk about inference.

Perfect maps

A DAG G is a **perfect map** of a distribution P if it satisfies the following property:

$$X \perp\!\!\!\perp Z | Y \Leftrightarrow Y \text{ d-separates } X \text{ and } Z$$

- A perfect map captures all the independencies of a distribution
- Perfect maps are unique, up to DAG equivalence
- How can we construct a perfect map for a distribution?

Some distributions do not have perfect maps!

Example: We have two independent unbiased coins that we toss. If both coins come up the same, a bell rings with probability 2/3.

Here, there are three minimal I-maps (which?) but none is a perfect map.

Constructing Bayes nets in practice

Usually, we do not construct Bayes nets based on knowledge of the joint probability distribution P. We have some vague idea of the dependencies in the world, and we need to make that precise in a Bayes net. This involves several steps:

- Formulating the problem
- Choosing random variables
- Choosing independence relations
- Assigning probabilities in the CPDs

Choosing random variables

- Variables must be precise. What are the values, how are they defined, and how are they measured?
  E.g. *Weather* - what values will it take? When do we assign the *bitter-cold* value?
- If the variables are continuous and we discretize them, a coarse discretization may introduce additional dependencies.
- There several kinds of variables:
  - Observable
  - Sometimes observable (e.g. medical tests)
  - Hidden - these may or may not be useful to include, depending on the other independencies that they generate

Choosing the structure

- Causal connections tend to make the graphs sparser. Note that we must judge causality *in the world!*
- In general, these models are approximate. There is a trade-off between precision and the size and sparsity of the graph.
  E.g., see the alarm network

Choosing numbers for the CPDs

- Conditional probabilities could come from a few sources:
  - An expert
    - People hate picking numbers!
    - Having a good network structure usually makes it easier to elicit numbers from people too.
  - An approximate analysis (e.g. in card games)
  - Guessing
  - Learning
- Bad news: In all these cases, the numbers are approximate!
- Good news: the numbers usually do not matter all that much.
- Sensitivity analysis can help in deciding whether certain numbers are critical or not for the conclusions

Important factors when choosing probabilities

- Avoid assigning zero probability to any events!
- The relative values (or ordering) of conditional probabilities for $P(X|\text{Parents}(X))$, given different values of Parents(X) is important
- Having probabilities that are orders of magnitude different can cause problems in the network

Example: Pathfinder (Heckerman, 1991)

- Medical diagnostic system for lymph node diseases
- Large net! 60 diseases, 100 symptoms and test results, 14000 probabilities
- Network built by medical experts
  - 8 hours to determine the variables
  - 35 hours for network topology
  - 40 hours for probability table values
- Experts found it easy to invent causal links and probabilities
- Pathfinder is now **outperforming world experts** in diagnosis
- Commercialized by Intellipath and Chapman Hall Publishing; extended to other medical domains