Introduction

Causal discovery: the problem of finding the causal graph using data

(this lecture: extension of Section 2.5 in the book)

Markov properties

We have related probability distributions to graphs in two ways:

- The joint distribution coming from an SCM must satisfy the *product decomposition rule*
- If two variables are d-separated in the SCM's graph given a set S, they must be independent given S in the distribution coming from that SCM ("global Markov property")

(More such "Markov properties" exist; they are equivalent under general conditions [see Lauritzen 1996: Graphical Models])

If a distribution and a graph are related in this way, we say they are Markov to each other

Faithfulness (1/2)

Recall that the Markov properties only say something in one direction:

- A d-separation implies a (conditional) independence
- A d-connection means a 'likely' dependence, but it might still be an independence

So a *complete* graph (in which all pairs of variables have an arrow between them) could explain any data set... but this is not the answer we want from a causal discovery method!

Faithfullness (2/2)

If a distribution has a real dependence wherever there is a d-connection in the graph, we say the distribution and the graph are faithful to each other

Suppose a graph exists that is Markov and faithful to the distribution of our data

- (This is a very strong assumption! We will weaken it later)

Then our goal would be: find such a graph

Markov equivalence

If two graphs have exactly the same d-separations, we can not choose between them by looking at conditional independences in data. The best we can do is find the *equivalence class* of graphs Markov and faithful to the data

Two graphs are called Markov equivalent if they have exactly the same d-separations

Theorem [Verma and Pearl, 1990]

Two directed acyclic graphs are Markov equivalent if and only if

- they have the same *skeleton* (= the graph when ignoring the directions of the arrows)
- they have the same *v-structures* (= patterns $X \to Y \leftarrow Z$ with X and Z not adjacent)

Study question 2.5.1 a–c

PC algorithm: overview

The PC algorithm consists of four main steps:

1. **Initialization**: Let G be the complete undirected graph
2. **Skeleton search**: Do independence tests to remove edges from G
3. **Orient v-structures**: Replace undirected edges by directed ones to form required v-structures
4. **Orientation rules**: Reason about the directions of the remaining edges

In a graph on n nodes, there are $\frac{1}{2}n(n-1)2^{n-2}$ possible (conditional) independences that could be tested: too many!

- In particular, avoid testing independences with large conditioning sets: such tests are unreliable

PC algorithm: initialization & skeleton search

Let G be the complete undirected graph
for $k = 0, 1, 2, \ldots$ **do**
    for all ordered pairs $X \in V$, $Y \in \mathrm{Adj}(X)$ **do**
        for all sets $S \subseteq \mathrm{Adj}(X) \setminus \{Y\}$ with $|S| = k$ **do**
            Test independence of X and Y given S in the data
            if they are independent **then**
                Remove edge $X — Y$ from G
                Store S in $\mathrm{SepSet}(\{X, Y\})$
                break
            end if
        end for
    end for
    if $|\mathrm{Adj}(X) \setminus \{Y\}| < k + 1$ for all $X \in V$, $Y \in \mathrm{Adj}(X)$ **then**
        break
    end if
end for

PC algorithm: Orient v-structures

Next, make sure the graph has all the v-structures it should have

Don't do new independence tests: use the results we already saw

for all unordered pairs $\{X, Y\}$ not adjacent in G, and all Z adjacent to both X and Y **do**
  if $Z \notin \mathrm{SepSet}(\{X, Y\})$ **then**
    Orient the two edges as $X \to Z$, $Y \to Z$
  end if
end for

PC algorithm: Orientation rules

In the final step of the algorithm, orient any edges G that were not part of a v-structure, but for which we do know the direction

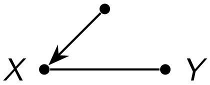
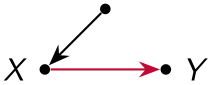
- using that we don't want extra v-structures, and that the graph may not have directed cycles

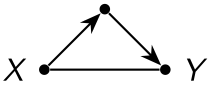
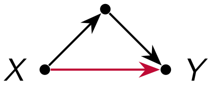
It was shown that this can be accomplished by applying three *orientation rules*: whenever a set of nodes matches a rule's pattern, an edge is oriented

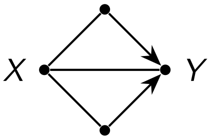
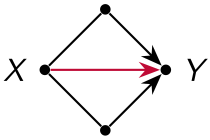
repeat

    Try applying each of the three orientation rules below

until no more rules could be applied

Rule 1:  becomes 

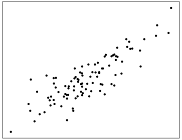
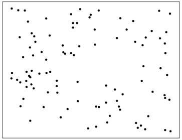
Rule 2:  becomes 

Rule 3:  becomes 

(A missing edge in a pattern means that edge must not be in G)

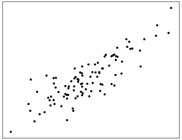
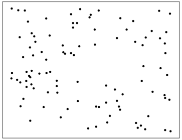
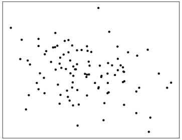
Weak signals

To discover the right causal graph, PC looks for conditional independencies in the data. But these are not always clear:

-  $\Rightarrow$ nodes must be d-connected
-  $\Rightarrow$ nodes must be d-separated

Weak signals

To discover the right causal graph, PC looks for conditional independencies in the data. But these are not always clear:

-  $\Rightarrow$ nodes must be d-connected
-  $\Rightarrow$ nodes must be d-separated
-  $\Rightarrow$???

Constraint-based vs. score-based approaches

To deal with these weak signals, instead of trying to draw a binary conclusion from each, it would be better to compare the strengths of all signals and find the graph that agrees best

Score-based methods do so by assigning an overall score to each graph, then finding the graph with the best score

The score should reflect:

- How well the data can be explained by the graph (i.e. if data has a dependence where graph has a d-separation, score should be bad)
- Occam's razor: explanation with fewer edges is better

Then we can also deal with data that is not faithful to any graph

Latent confounders

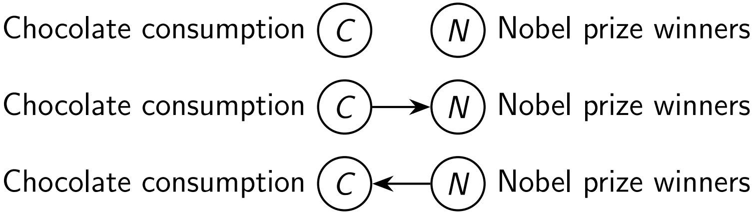
If the only graphs we are considering are directed acyclic graphs on the observed variables, then we end up trying to explain every dependence we see by causal relations between those variables

Important alternative explanation: a latent ($=$ unobserved) variable that is a cause of two observed variables

The assumption that there are no latent confounders is called causal sufficiency

Latent confouders

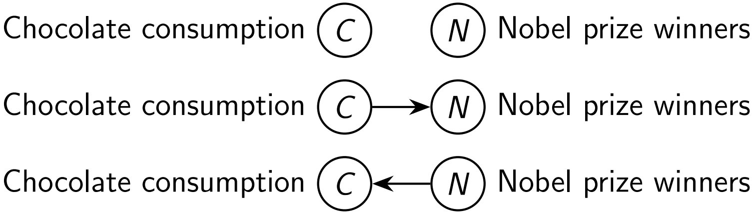
What is the right causal graph for chocolate consumption vs. Nobel prize winners?



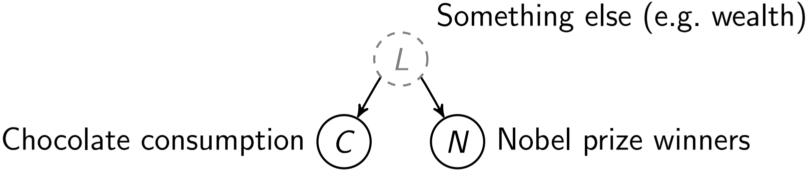
All seem unlikely!

Latent confouders

What is the right causal graph for chocolate consumption vs. Nobel prize winners?



All seem unlikely!
Fourth possibility:



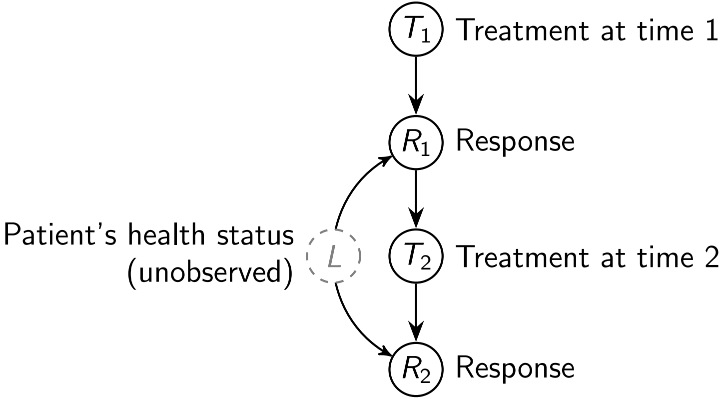
FCI

An causal discovery algorithm that deals with latent confounders is FCI (fast causal inference)

- Constraint-based like PC (in fact, works quite similarly to PC)
- Output is a so-called MAG: a graph with various additional types of edges; has the strange property that a directed arrow does not always mean a direct causal relation

Latent confouders and non-independence constraints

With latent confounders, conditional independence is no longer enough!



Current research

More algorithms have been developed that I haven't mentioned, but there is plenty of room for improvement here!

We want:

- Score-based algorithms that can properly compare the strength of conflicting pieces of evidence
- . . . possibly including evidence from non-independence constraints
- . . . and can be guaranteed to find the graph (or equivalence class of graphs) that really optimize the score
- . . . while still being computationally fast