Learning Causal Structure from Observational Data

Lizzie Silver

University of Melbourne

MLAI Meetup
September 18, 2018

Last month, Ross Gayler said his talk was going to be an Icelandic Saga. This is gonna be more like a wikipedia article. I'm going to try to give a high-level overview of this topic that really deserves a deeper treatment, with variable quality. And none of this is my own work.

Fig 1: Prepare to be spooked

One ground rule: Because the talk is about causation, there's probably at least one person here who's just waiting for a moment to say, "But like ... correlation isn't causation, so isn't this all impossible?" If that's you, I just want to ask you to hold that question and see whether you still have it at the end of the talk. I am at least as worried as you are about validity, and I'm going to try to give you more precise things to be skeptical about.

Why learn causal relationships:

- Predict results of *interventions*

Machine learning and AI are really good at predicting, and you don't need to know causation in order to predict. So why study causation? There are three areas where we want to know about causal relationships:

- First, when we want to choose an intervention, we need to predict the result of that intervention - like choosing a medical treatment. But out past data doesn't tell us what will happen if we change the situation.
- Second, when the environment we're studying will change over time, and we want to know what relationships will stay predictive in a range of different settings.

Why learn causal relationships:

- Predict results of *interventions*
- Find relationships that are *robust* to changes in the distribution

Machine learning and AI are really good at predicting, and you don't need to know causation in order to predict. So why study causation? There are three areas where we want to know about causal relationships:

- First, when we want to choose an intervention, we need to predict the result of that intervention - like choosing a medical treatment. But out past data doesn't tell us what will happen if we change the situation.
- Second, when the environment we're studying will change over time, and we want to know what relationships will stay predictive in a range of different settings.
- Third, we don't just care about prediction. Sometimes we have a scientific kind of interest, we don't just want to know what will happen – we want to know why and how. Ideally we want to learn the mechanisms involved.

Now I'll start by showing you what the inputs and outputs of a causal structure learning algorithm look like.

Why learn causal relationships:

- Predict results of *interventions*
- Find relationships that are *robust* to changes in the distribution
- Learn about the *mechanisms* that produce the data

Machine learning and AI are really good at predicting, and you don't need to know causation in order to predict. So why study causation? There are three areas where we want to know about causal relationships:

- First, when we want to choose an intervention, we need to predict the result of that intervention - like choosing a medical treatment. But out past data doesn't tell us what will happen if we change the situation.
- Second, when the environment we're studying will change over time, and we want to know what relationships will stay predictive in a range of different settings.
- Third, we don't just care about prediction. Sometimes we have a scientific kind of interest, we don't just want to know what will happen – we want to know why and how. Ideally we want to learn the mechanisms involved.

Now I'll start by showing you what the inputs and outputs of a causal structure learning algorithm look like.

Inputs to causal structure learning algorithms

ID	Smoking	Yellow teeth	Asbestos	Lung cancer
1	Yes	Yes	No	No
2	No	No	Yes	No
⋮	⋮	⋮	⋮	⋮

Now I'll start by showing you what the inputs and outputs of a causal structure learning algorithm look like.

The input is just tabular data like this example of a public health dataset, with:

- Random variables in the columns
- Observations of cases in the rows

So the input is simple.

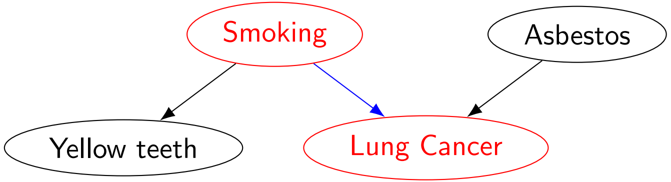
But the *output* of causal models is going to be new to most of you. We need a way to represent causal information, statements like "smoking causes cancer", and it has to fit with our complicated ideas about how causation works. The representation is really important, it's new material, and the rest of the talk will build on it.

Causal Structural Equation Models

$$\begin{aligned}
\mathit{Smoking} &:= f_1(\epsilon_{smoking}) \\
\mathit{Asbestos} &:= f_2(\epsilon_{asbestos}) \\
\mathit{Yellow\ Teeth} &:= f_3(\mathit{Smoking}, \epsilon_{teeth}) \\
\mathit{Lung\ Cancer} &:= f_4(\mathit{Smoking}, \mathit{Asbestos}, \epsilon_{cancer})
\end{aligned}$$

- A natural way to represent the causal model is using structural equations. Each variable is a function of its causes, plus its own error term. This captures the idea that causation can be stochastic.
- I've written the assignment sign rather than the equals sign, because we want to capture that this equality isn't accidental – these equations represent how those variables were generated. In the literature you'll often see this as an equals sign but it should be read as assignment.
- From here on in the talk I'm going to use graphs as a shorthand to refer to these structural equation models.
- (Structural equation models have a long history; in some sciences, like economics and psychology, the equations are often assumed to be linear, because linear models are so much easier to deal with. I've represented them as general functions to make clear that we're not assuming any specific functional form.)

Causal Graphical Models: $G = \{\mathbf{V}, \mathbf{E}\}$



Nodes represent random variables

Edges represent causal relationships

DAG: ‘Directed Acyclic Graph’

The output – what we’re ideally trying to learn – is a directed graph with a causal interpretation. The graph has a set of vertices, also called nodes, and a set of edges, the arrows.

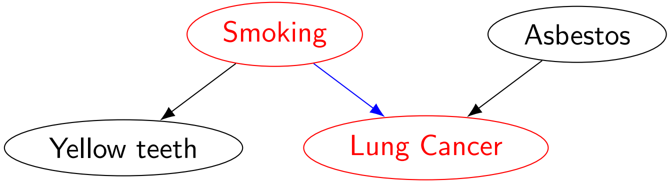
Each node represents one of our random variables. They come straight from the column names in our data table.

DAG - Directed Acyclic Graph. By the way, directed graph terminology often uses kinship/ancestral terms; so here we’d say Smoking is a “parent” of Lung Cancer, or Lung Cancer is a “child” of Smoking.

Relation to the structural equation model: if a variable A is one of the arguments in the SEM for variable B, then in the graph, there’s an edge from A to B.

The edges are what we’re trying to learn from our data. The edges represent causal relationships, which we interpret in terms of interventions.

Causal Graphical Models: $G = \{\mathbf{V}, \mathbf{E}\}$



$$P(Lung\ Cancer | do(Smoking = \text{“Yes”}), do(\mathbf{V} = \mathbf{v}))$$
$$\neq$$
$$P(Lung\ Cancer | do(Smoking = \text{“No”}), do(\mathbf{V} = \mathbf{v}))$$

for some setting of the other variables $\mathbf{V} = \{Asbestos,\ Yellow\ Teeth\}$, where $do(\mathbf{V} = \mathbf{v})$ means you intervene to set $\mathbf{V}$ to the value $\mathbf{v}$.

In our example network, the edge from Smoking to Lung Cancer means that if you intervene to set the value of Smoking to "yes", the probability of Lung Cancer will be different than if you'd set the value to "no" ... given some setting of the other variables. Things to notice:

1. Causal dependence is not probabilistic dependence! For example, if I **observe** that someone's teeth are very yellow, it's more likely that they have lung cancer, compared to someone with white teeth. That's a probabilistic dependence. But if I **intervene** to paint their teeth yellow, I don't change the probability that they get lung cancer. There's no causal dependence. This is the old “correlation isn't causation” example.
2. Correlation is symmetric, but causation is asymmetric. If you make someone smoke, their teeth are more likely to turn yellow. But if you paint their teeth yellow, they're not more likely to smoke.
3. Causation is not deterministic, it's probabilistic. This fits with the intuition that smoking causes cancer, even though we know that not everyone who smokes, gets lung cancer.

'Direct' cause vs. Ancestor

Causal Graph $G = \{\mathbf{V}, \mathbf{E}\}$
Each edge $X \rightarrow Y$ represents a direct causal claim:

X is a direct cause of Y relative to $\mathbf{V}$

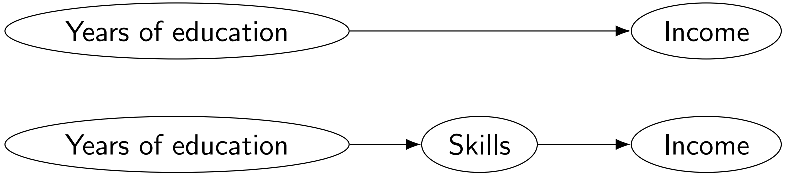


Figure: Example: Mediating variables can be omitted

- Note: The "true" set of edges is relative to the set of variables!

- If my graph just includes Education and Income, there's an edge from education to income. But if I now include the variable 'skills', perhaps the entire effect of education on income goes through skills – so for example, if you get a PhD in philosophy, maybe that doesn't increase your skills so it has no effect on income. In the new model, there's no direct edge from education to income.

Modeling interventions & confounders

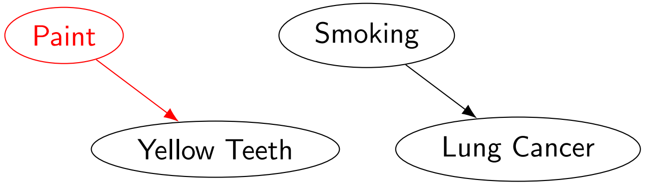


Figure: Example: A “surgical” intervention

- Interventions add a node & an edge to the model
- ‘Hard’ interventions break all other edges into the child of the intervention node
- Requirements: (1) Unambiguous, (2) ‘No Fat Hand’

Graphical models let us represent interventions. To model an intervention, you add a new node and an edge to the model - the intervention variable, and the variable it acts on.

If, say, I flip a coin and decide to paint people’s teeth either yellow or white, that would be a “hard” or surgical intervention - so called because it breaks the other edges into yellow teeth. Now smoking has no effect on tooth colour.

This also lets us represent confounders, as common causes of the variables we’re interested in. In this case Smoking is a confounder of the relationship between yellow teeth and lung cancer. It explains why randomised controlled trials break the influence of confounders, because the hard interventions break edges in the graphs.

I might instead do a ‘soft’ intervention where I influence the value without breaking the influence of other causes. This is common – for example, economic experiments on income can give people money, but they can’t take away the money people already have, at least not with ethics approval. The intervention variable itself is still exogenous so we can still measure its impact without worrying about confounders.

‘Intervention’ is taken as a basic concept. This formalism doesn’t represent exactly how the intervention works; that’s left implicit.

Output may include some uncertainty

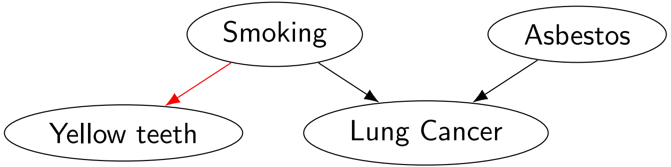


Figure: A DAG over our four variables

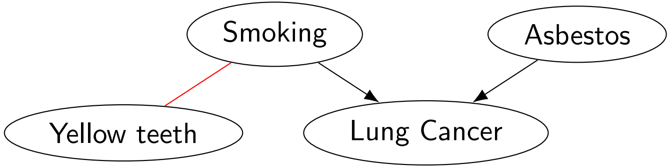


Figure: Pattern representing the Markov equivalence class of the DAG above

Here's something I'll mention now and cover in a bit more detail in a few minutes. The output of the algorithms may include some uncertainty. If the true graph is the model above, and you were to feed data from that model to most mainstream causal learning algorithms, you would only be able to learn the direction of two of the three edges. That's still helpful – we've still ruled out a lot of models – and we've accurately represented the remaining uncertainty. That's all we can ask for.

Learning structure vs. Fitting a structure to data

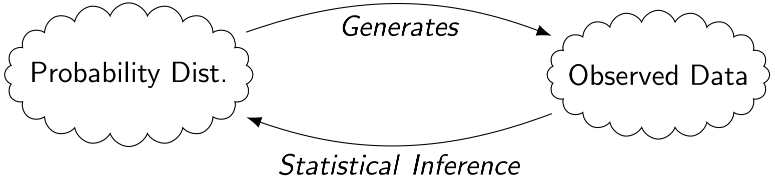
Steps in causal learning:

1. First, learn the set of edges
2. Second, fit each structural equation
3. ???
4. Profit

I'll only cover step 1.

- Distinguish between learning the structure of the graph, and estimating the model
- once you have the structure, there are many ways to estimate - with difficulties if you can't measure all the variables of course. But this is the kind of statistical estimation you are all familiar with: start with the correct model and some data, and fit the model to the data.
- I'm going to talk about step 1, learning the structure. You can also think of that as avoiding model misspecification.

Probability & statistical inference, vs. Causal generation & causal inference

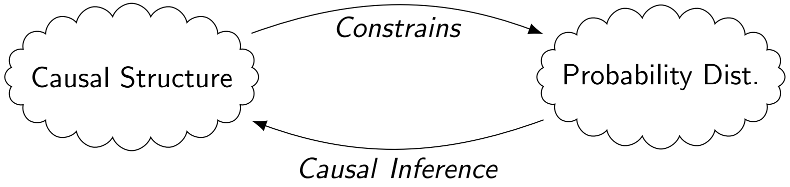


Probability Dist.

Generates

Observed Data

Statistical Inference



Causal Structure

Constrains

Probability Dist.

Causal Inference

High level overview of what we're doing:

- Probability theory lets you deduce how true distributions generate data; statistical inference tells you how to go backwards, and learn about the true distribution from the data

- Axioms about causality tell us how causal structures constrain the probability distributions that could generate the data. The task is now to take the constraints we observe, and use them to infer back to the causal structure.

Constraint types

Constraint Type	Distributions
Conditional Independences	All
Vanishing determinants of partial covariance matrices	Linear Gaussian with unobserved confounders
Unequal dependence on residuals	Non-linear additive noise; or Linear non-Gaussian
⋮	⋮

Other constraints come from the sampling conditions (experimental manipulation, time series, etc.) or we may make assumptions (acyclicity, sufficiency, etc.).

There are lots of different kinds of constraints!

The most general constraints, which occur in all parametric families, are the conditional independence constraints. I'll mostly talk about them.

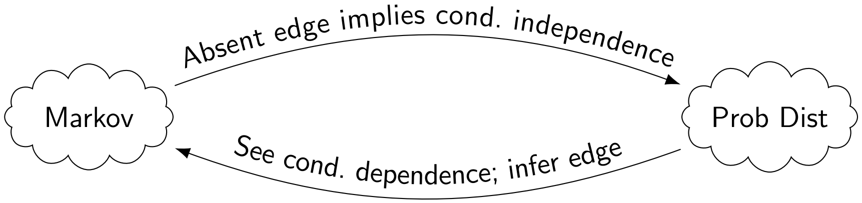
I won't get to cover all the kinds of constraints, and we don't need to, I just want to show you that we are not limited to conditional independences, even though I'll mostly talk about them.

Connect to data: The Markov condition

G is associated with a distribution P_G, from which we sample our data. What do we know about P_G if we know G?

The Local Markov Condition:

> For any variable $X \in \mathbf{V}$, X is independent of its *non-descendants* conditional on its *parents* in G.



Absent edge implies cond. independence

Markov

Prob Dist

See cond. dependence; infer edge

The first condition that gives us constraints is the Causal Markov Condition. It says that if there's no graphical connection between two variables, there's some conditional independence(s) between them in the probability distribution.

What can we then use to infer back to the model? We can take the contrapositive: If two variables are always dependent, over all conditioning sets, there must be an edge between them.

When does Markov fail? Omitted common causes

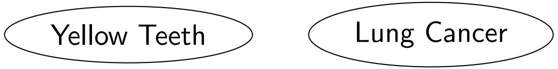


Figure: G represents the lack of causal relationship between *Yellow Teeth* and *Lung Cancer*. But $\mathbf{V}$ is not causally sufficient, so Markov doesn't hold

Causal sufficiency:

For any pair of variables $\{X, Y\} \in \mathbf{V}$, if there exists a variable Z such that Z is a direct cause of both X and Y, then $Z \in \mathbf{V}$

The Markov condition might not hold, if we've ommitted a confounder variable from our model. For example: if we look just at yellow teeth and lung cancer, there's no causal edge between them - neither causes the other. But they are correlated!

If the causal Markov condition held, we could see that correlation and infer an edge. So Markov doesn't hold unless we have 'Causal Sufficiency', which just means that all the confounders are in our model.

Note, just being in the model doesn't mean we've observed the variable. We can include unobserved variables. We won't be able to estimate their effect, but we won't make mistaken inferences about the other causal relationships in the model either.

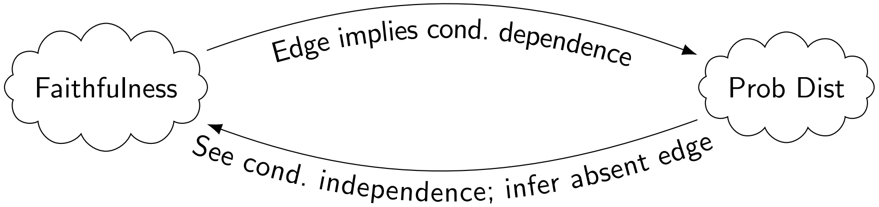
While Markov might not be true of a particular model, in most cases it's possible to find a larger model where Markov is satisfied. We don't usually worry about real causal systems being set up in such a way that Markov fails (except for quantum phenomena, etc.).

Connect to data: Faithfulness

The Markov condition only entails *conditional independences*. What about *conditional dependences*?

Faithfulness:

> The only conditional independences in P_G are those entailed by the Markov condition.



Edge implies cond. dependence
Faithfulness
Prob Dist
See cond. independence; infer absent edge

- Markov lets us add edges to the model. But the complete graph can fit any distribution, so we also need a reason to exclude edges.

- Faithfulness is just the converse of Markov. Faithfulness says: if there's an edge in the graph, those variables will be dependent, for any conditioning set. So again, to infer from the data back to the structure, we take the contrapositive: if you see an independence, you can infer that the two variable are not connected.

When does faithfulness fail? Canceling paths, etc.

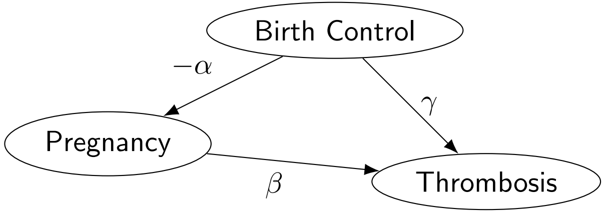


Figure: Example: P_G is unfaithful when $\alpha\beta = \gamma$

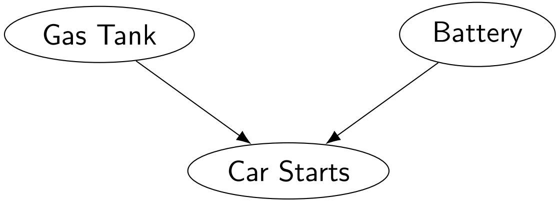
Faithfulness may fail if we have probability distributions that are set up juuust right.

Example of cancelling paths: birth control increases blood clots (thrombosis), but it also decreases pregnancy, and pregnancy increases blood clots. What if those effects exactly cancel out?

Note that in self-regulatory systems, we expect faithfulness to fail. Your thermostat ensures that the indoor temp is independent of the outdoor temp by design! There are lots of examples of this in biology - that's what homeostasis is for.

So unlike the Markov condition, we expect some failures of faithfulness to occur, just because of how real causal systems work.

Distinguishing models: V-structures



$$Gas\ Tank \perp\!\!\!\perp Battery$$

But

$$Gas\ Tank \not\perp\!\!\!\perp Battery \mid Car\ Starts = 0$$

Intuitive example to see how we can orient edges into a collider, a.k.a. V-structure

Emphasize that v-structures are the only structure with this pattern of conditional independences

$A \perp\!\!\!\perp B|C$ means 'A is independent of B given C'. $\not\perp\!\!\!\perp$ means 'dependent'.

What we can learn: the Markov equivalence class

True DAG	$A \rightarrow B \rightarrow C$	$A \rightarrow B \leftarrow C$
Observed CIs	$A \perp\!\!\!\perp C \mid B$	$A \perp\!\!\!\perp C \mid \emptyset$
Set of DAGs in MEC	$A \rightarrow B \rightarrow C$ $A \leftarrow B \leftarrow C$ $A \leftarrow B \rightarrow C$	$A \rightarrow B \leftarrow C$
CPDAG	$A - B - C$	$A \rightarrow B \leftarrow C$

Unfortunately, sometimes a group of different models imply the same conditional independences. We call this set the Markov equivalence class. In this example, each of these graphs entail one conditional independence. The graph on the right is the *only* graph that entails that independence; whereas on the left there are three graphs that entail the same independence as the true graph. We can't learn which of the three graphs is the true one.

Notice that we can learn something: in all three models, A is not adjacent to C. We can represent the whole MEC using a type of graph called a CPDAG ('Complete Partially Directed Acyclic Graph'). If an edge in the CPDAG is directed, that means it is oriented that way in all the DAGs in the equivalence class. If it's undirected, that means it's oriented one way in some DAGs, and the other way in other DAGs in the equivalence class. (Note that a CPDAG can have a combination of directed and undirected edges.) Happily, all DAGs in a MEC have the same adjacencies and v-structures.

Our goal is to learn as much about the structure as we can, and accurately represent the remaining uncertainty. So causal search algorithms return CPDAGs instead of DAGs.

Some publications refer to CPDAGs as 'patterns'.

d-separation: a recipe linking CIs to graph constraints

Graph separation $\Leftrightarrow$ Probabilistic independence

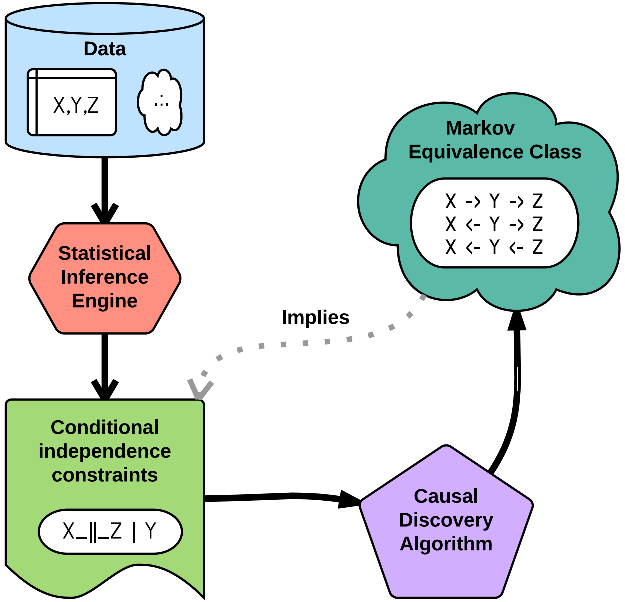
X is *d*-separated from Y conditional on $\mathbf{Z}$ in $G \Leftrightarrow X \perp\!\!\!\perp_{P_G} Y|\mathbf{Z}$ for a causally sufficient graph G and a distribution P_G that is Markov and Faithful to G.

- X is *d*-separated from Y conditional on $\mathbf{Z}$ if every path between X and Y is blocked.
- A path p is blocked if any node on it is *inactive*, and unblocked otherwise.
- A node W on p is inactive iff:
  - W is a non-collider on p, and $W \in \mathbf{Z}$, or
  - W is a collider on p, and $W \notin \mathbf{Z}$, and for any V that is a descendant of W, $V \notin \mathbf{Z}$.

I've given a couple of intuitive examples, but I want to be clear that there is an exact recipe for figuring out what independences are entailed by a graph. *d*-separation gives a complete list of them. I don't really want to go through the exact definition of d-sep, but I want to let you know that it's there, it's not tooo complicated, and it's what powers the constraint-based algorithms.

d-separation lets you translate from graph structure to independence constraints, and vice versa (under the assumptions of Markov and Faithfulness).

Constraint-based search


Data
X,Y,Z
.:.
Statistical Inference Engine
Implies
Markov Equivalence Class
X -> Y -> Z
X <- Y -> Z
X <- Y <- Z
Conditional independence constraints
X_||_Z | Y
Causal Discovery Algorithm

The general idea of constraint-based search

Note that the statistical inference engine is separate from the causal discovery algorithm. In this sense constraint-based search is nonparametric - you can plug in whichever conditional independence test you like.

It's a simple approach. But it's still a hard problem!

The search space of causal models

- Search for structure v. Estimation of causal search
- How big is the search space?
  - If $|\mathbf{V}| = n$, we have $\binom{n}{2} = \frac{1}{2}(n-1)n$ distinct pairs of variables
  - Each pair $\{A, B\}$ may be $A \rightarrow B$, $A \leftarrow B$ or $A \quad B$
  - We exclude cyclic graphs. But $\{A, B\}$ may always be either adjacent or non-adjacent.

$$2^{\frac{1}{2}(n-1)n} \leq S \leq 3^{\frac{1}{2}(n-1)n}$$

S grows super-exponentially in n.

Practical difficulty: search space of models is **huge**.

PC Algorithm worked example

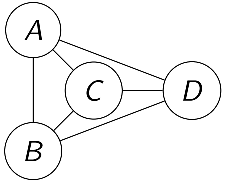


Figure: PC Reconstruction

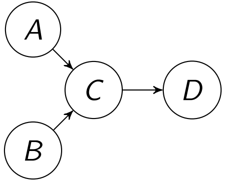


Figure: True graph

1. Start with the complete graph

Named for Peter Spirtes & Clark Glymour. they already used their last names on the Spirtes-Glymour-Scheines algorithm, which is like PC but less efficient.

In this example: every edge we take out is because of Faithfulness; every edge we leave in is because of Markov.

PC Algorithm worked example

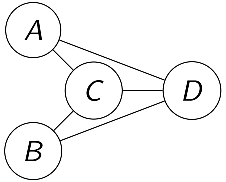


Figure: PC Reconstruction

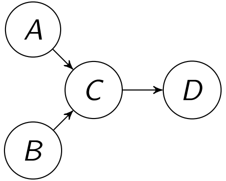


Figure: True graph

1. Start with the complete graph
2. Zero-order conditional independences: $A \perp\!\!\!\perp B$

Start with the most robust CI tests, only do the less robust ones if we need to

PC Algorithm worked example

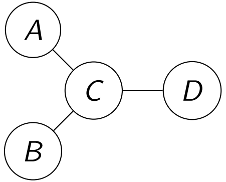


Figure: PC Reconstruction

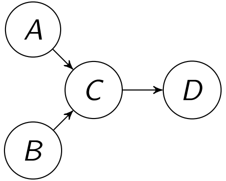


Figure: True graph

1. Start with the complete graph
2. Zero-order conditional independences: $A \perp\!\!\!\perp B$
3. First-order conditional independences: $A \perp\!\!\!\perp D|C$, and $B \perp\!\!\!\perp D|C$
4. No higher-order conditional independences observed
   - Conditioning sets only need to contain neighbours of the two nodes

Markov tells us that you can screen off non-descendants by conditioning on the parents. We don't know which nodes are the parents, but they must be a subset of the neighbours. This means we don't have to iterate over so many possible conditioning sets.

PC Algorithm worked example

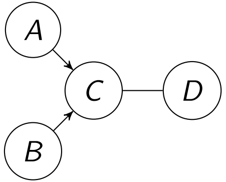


Figure: PC Reconstruction

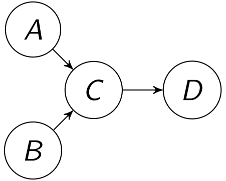


Figure: True graph

1. Start with the complete graph
2. Zero-order conditional independences: $A \perp\!\!\!\perp B$
3. First-order conditional independences: $A \perp\!\!\!\perp D|C$, and $B \perp\!\!\!\perp D|C$
4. No higher-order conditional independences observed
   - Conditioning sets only need to contain neighbours of the two nodes
5. Orient V-structure: $A \not\!\perp\!\!\!\perp B|C$

We saw a v-structure CI pattern, so we can orient those edges

PC Algorithm worked example

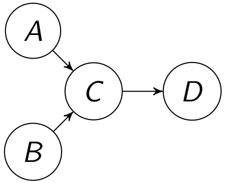


Figure: PC Reconstruction

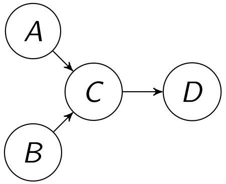


Figure: True graph

1. Start with the complete graph
2. Zero-order conditional independences: $A \perp\!\!\!\perp B$
3. First-order conditional independences: $A \perp\!\!\!\perp D|C$, and $B \perp\!\!\!\perp D|C$
4. No higher-order conditional independences observed
   - Conditioning sets only need to contain neighbours of the two nodes
5. Orient V-structure: $A \not\!\perp\!\!\!\perp B|C$
6. Orient edge away from V-structure

If we oriented the edge $D \rightarrow C$, we'd create another v-structure. But we didn't see a v-structure CI pattern. So the edge must go the other way.

Benefits of PC:

- Tests most robust CIs first
- Removes edges as it goes, reducing number of future tests

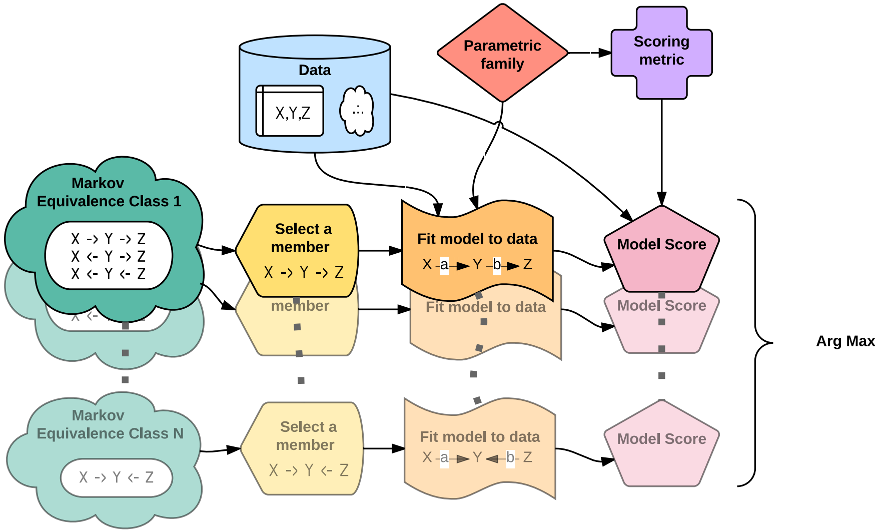
Downsides:

- Propagating errors
- Not so efficient with dense models
- Lots of tests; hard p-value boundary

Note difference between PC-stable and PC

Given infinite data, you'll get all the CI tests right, and if Markov and faithfulness hold, then you'll recover the true model.

Score-based search



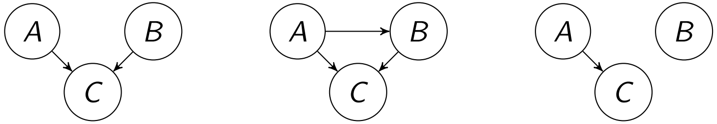
Score-based search requires a parameterization. Like constraint based search, the difficulty is the number of models that need to be evaluated.

Some approaches artificially limit the search space (e.g. by restricting the size of the parent set) and then do exhaustive scoring on the remaining subset. I'll talk about an alternative approach that greedily evaluates a subset of the models.

Score differences

Conditional independences ⇒ score differences

Using a consistent score (like BIC):
$\lim_{n \to \infty} S(G*) > S(G+) > S(G-)$



(a) $G*$ (b) $G+$: $A \not\perp\!\!\!\perp B$ (c) $G-$: $B \perp\!\!\!\perp C$

$$S(G) = \ln(\hat{\mathcal{L}}) - \frac{1}{2}\ln(n)k$$

Say the true model is $G*$. If we use a consistent score to compare it to a model with an extra edge, and a model with a missing edge, the larger model will beat the smaller model, and the true model will beat the larger model.

BIC balances model fit with parsimony. BIC is a consistent score, which means that in the limit of infinite data, it will pick the most parsimonious model that can fit the data. Assuming something a bit weaker than faithfulness, it will pick the true model.

Max Chickering proved the consistency of BIC, and that these score differences can be isolated to local differences of a single edge, which means we can greedily add and remove edges, and will eventually find the true model ... given infinite data.

Greedy Equivalence Search (GES) Algorithm

1. Start with the empty model
2. Forward phase: while there is some valid* edge addition that improves the score,
   2.1 Greedily add whichever valid* edge most improves the score
   2.2 Rebuild the MEC of the augmented model
3. Backward phase: while there is some valid* edge removal that improves the score,
   3.1 Greedily remove whichever valid* edge most improves the score
   3.2 Rebuild the MEC of the diminished model

You can get away with having two phases because the benefits of accounting for real dependences grow with n, whereas the penalty for spurious edges grows with $\log(n)$, so the forward phase will always find a superset of the true edges. But it won't have too many extras because of the complexity penalty, and the extras will all be removed in the backwards phase.

You might wonder why I'm talking about consistency, especially given it only holds when we have infinite data. I emphasise the theoretical results because validation is difficult!

How do we validate our results?

- Prediction: test on set of observed outcomes

Validation is the soft underbelly of causal discovery. In predictive models, we can validate on the same kind of data we use to train.

How do we validate our results?

- Prediction: test on set of observed outcomes
- Causation: test on set of *experimental* outcomes

Validation is the soft underbelly of causal discovery. In predictive models, we can validate on the same kind of data we use to train.

But in causal discovery, we are trying to predict the consequences of interventions, so we need to validate on experimental data.

How do we validate our results?

- Prediction: test on set of observed outcomes
- Causation: test on set of *experimental* outcomes
- Space of potential experiments is huge
- Few instances of good test sets

Validation is the soft underbelly of causal discovery. In predictive models, we can validate on the same kind of data we use to train.

But in causal discovery, we are trying to predict the consequences of interventions, so we need to validate on experimental data.

You can experiment on any node - or any subset of nodes, so the powerset of your variables. And you can set the values any way you like. There aren't many test sets with a variety of different experiments.

How do we validate our results?

- Prediction: test on set of observed outcomes
- Causation: test on set of *experimental* outcomes
- Space of potential experiments is huge
- Few instances of good test sets
- Alternatives:
  - Assumptions + theory
  - Simulations
  - Background knowledge

Validation is the soft underbelly of causal discovery. In predictive models, we can validate on the same kind of data we use to train.

But in causal discovery, we are trying to predict the consequences of interventions, so we need to validate on experimental data.

You can experiment on any node - or any subset of nodes, so the powerset of your variables. And you can set the values any way you like. There aren't many test sets with a variety of different experiments.

The alternatives are all somewhat unsatisfying. Theory is nice, but the assumptions will always be violated to some degree - we want to know how robust the algorithms are to realistic violations of the assumptions.

Often people will use simulation - create a synthetic causal structure, generate data from it, and see how well they can recover the known structure from the data. That's generally a good approach, and most papers in the field use it. But it is really hard to generate synthetic data that violates the assumptions in a realistic way.

Background knowledge is a way of saying we did the experiment in the past already.

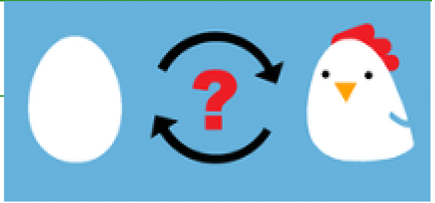
I'll show you a couple of really good examples of validation, because it can be done!

Case study: *Arabidopsis thaliana*

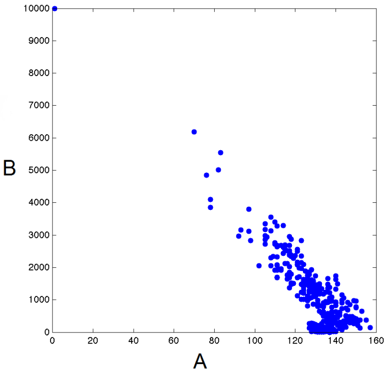
Stekhoven, Daniel J., et al. "Causal stability ranking." *Bioinformatics* 28.21 (2012): 2819-2823.

- Used observational data on gene expression and flowering time
- Methods: PC, subsample for edge stability, estimate total causal effects with IDA
- Of the 25 top genes:
  - 5 were known influences on flowering time
  - 13 others had mutant lines available
  - 9 produced enough viable plants
  - 4 of the 9 had a different flowering time

Kaggle Cause-Effect Pairs



- 4k training pairs
- Most are semi-artificial; real pairs number in the hundreds
- “from domains as diverse as chemistry, climatology, ecology, economy, engineering, epidemiology, genomics, medicine, physics and sociology”
- Winning entry had accuracy 0.82

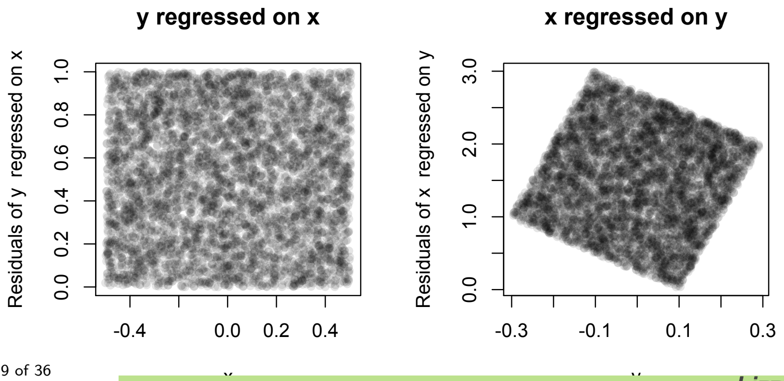


Organized by Isabelle Guyon

Website: http://www.causality.inf.ethz.ch/cause-effect.php

LiNGAM Example

```
# uniform noise
x <- runif(5000)
y <- runif(5000) + 2*x
```



Linear Non-Gaussian Acyclic Model

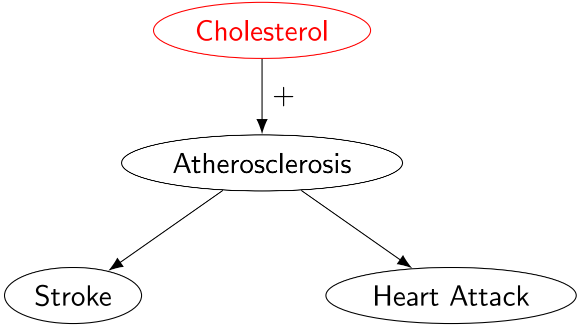
Intuition behind LiNGAM: if you fit the reverse model, the residuals become dependent on the regressor.

Assumptions

- Markov
- Faithfulness
- Acyclicity
- Causal sufficiency
- Non-linearity
- Non-Gaussianity
- Compositionality
- Positivity
- No measurement error
- ...

Let's talk about some other assumptions that can fail!

Ambiguous variables

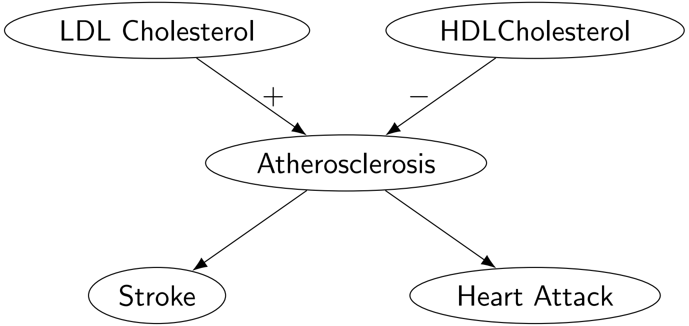


In the 80s there were a bunch of RCTs that tested drugs to reduce cholesterol. And some, like the Coronary Drug Project, successfully reduced cholesterol, but had no effect on heart disease. Why? Because cholesterol is really two things - LDL cholesterol, which increases heart disease, and HDL cholesterol, which decreases it. Some interventions were targeting one, or the other, or both. If your variables don't have an unambiguous interpretation in terms of interventions, your model won't represent the results of those interventions properly.

I mention this in the context of machine learning particularly, because often we do a lot of feature engineering and throw everything into the model. That might improve prediction - and it's tempting to throw in as much as possible, to avoid omitting common causes. But if it screws up the intervention interpretation of those variables, you can run into trouble.

I want to emphasize that I don't know of a solution for this, a method for identifying the causal variables. I just want to highlight the problem.

Ambiguous variables



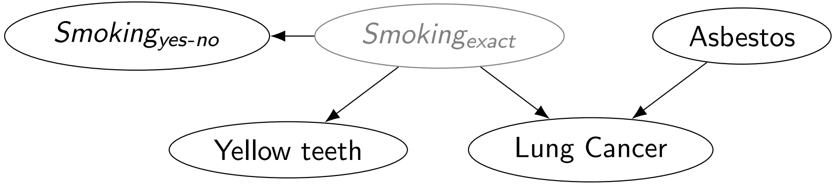
In the 80s there were a bunch of RCTs that tested drugs to reduce cholesterol. And some, like the Coronary Drug Project, successfully reduced cholesterol, but had no effect on heart disease. Why? Because cholesterol is really two things - LDL cholesterol, which increases heart disease, and HDL cholesterol, which decreases it. Some interventions were targeting one, or the other, or both. If your variables don't have an unambiguous interpretation in terms of interventions, your model won't represent the results of those interventions properly.

I mention this in the context of machine learning particularly, because often we do a lot of feature engineering and throw everything into the model. That might improve prediction - and it's tempting to throw in as much as possible, to avoid omitting common causes. But if it screws up the intervention interpretation of those variables, you can run into trouble.

I want to emphasize that I don't know of a solution for this, a method for identifying the causal variables. I just want to highlight the problem.

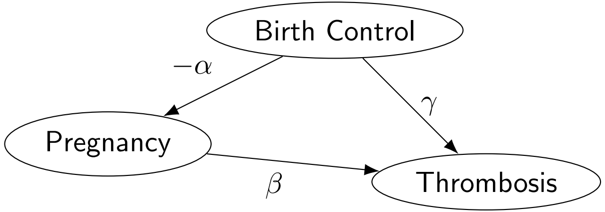
Measurement error

$$
\begin{aligned}
\mathit{Smoking}_{\mathit{exact}} &:= f_1(\epsilon_{\mathit{smoking\text{-}exact}}) \\
\mathit{Smoking}_{\mathit{yes\text{-}no}} &:= f_2(\mathit{Smoking}_{\mathit{exact}}, \epsilon_{\mathit{smoking\text{-}yes\text{-}no}}) \\
\mathit{Asbestos} &:= f_3(\epsilon_{\mathit{asbestos}}) \\
\mathit{Yellow\ Teeth} &:= f_4(\mathit{Smoking}_{\mathit{exact}}, \epsilon_{\mathit{teeth}}) \\
\mathit{Lung\ Cancer} &:= f_5(\mathit{Smoking}_{\mathit{exact}}, \epsilon_{\mathit{cancer}})
\end{aligned}
$$



Say you don't measure smoking exactly, but just some coarsened version of it. Then the structural equations will have the coarsened version as a function of the exact one; and the graph will show the coarsened variable as a child of the exact one. But then you won't be able to block that path from yellow teeth to lung cancer; you won't see the conditional independence you expect. I mention this because measurement error is pervasive!

Not uniformly consistent



Say $\alpha\beta = \gamma + \varepsilon$.

For any $n < \infty$, I can pick an ε such that PC says there's no edge *Birth Control* → *Thrombosis*

The error is $\gamma - \hat{\gamma} = \gamma - 0$ (arbitrarily large)

What if we assume ε is bounded away from zero?

For any finite sample size, we can make an arbitrarily large error (in terms of edge coefficients). So even though we get the right answer with infinite data (pointwise consistency), we don't have uniform consistency, which would let us bound our error at finite sample sizes.

A tempting response is to say, okay, we can't get that close to unfaithfulness. Assume epsilon can't be smaller than some value, and you get back uniform consistency. However ...

How strong is the strong faithfulness assumption?

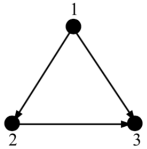
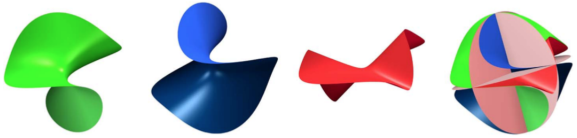


FIG. 1. *Motivating example: 3-node graph.*



(a) $\mathrm{cov}(X_1, X_3)=0$ (b) $\mathrm{cov}(X_1, X_2|X_3)=0$ (c) $\mathrm{cov}(X_2, X_3)=0$ (d) All 6 surfaces

FIG. 2. *Parameter values corresponding to unfaithful distributions in the 3-node case.*

Uhler, Caroline, et al. "Geometry of the faithfulness assumption in causal inference." The Annals of Statistics (2013): 436-463.

The faithfulness assumption is often justified on the grounds that the surface of unfaithfulness is Lebesgue measure zero, so has zero probability. But the volume *close* to the unfaithful surface may be surprisingly large.

Mention k-triangle faithfulness?

Why use causal structure learning algorithms?

In some circumstances the only alternative may be "guess and test"

Often what scientists do - for example, in psychology - is 'guess and test'. Specify (guess) a model, fit it to data, test the fit. If the test passes, declare that you have found the true model. There are no theoretical guarantees for this procedure. It won't even tell you that another model in the same Markov equivalence class would both pass the test!

Whatever worries you have about causal discovery algorithms – and you should have a lot! – you should have *more* worries about 'guess and test'.

Use causal discovery. It's the least bad option. And do your best to validate!

Why use causal structure learning algorithms?

In some circumstances the only alternative may be “guess and test”

That’s all, folks

Often what scientists do - for example, in psychology - is ‘guess and test’. Specify (guess) a model, fit it to data, test the fit. If the test passes, declare that you have found the true model. There are no theoretical guarantees for this procedure. It won’t even tell you that another model in the same Markov equivalence class would both pass the test!

Whatever worries you have about causal discovery algorithms – and you should have a lot! – you should have *more* worries about ‘guess and test’.

Use causal discovery. It’s the least bad option. And do your best to validate!

Selected references

- **GES:** Chickering, David Maxwell. "Optimal structure identification with greedy search." *Journal of machine learning research* 3.Nov (2002): 507-554.
- **PC-Stable:** Maathuis, Marloes H., et al. "Predicting causal effects in large-scale systems from observational data." *Nature Methods* 7.4 (2010): 247.
- **Strong Faithfulness:** Uhler, Caroline, et al. "Geometry of the faithfulness assumption in causal inference." *The Annals of Statistics* (2013): 436-463.
- **LiNGAM:** Shimizu, Shohei, et al. "A linear non-Gaussian acyclic model for causal discovery." *Journal of Machine Learning Research* 7.Oct (2006): 2003-2030.
- **PC (and causal discovery in general):** Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., & Richardson, T. (2000). *Causation, prediction, and search*. MIT press.
- **Arabidopsis validation:** Stekhoven, Daniel J., et al. "Causal stability ranking." *Bioinformatics* 28.21 (2012): 2819-2823.
- **FCI:** Zhang, Jiji. *Causal inference and reasoning in causally insufficient systems.* Diss. PhD thesis, Carnegie Mellon University, 2006.

Another relevant talk: 'All of Causal Discovery', by Frederick Eberhardt (on YouTube)