Chapter 25

Discovering Causal Structure from Observations

The last few chapters have, hopefully, convinced you that when you want to do causal inference, knowing the causal graph is very helpful. We have looked at how it would let us calculate the effects of actual or hypothetical manipulations of the variables in the system. Furthermore, knowing the graph tells us about what causal effects we can and cannot identify, and estimate, from observational data. But everything has posited that we know the graph somehow. This chapter finally deals with where the graph comes from.

There are fundamentally three ways to get the DAG:

- Prior knowledge
- Guessing-and-testing
- Discovery algorithms

There is only a little to say about the first, because, while it's important, it's not very statistical. As functioning adult human beings, you have a lot of everyday causal knowledge, which does not disappear the moment you start doing data analysis. Moreover, you are the inheritor of a vast scientific tradition which has, through patient observation and toilsome experiments, acquired even more causal knowledge. You can and should use this. Someone's sex or race or caste might be causes of the job they get or their pay, but not the other way around. Running an electric current through a wire produces heat at a rate proportional to the square of the current. Malaria is due to a parasite transmitted by mosquitoes, and spraying mosquitoes with insecticides makes the survivors more resistant to those chemicals. All of these sorts of ideas can be expressed graphically, or at least as constraints on graphs.

We can, and should, also use graphs to represent scientific ideas which are not as secure as Ohm's law or the epidemiology of malaria. The ideas people work with in areas like psychology or economics, are really quite tentative, but they are ideas

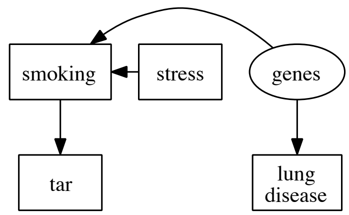


Figure 25.1: A hypothetical causal model in which smoking is associated with lung disease, but does not cause it. Rather, both smoking and lung disease are caused by common genetic variants. (This idea was due to R. A. Fisher.) Smoking is also caused, in this model, by stress.

about the causal structure of parts of the world, and so graphical models are implicit in them.

All of which said, even if we think we know very well what's going on, we will often still want to check it, and that brings us the guess-and-test route.

25.1 Testing DAGs

A graphical causal model makes two kinds of qualitative claims. One is about direct causation. If the model says X is a parent of Y, then it says that changing X will change the (distribution of) Y. If we experiment on X (alone), moving it back and forth, and yet Y is unaltered, we know the model is wrong and can throw it out.

The other kind of claim a DAG model makes is about probabilistic conditional independence. If S d-separates X from Y, then $X \perp\!\!\!\perp Y|S$. If we observed X, Y and S, and see that $X \not\perp\!\!\!\perp Y|S$, then we know the model is wrong and can throw it out. (More: we know that there is a path linking X and Y which isn't blocked by S.) Thus in the model of Figure 25.1, lungdisease$\perp\!\!\!\perp$tar|smoking. If lung disease and tar turn out to be dependent when conditioning on smoking, the model must be wrong.

This then is the basis for the guess-and-test approach to getting the DAG:

- Start with an initial guess about the DAG.
- Deduce conditional independence relations from d-separation.
- Test these, and reject the DAG if variables which ought to be conditionally independent turn out to be dependent.

This is a distillation of primary-school scientific method: formulate a hypotheses (the DAG), work out what the hypothesis implies, test those predictions, reject hypotheses which make wrong predictions.

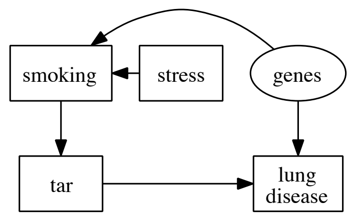


Figure 25.2: As in Figure 25.1, but now tar in the lungs does cause lung disease.

It may happen that there are only a few competing, scientifically-plausible models, and so only a few, competing DAGs. Then it is usually a good idea to focus on checking predictions which *differ* between them. So in both Figure 25.1 and in Figure 25.2, stress$\perp\!\!\!\perp$tar|smoking. Checking that independence thus does nothing to help us distinguish between the two graphs. In particular, confirming that stress and tar are independent given smoking really doesn't give us evidence *for* the model from Figure 25.1, since it equally follows from the other model. If we want such evidence, we have to look for something they *disagree* about.

In any case, testing a DAG means testing conditional independence, so let's turn to that next.

25.2 Testing Conditional Independence

Recall from §22.4 that conditional independence is equivalent to zero conditional information: $X \perp\!\!\!\perp Y|Z$ if and only if $I[X;Y|Z] = 0$. In principle, this solves the problem. In practice, estimating mutual information is non-trivial, and in particular the sample mutual information often has a very complicated distribution. You *could* always bootstrap it, but often something more tractable is desirable. Completely general conditional independence testing is actually an active area of research. Some of this work is still quite mathematical (Sriperumbudur *et al.*, 2010), but it has already led to practical tests (Székely and Rizzo, 2009; Gretton *et al.*, 2012; Zhang *et al.*, 2011) and no doubt more are coming soon.

If all the variables are discrete, one just has a big contingency table problem, and could use a G^2 or χ^2 test. If everything is linear and multivariate Gaussian, $X \perp\!\!\!\perp Y|Z$ is equivalent to zero partial correlation[1]. Nonlinearly, if $X \perp\!\!\!\perp Y|Z$, then $\mathbf{E}[Y|Z] = \mathbf{E}[Y|X,Z]$, so if smoothing Y on X and Z leads to different predictions than just smoothing on Z, conditional independence fails. To reverse this, and go from $\mathbf{E}[Y|Z] = \mathbf{E}[Y|X,Z]$ to $X \perp\!\!\!\perp Y|Z$, requires the extra assumption that Y doesn't

[1]Recall that the partial correlation between X and Y given Z is the correlation between X and Y, after linearly regressing each of them on Z separately. That is, it is the correlation of their residuals.

depend on X through its variance or any other moment. (This is weaker than the linear-and-Gaussian assumption, of course.)

The conditional independence relation $X \perp\!\!\!\perp Y | Z$ is fully equivalent to $\Pr(Y|X,Z) = \Pr(Y|Z)$. We could check this using non-parametric density estimation, though we would have to bootstrap the distribution of the test statistic. A more automatic, if slightly less rigorous, procedure comes from the idea mentioned in Chapter 15: If X is in fact useless for predicting Y given Z, then an adaptive bandwidth selection procedure (like cross-validation) should realize that giving any finite bandwidth to X just leads to over-fitting. The bandwidth given to X should tend to the maximum allowed, smoothing X away altogether. This argument can be made more formal, and made into the basis of a test (Hall *et al.*, 2004; Li and Racine, 2007).

25.3 Faithfulness and Equivalence

In graphical models, d-separation implies conditional independence: if S blocks all paths from U to V, then $U \perp\!\!\!\perp V | S$. To reverse this, and conclude that if $U \perp\!\!\!\perp V | S$ then S must d-separate U and V, we need an additional assumption, already referred to in §22.2, called **faithfulness**. More exactly, if the distribution is faithful to the graph, then if S does not d-separate U from V, $U \not\perp\!\!\!\perp V | S$. The combination of faithfulness and the Markov property means that $U \perp\!\!\!\perp V | S$ if and only if S d-separates U and V.

This seems extremely promising. We can test whether $U \perp\!\!\!\perp V | S$ for any sets of variables we like. We could in particular test whether each pair of variables is independent, given all sorts of conditioning variable sets S. If we assume faithfulness, when we find that $X \perp\!\!\!\perp Y | S$, we know that S blocks all paths linking X and Y, so we learn something about the graph. If $X \not\perp\!\!\!\perp Y | S$ for all S, we would seem to have little choice but to conclude that X and Y are directly connected. Might it not be possible to reconstruct or discover the right DAG from knowing all the conditional independence and dependence relations?

This is on the right track, but too hasty. Start with just two variables:

$$X \rightarrow Y \quad \Rightarrow \quad X \not\perp\!\!\!\perp Y \tag{25.1}$$
$$X \leftarrow Y \quad \Rightarrow \quad X \not\perp\!\!\!\perp Y \tag{25.2}$$

With only two variables, there is only one independence (or dependence) relation to worry about, and it's the same no matter which way the arrow points.

Similarly, consider these arrangements of three variables:

$$X \quad \rightarrow Y \quad \rightarrow Z \tag{25.3}$$
$$X \quad \leftarrow Y \quad \leftarrow Z \tag{25.4}$$
$$X \quad \leftarrow Y \quad \rightarrow Z \tag{25.5}$$
$$X \quad \rightarrow Y \quad \leftarrow Z \tag{25.6}$$

The first two are chains, the third is a fork, the last is a collider. It is not hard to check (Exercise 1) that the first three DAGs all imply exactly the same set of conditional independence relations, which are different from those implied by the fourth[2].

These examples illustrate a general problem. There may be multiple graphs which imply the same independence relations, even when we assume faithfulness. When this happens, the exact same distribution of observables can factor according to, and be faithful to, all of those graphs. The graphs are thus said to be **equivalent**, or **Markov equivalent**. Observational alone cannot distinguish between equivalent DAGs. Experiment can, of course — changing Y alters both X and Z in a fork, but not a chain — which shows that there really is a difference between the DAGs, just not one *observational* data can track.

25.3.1 Partial Identification of Effects

Chapters 23–24 considered the identification and estimation of causal effects under the assumption that there was a single known graph. If there are multiple equivalent DAGs, then, as mentioned above, no amount of purely observational data can select a single graph. Background knowledge lets us rule out some equivalent DAGs[3], but it may not narrow the set of possibilities to a single graph. How then are we to actually do our causal estimation?

We *could* just pick one of the equivalent graphs, and do all of our calculations as though it were the only possible graph. This is often what people seem to do. The kindest thing one can say about it is that it shows confidence; phrases like "lying by omission" also come to mind.

A more principled alternative is to admit that the uncertainty about the DAG means that causal effects are only *partially* identified. Simply put, one does the estimation in each of the equivalent graphs, and reports the range of results[4]. If each estimate is consistent, then this gives a consistent estimate of the range of possible effects. Because the effects are not fully identified, this range will not narrow to a single point, even in the limit of infinite data, but admitting this, rather than claiming a non-existent precision, is simple scientific honesty.

25.4 Causal Discovery with Known Variables

Section 25.1 talks about how we can test a DAG, once we have it. This lets us eliminate some DAGs, but still leaves mysterious where they come from in the first place. While in principle there is nothing wrong which deriving your DAG from a vision of serpents biting each others' tails, so long as you test it, it would be nice to have a systematic way of finding good models. This is the problem of model discovery, and especially of causal discovery.

[2]In all of the first three, $X \not\perp\!\!\!\perp Z$ but $X \perp\!\!\!\perp Z|Y$, while in the collider, $X \perp\!\!\!\perp Z$ but $X \not\perp\!\!\!\perp Z|Y$. Remarkably enough, the work which introduced the notion of forks and colliders, Reichenbach (1956), missed this — he thought that $X \perp\!\!\!\perp Z|Y$ in a collider as well as a fork. Arguably, this one mistake delayed the development of causal inference by thirty years or more.

[3]If we know that X, Y and Z have to be in either a chain or a fork, with Y in the middle, and we know that X comes before Y in time, then we can rule out the fork and the chain $X \leftarrow Y \rightarrow Z$.

[4]Sometimes the different graphs will gave the same estimates of certain effects. For example, the chain $X \rightarrow Y \rightarrow Z$ and the fork $X \leftarrow Y \rightarrow Z$ will agree on the effect of Y on Z.

Causal discovery is silly with just one variable, and too hard for us with just two.[5]

With three or more variables, we have however a very basic principle. If there is no edge between X and Y, in either direction, then X is neither Y's parent nor its child. But any variable is independent of its non-descendants given its parents. Thus, for some set[6] of variables S, $X \perp\!\!\!\perp Y|S$ (Exercise 2). If we assume faithfulness, then the converse holds: if $X \perp\!\!\!\perp Y|S$, then there cannot be an edge between X and Y. Thus, there is no edge between X and Y if and only if we can make X and Y independent by conditioning on some S. Said another way, there is an edge between X and Y if and only if we cannot make the dependence between them go away, no matter what we condition on[7].

So let's start with three variables, X, Y and Z. By testing for independence and conditional independence, we could learn that there had to be edges between X and Y and Y and Z, but not between X and Z. But conditional independence is a symmetric relationship, so how could we **orient** those edges, give them direction? Well, to rehearse a point from the last section, there are only four possible directed graphs corresponding to that undirected graph:

- $X \rightarrow Y \rightarrow Z$ (a chain);
- $X \leftarrow Y \leftarrow Z$ (the other chain);
- $X \leftarrow Y \rightarrow Z$ (a fork on Y);
- $X \rightarrow Y \leftarrow Z$ (a collision at Y)

With the fork or either chain, we have $X \perp\!\!\!\perp Z|Y$. On the other hand, with the collider we have $X \not\perp\!\!\!\perp Z|Y$. Thus $X \not\perp\!\!\!\perp Z|Y$ if and only if there is a collision at Y. By testing for *this* conditional dependence, we can either definitely orient the edges, or rule out an orientation. If $X - Y - Z$ is just a subgraph of a larger graph, we can still identify it as a collider if $X \not\perp\!\!\!\perp Z|\{Y, S\}$ for *all* collections of nodes S (not including X and Z themselves, of course).

With more nodes and edges, we can **induce** more orientations of edges by consistency with orientations we get by identifying colliders. For example, suppose we know that X, Y, Z is either a chain or a fork on Y. If we learn that $X \rightarrow Y$, then the triple *cannot* be a fork, and must be the chain $X \rightarrow Y \rightarrow Z$. So orienting the $X - Y$ edge induces an orientation of the $Y - Z$ edge. We can also sometimes orient edges through background knowledge; for instance we might know that Y comes later in time than X, so if there is an edge between them it *cannot* run from Y to X.[8] We can

[5]But see Janzing (2007); Hoyer *et al.* (2009) for some ideas on how you could do it if you're willing to make some extra assumptions. The basic idea of these papers is that the distribution of effects given causes should be simpler, in some sense, than the distribution of causes given effects.

[6]Possibly empty: conditioning on the empty set of variables is the same as not conditioning at all.

[7]"No causation without association", as it were.

[8]Some have argued, or at least entertained the idea, that the logic here is backwards: rather than order in time constraining causal relations, causal order *defines* time order. (Versions of this idea are discussed by, inter alia, Russell (1927); Wiener (1961); Reichenbach (1956); Pearl (2009b); Janzing (2007) makes a related suggestion). Arguably then using order in time to orient edges in a causal graph begs the question, or commits the fallacy of *petitio principii*. But of course every syllogism does, so this isn't a distinctively *statistical* issue. (Take the classic: "All men are mortal; Socrates is a man; therefore Socrates is mortal."

eliminate other edges based on similar sorts of background knowledge: men tend to be heavier than women, but changing weight does not change sex, so there can't be an edge (or even a directed path!) from weight to sex, though there could be one the other way around.

To sum up, we can rule out an edge between X and Y whenever we can make them independent by conditioning on other variables; and when we have an $X - Y - Z$ pattern, we can identify colliders by testing whether X and Z are dependent given Y. Having oriented the arrows going into colliders, we induce more orientations of other edges.

Putting these three things — edge elimination by testing, collider finding, and inducing orientations — gives the most basic causal discovery procedure, the SGS (Spirtes-Glymour-Scheines) algorithm (Spirtes *et al.*, 2001, §5.4.1, p. 82). This assumes:

1. The data-generating distribution has the causal Markov property on a graph G.
2. The data-generating distribution is faithful to G.
3. Every member of the population has the same distribution.
4. All relevant variables are in G.
5. There is only *one* graph G to which the distribution is faithful.

Abstractly, the algorithm works as follows:

- Start with a complete undirected graph on all p variables, with edges between all nodes.
- For each pair of variables X and Y, and each set of other variables S, see if $X \perp\!\!\!\perp Y | S$; if so, remove the edge between X and Y.
- Find colliders by checking for conditional dependence; orient the edges of colliders.
- Try to orient undirected edges by consistency with already-oriented edges; do this recursively until no more edges can be oriented.

Pseudo-code is in Appendix F.

Call the result of the SGS algorithm $\widehat{G}$. If all of the assumptions above hold, and the algorithm is correct in its guesses about when variables are conditionally independent, then $\widehat{G} = G$. In practice, of course, conditional independence guesses are really statistical tests based on finite data, so we should write the output as $\widehat{G}_n$, to indicate that it is based on only n samples. If the conditional independence test is consistent, then

$$\lim_{n\rightarrow\infty} \Pr\left(\widehat{G}_n \neq G\right) = 0 \tag{25.7}$$

How can we know that *all* men are mortal until we know about the mortality of this particular man, Socrates? Isn't this just like asserting that tomatoes and peppers must be poisonous, because they belong to the nightshade family of plants, all of which are poisonous?) While these philosophical issues are genuinely fascinating, this footnote has gone on long enough, and it is time to return to the main text.

In other words, the SGS algorithm converges in probability on the correct causal structure; it is consistent for all graphs G. Of course, at finite n, the probability of error — of having the wrong structure — is (generally!) not zero, but this just means that, like any statistical procedure, we cannot be absolutely certain that it's not making a mistake.

One consequence of the independence tests making errors on finite data can be that we fail to orient some edges — perhaps we missed some colliders. These unoriented edges in $\widehat{G}_n$ can be thought of as something like a confidence region — they have *some* orientation, but multiple orientations are all compatible with the data.[9] As more and more edges get oriented, the confidence region shrinks.

If the fifth assumption above fails to hold, then there are multiple graphs G to which the distribution is faithful. This is just a more complicated version of the difficulty of distinguishing between the graphs $X \rightarrow Y$ and $X \leftarrow Y$. All the graphs in the equivalence class may have some arrows in common; in that case the SGS algorithm will identify those arrows. If some edges differ in orientation across the equivalence class, SGS will not orient them, even in the limit. In terms of the previous paragraph, the confidence region never shrinks to a single point, just because the data doesn't provide the information needed to do this. The graph is only partially identified.

If there *are* unmeasured relevant variables, we can get not just unoriented edges, but actually arrows pointing in both directions. This is an excellent sign that some basic assumption is being violated.

25.4.1 The PC Algorithm

The SGS algorithm is statistically consistent, but very computationally inefficient; the number of tests it does grows exponentially in the number of variables p. This is the worst-case complexity for *any* consistent causal-discovery procedure, but this algorithm just proceeds immediately to the worst case, not taking advantage of any possible short-cuts.

Since it's enough to find *one* S making X and Y independent to remove their edge, one obvious short-cut is to do the tests in some order, and skip unnecessary tests. On the principle of doing the easy work first, the revised edge-removal step would look something like this:

- For each X and Y, see if $X \perp\!\!\!\perp Y$; if so, remove their edge.
- For each X and Y which are still connected, and each third variable Z_1, see if $X \perp\!\!\!\perp Y | Z$; if so, remove the edge between X and Y.
- For each X and Y which are still connected, and each third and fourth variables Z_1 and Z_2, see if $X \perp\!\!\!\perp Y | Z_1, Z_2$; if so, remove their edge.
- $\ldots$

[9] I say "multiple orientations" rather than "all orientations", because picking a direction for one edge might induce an orientation for others.

- For each X and Y which are still connected, see if $X \perp\!\!\!\perp Y|$ all the $p-2$ other variables; if so, remove, their edge.

If all the tests are done correctly, this will give the same result as the SGS procedure (Exercise 3). And if some of the tests give erroneous results, conditioning on a small number of variables will tend to be more reliable than conditioning on more (why?).

We can be even more efficient, however. If $X \perp\!\!\!\perp Y|S$ for any S at all, then $X \perp\!\!\!\perp Y|S'$, where all the variables in S' are adjacent to X or Y (or both) (Exercise 4). To see the sense of this, suppose that there is a single long directed path running from X to Y. If we condition on any of the variables along the chain, we make X and Y independent, but we could always move the point where we block the chain to be either right next to X or right next to Y. So when we are trying to remove edges and make X and Y independent, we only need to condition on variables which are still connected to X and Y, not ones in totally different parts of the graph.

This then gives us the PC[10] algorithm (Spirtes *et al.* 2001, §5.4.2, pp. 84–88; see also Appendix F). It works exactly like the SGS algorithm, except for the edge-removal step, where it tries to condition on as few variables as possible (as above), and only conditions on adjacent variables. The PC algorithm has the same assumptions as the SGS algorithm, and the same consistency properties, but generally runs much faster, and does many fewer statistical tests. It should be the default algorithm for attempting causal discovery.

25.4.2 Causal Discovery with Hidden Variables

Suppose that the set of variables we measure is *not* causally sufficient. Could we at least discover this? Could we possibly get hold of *some* of the causal relationships? Algorithms which can do this exist (e.g., the CI and FCI algorithms of Spirtes *et al.* (2001, ch. 6)), but they require considerably more graph-fu. (The RFCI algorithm (Colombo *et al.*, 2012) is a modern, fast successor to FCI.) The results of these algorithms can succeed in removing *some* edges between observable variables, and definitely orienting some of the remaining edges. If there are actually no latent common causes, they end up acting like the SGS or PC algorithms.

Partial identification of effects When all relevant variables are observed, all effects are identified within one graph; partial identification happens because multiple graphs are equivalent. When some variables are not observed, we may have to use the identification strategies to get at the same effect. In fact, the same effect my be identified in one graph and not identified in another, equivalent graph. This is, again, unfortunate, but when it happens it needs to be admitted.

25.4.3 On Conditional Independence Tests

The abstract algorithms for causal discovery assume the existence of consistent tests for conditional independence. The implementations known to me mostly assume either that variables are discrete (so that one can basically use the χ^2 test), or that

[10] Peter-Clark

they are continuous, Gaussian, and linearly related (so that one can test for vanishing partial correlations), though the `pcalg` package does allow users to provide their own conditional independence tests as arguments. It bears emphasizing that these restrictions are *not* essential. As soon as you have a consistent independence test, you are, in principle, in business. In particular, consistent *non-parametric* tests of conditional independence would work perfectly well. An interesting example of this is the paper by Chu and Glymour (2008), on finding causal models for the time series, assuming additive but non-linear models.

25.5 Software and Examples

The PC and FCI algorithms are implemented in the stand-alone Java program `Tetrad` (`http://www.phil.cmu.edu/projects/tetrad/`). They are also implemented in the `pcalg` package on CRAN (Kalisch *et al.*, 2010, 2011). This package also includes functions for calculating the effects of interventions from fitted graphs, assuming linear models. The documentation for the functions is somewhat confusing; rather see Kalisch *et al.* (2011) for a tutorial introduction.

It's worth going through how `pcalg` works[11]. The code is designed to take advantage of the modularity and abstraction of the PC algorithm itself; it separates actually finding the graph completely from performing the conditional independence test, which is rather a function the user supplies. (Some common ones are built in.) For reasons of computational efficiency, in turn, the conditional independence tests are set up so that the user can just supply a set of sufficient statistics, rather than the raw data.

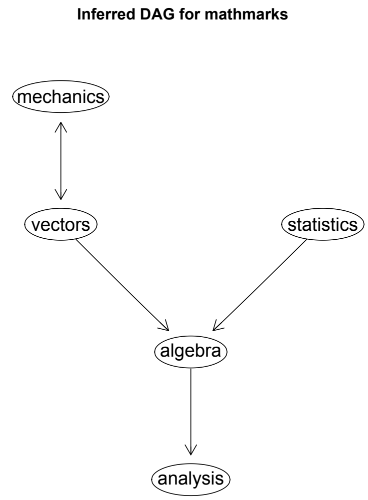
Let's walk through an example[12], using the `mathmarks` data set you saw in the second exam. There we had grades ("marks") from 88 students in five mathematical subjects, algebra, analysis, mechanics, statistics and vectors. All five variables are positively correlated with each other.

```
library(pcalg)
library(SMPracticals)
data(mathmarks)
suffStat <- list(C=cor(mathmarks),n=nrow(mathmarks))
pc.fit <- pc(suffStat, indepTest=gaussCItest, p=ncol(mathmarks),alpha=0.005
```

This uses a Gaussian (-and-linear) test for conditional independence, `gaussCItest`, which is built into the `pcalg` package. Basically, it tests whether $X \perp\!\!\!\perp Y|Z$ by testing whether the partial correlation of X and Y given Z is close to zero. These partial correlations can all be calculated from the correlation matrix, so the line before creates the sufficient statistics needed by `gaussCItest` — the matrix of correlations

[11]A word about installing the package: you'll need the package `Rgraphviz` for drawing graphs, which is hosted not on CRAN (like `pcalg`) but on BioConductor. Try installing it, and its dependencies, before installing `pcalg`. See `http://www.bioconductor.org/packages/2.10/bioc/readmes/Rgraphviz/README` for help on installing `Rgraphviz`.

[12]After Spirtes *et al.* (2001, §6.12, pp. 152–154).




```
library(Rgraphviz)
plot(pc.fit,labels=colnames(mathmarks),main="Inferred DAG for mathmarks")
```

Figure 25.3: DAG inferred by the PC algorithm from the `mathmarks` data. Two-headed arrows, like undirected edges, indicate that the algorithm was unable to orient the edge. (It is obscure why `pcalg` sometimes gives an edge it cannot orient no heads and sometimes two.)

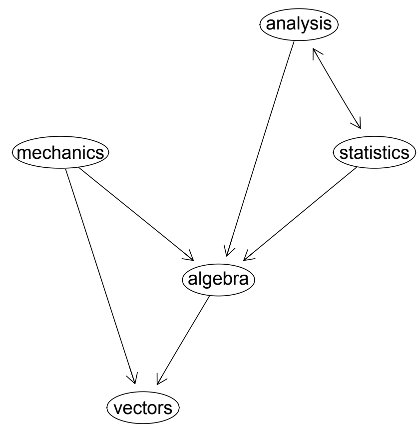
and the number of data points. We also have to tell `pc` how many variables there are, and what significance level to use in the test (here, 0.5%).

Before going on, I encourage you to run `pc` as above, but with `verbose=TRUE`, and to study the output.

Figure 25.3 shows what it looks like. If we take it seriously, it says that grades in analysis are driven by grades in algebra, while algebra in turn is driven by statistics and vectors. While one could make up stories for why this would be so (perhaps something about the curriculum?), it seems safer to regard this as a warning against *blindly* trusting any algorithm —- a key assumption of the PC algorithm, after all, is that there are no unmeasured but causally-relevant variables, and it is easy to believe these are violated. For instance, while *knowledge of* different mathematical fields may be causally linked (it would indeed be hard to learn much mechanics without knowing about vectors), test scores are only imperfect measurements of knowledge.

The size of the test may seem low, but remember we are doing a lot of tests:

```
> summary(pc.fit)
Object of class 'pcAlgo', from Call:
 skeleton(suffStat = suffStat, indepTest = indepTest, p = p, alpha = alpha,
      verbose = verbose, fixedGaps = fixedGaps, fixedEdges = fixedEdges,
```




```
plot(pc(suffStat, indepTest=gaussCItest, p=ncol(mathmarks),alpha=0.05),
  labels=colnames(mathmarks),main="")
```

Figure 25.4: Inferred DAG when the size of the test is 0.05.

```
     NAdelete = NAdelete, m.max = m.max)

Nmb. edgetests during skeleton estimation:
===========================================
Max. order of algorithm:  3
Number of edgetests from m = 0 up to m = 3 :  20 31 4 0

Graphical properties of skeleton:
=================================
Max. number of neighbours:  2 at node(s) 2
Avg. number of neighbours:  1
```

This tells us that it considered going up to conditioning on three variables (the maximum possible, since there are only five variables), that it did twenty tests of unconditional independence, 31 tests where it conditioned on one variable, four tests where it conditioned on two, and none where it conditioned on three. This 55 tests in all, so a simple Bonferroni correction suggests the over-all size is $55 \times 0.005 = 0.275$. This is probably pessimistic (the Bonferroni correction typically is). Setting $\alpha = 0.05$ gives a somewhat different graph (Figure 25.4).

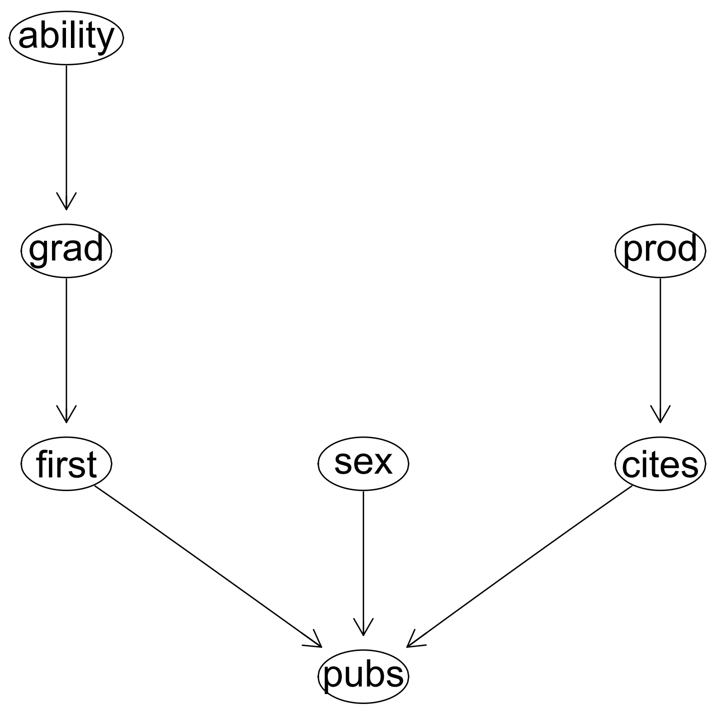
For a second example[13], let's use some data on academic productivity among psy-

[13]Following Spirtes *et al.* (2001, §5.8.1, pp. 98–102).

chologists. The two variables of ultimate interest were the publication (pubs) and citation (cites) rates, with possible measured causes including ability (basically, standardized test scores), graduate program quality grad (basically, the program's national rank), the quality of the psychologist's first job, first, a measure of productivity prod, and sex. There were 162 subjects, and while the actual data isn't reported, the correlation matrix is.

```
> rm
        ability grad prod first   sex cites pubs
ability    1.00 0.62 0.25  0.16 -0.10  0.29 0.18
grad       0.62 1.00 0.09  0.28  0.00  0.25 0.15
prod       0.25 0.09 1.00  0.07  0.03  0.34 0.19
first      0.16 0.28 0.07  1.00  0.10  0.37 0.41
sex       -0.10 0.00 0.03  0.10  1.00  0.13 0.43
cites      0.29 0.25 0.34  0.37  0.13  1.00 0.55
pubs       0.18 0.15 0.19  0.41  0.43  0.55 1.00
```

The model found by pcalg is fairly reasonable (Figure 25.5). Of course, the linear-and-Gaussian assumption has no particular support here, and there is at least one variable for which it must be wrong (which?), but unfortunately with just the correlation matrix we cannot go further.




```
plot(pc(list(C=rm,n=162),indepTest=gaussCItest,p=7,alpha=0.01),
  labels=colnames(rm),main="")
```

Figure 25.5: Causes of academic success among psychologists. The arrow from citations *to* publications is a bit odd, but not impossible — people who get cited more might get more opportunities to do research and so to publish.

25.6 Limitations on Consistency of Causal Discovery

There are some important limitations to causal discovery algorithms (Spirtes *et al.*, 2001, §12.4). They are *universally* consistent: for all causal graphs G,[14]

$$\lim_{n\to\infty} \Pr\left(\widehat{G}_n \neq G\right) = 0 \tag{25.8}$$

The probability of getting the graph wrong can be made arbitrarily small by using enough data. However, this says nothing about *how much* data we need to achieve a given level of confidence, i.e., the *rate* of convergence. *Uniform* consistency would mean that we could put a bound on the probability of error as a function of n which did not depend on the true graph G. Robins *et al.* (2003) proved that *no* uniformly-consistent causal discovery algorithm can exist. The issue, basically, is that the Adversary could make the convergence in Eq. 25.8 arbitrarily slow by selecting a distribution which, while faithful to G, came *very close* to being unfaithful, making some of the dependencies implied by the graph arbitrarily small. For any given dependence strength, there's some amount of data which will let us recognize it with high confidence, but the Adversary can make the required data size as large as he likes by weakening the dependence, without ever setting it to zero.[15]

The upshot is that so *uniform*, *universal* consistency is out of the question; we can be *universally* consistent, but without a uniform rate of convergence; or we can converge *uniformly*, but only on some less-than-universal class of distributions. These might be ones where all the dependencies which do exist are not too weak (and so not too hard to learn reliably from data), or the number of true edges is not too large (so that if we haven't seen edges yet they probably don't exist; Janzing and Herrmann, 2003; Kalisch and Bühlmnann, 2007).

It's worth emphasizing that the Robins *et al.* (2003) no-uniform-consistency result applies to *any* method of discovering causal structure from data. Invoking human judgment, Bayesian priors over causal structures, etc., etc., won't get you out of it.

[14]If the true distribution is faithful to multiple graphs, then we should read G as their equivalence class, which has some undirected edges.

[15]Roughly speaking, if X and Y are dependent given Z, the probability of missing this conditional dependence with a sample of size n should go to zero like $O(2^{-nI[X;Y|Z]})$, I being mutual information. To make this probability equal to, say, α we thus need $n = O(-\log \alpha / I)$ samples. The Adversary can thus make n extremely large by making I very small, yet positive.

25.7 Further Reading

The best single reference on causal discovery algorithms remains Spirtes *et al.* (2001). A lot of work has been done in recent years by the group centered around ETH-Zürich, beginning with Kalisch and Bühlmnann (2007), connecting this to modern statistical concerns about sparse effects and high-dimensional data.

As already mentioned, the best reference on partial identification is Manski (2007). Partial identification of causal effects due to multiple equivalent DAGs is considered in Maathuis *et al.* (2009), along with efficient algorithms for liner systems, which are applied in Maathuis *et al.* (2010), and implemented in the `pcalg` package as `ida()`.

Discovery is possible for directed cyclic graphs, though since it's harder to understand what such models mean, it less well-developed. Important papers on this topic include Richardson (1996) and Lacerda *et al.* (2008).

25.8 Exercises

To think through, not to hand in.

1. Prove that, assuming faithfulness, a three-variable chain and a three-variable fork imply exactly the same set of dependence and independence relations, but that these are different from those implied by a three-variable collider. Are any implications common to chains, forks, and colliders? Could colliders be distinguished from chains and forks without assuming faithfulness?

2. Prove that if X and Y are not parent and child, then either $X \perp\!\!\!\perp Y$, or there exists a set of variables S such that $X \perp\!\!\!\perp Y | S$. *Hint:* start with the Markov property, that any X is independent of all its non-descendants given its parents, and consider separately the cases where Y a descendant of X and those where it is not.

3. Prove that the graph produced by the edge-removal step of the PC algorithm is exactly the same as the graph produced by the edge-removal step of the SGS algorithm. *Hint:* SGS removes the edge between X ad Y when $X \perp\!\!\!\perp Y | S$ for *even one* set S.

4. Prove that if $X \perp\!\!\!\perp Y | S$ for some set of variables S, then $X \perp\!\!\!\perp Y | S'$, where every variable in S' is a neighbor of X or Y.

5. When, exactly, does $\mathbf{E}[Y|X,Z] = \mathbf{E}[Y|Z]$ imply $Y \perp\!\!\!\perp X | Z$?

6. Would the SGS algorithm work on a non-causal, merely-probabilistic DAG? If so, in what sense is it a *causal* discovery algorithm? If not, why not?

7. Describe how to use bandwidth selection as a conditional independence test.

8. Read `pcalg-paper` and write a conditional independence test function based on bandwidth selection. Check that your test function gives the right size when run on test cases where you know the variables are conditionally independent. Check that your test function works with `pcalg::pc`.