Notes About Causal Search and Inference

Robin Fisher

These are notes on causal graphical models, especially in the context of nonparametric casual search where we do not make the assumption of *causal sufficiency*. Think of them as an outline of the pieces we need to implement causal models.

Preliminaries and Assumptions

The basic assumption is that there is a causal DAG underlying the generation of the data. Let $G =< V, E >$, where $V = O\dot{\cup}H\dot{\cup}S$, where O repesents the observed variables, H, hidden or latent variables, and S are selection variables. We get our data by observing O, not observing H, thereby marginalizing it out, and conditioning on S.

The set of DAGs is not closed under marginalization or conditioning, but it is a subset of the set of Maximal Ancestral Graphs (MAGs), which is. We'll assume that the data model is a MAG, $M =< O, E_O >$, where each edge $e \in E$ is a line where each end is $\to$, or $-$. $X \to Y$ means 'X causes Y or some selection variable', $X \leftrightarrow Y$ means X and Y have a common cause but neither causes a selection variable, $X - Y$ means X and Y are ancestors, and thus causes, of some members of S. Methodology for the selection variable is less well developed, but I think it's important. There's some theory showing that if a bunch of variables all have the same common cause among latent variables, it can be represented with pairwise $\leftrightarrow$ (See (Pearl 2009)). If G is an independence map (I-map) of the probability distribution of V (say, $P(v)$) with *d-seperation*, M is an I-map of $P(O)$ with *m-seperation* (J. Zhang 2008a). Further, every MAG M has a *canonical DAG* $\mathcal{D}(M)$, the projection of which is M. $\mathcal{D}(M)$ is the minimal DAG with the corresponding MAG M. See (Richardson, Spirtes, and others 2002) for a discussion and an easy algorithm for finding $\mathcal{D}(M)$. This seems important to me for a couple of reasons. Since we can show that for any undelying causal graph, the data model is a MAG M, then I think the DAG $\mathcal{D}(M)$ represents a minimal faithful I-map of the data model. This DAG can be converted to a decomposable model sutable for quick inference.

An important kind of $\to$ is the 'visible' arrow, which has identifiable characteristics in the graph. From (J. Zhang 2008a),

> Definition 8 (Visibility) Given a MAG M , a directed edge A → B in M is v*isible* if there is a vertex C not adjacent to B, such that either there is an edge between C and A that is into A, or there is a collider path between C and A that is into A and every vertex on the path is a parent of B. Otherwise A → B is said to be *invisible*.

From (Perković et al. 2015):

> Definition 3.2.(Amenability for DAGs, CPDAGs, MAGs and PAGs) A DAG, CPDAG, MAG or PAG $\mathcal{G}$ is said to be *adjustment amenable*, relative to (X, Y) if every possibly directed proper path from X to Y in $\mathcal{G}$ starts with a visible edge out of X.

If the arrow is not visible, the causal relationship itself is identified but there may also be confounding or intermediary variables. (Perković et al. 2015) show that amenability, defined above, is a necessary condition for causal identifiability, so there must be a visible edge. To get a visible edge, we need the instrument (maybe in the form of hte collider path) but, it turns out, the instument does not guarantee the existence of an adjustment set. I'll talk about that more below. I have an example of visibility in Figure 1.

Unfortunately, MAGS are not identifiable. That is, even with perfect knowledge of the joint distribution of O, the set of MAGs which are I-maps of $P(O)$ typically has several members; sets defined this way are equivalence classes, with each class corresponding to an I-map over O. We can represent such a class of MAGs with a *Partial Ancestral Graph* (PAG). A PAG is a graph $Q =< O, E_O >$ where, this time, an edge $e \in E$ is a line, where each end is $\circ$, $\to$, or $-$. $\circ$ implies that the ending of the line is not identifiable. In a

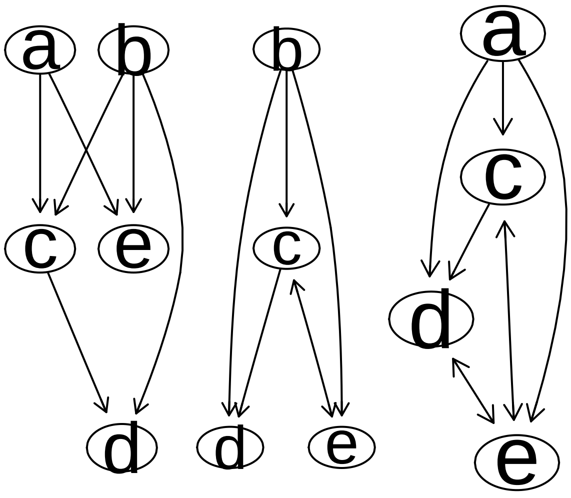


Figure 1: DAG on the left; Center, a is marginalized out; Right, b is marginalized out. Where a is marginalized, out, there is a $\leftrightarrow$ between c and e. On the right, where b is marginalized out, there are $\leftrightarrow$'s between d and e and between c and e. There is a $\rightarrow$ from c to d in the DAG which is preserved in the right graph, but it is confounded with a common parent. This would be an *invisible* arc.

PAG, if we see an from X to Y, that is, $X \rightarrow Y$, we have detected a causal relationship. If it is *definitely visible*, where it visible for all MAGs in the equivalence class defined by the PAG, we at least have a chance of identifying the causal effect. The other relationships from MAGs carry over in similar ways.

Algorithms for Estimating the PAG when the underlying causal graph is a DAG

General

There are only a few algorithms to estimate the PAG. There are a couple of ways to divide them up; the familiar division is by whether they make parametric assumptons or not. the second is whether the algorithm is score-basd or constraint-based. In score-based algorithms, there is some measure of how well the model fits the data, oftne with a penalty for complexity. These methods are pretty well-developed for models with the causal sufficiency assumption or, models without the causal interpretation. Constraint-based models estimate the set of conditional independence relations and use those to infer the structure of the PAG.

The most prominent of the algorithms suitable for nonparametric causal inference are the *FCI* and the *RFCI* algorithms (the Fast Causal Inference and Really Fast Causal Inference algorithms, respectively) These, thogether with some variations, are the main algorithms known to be sound and complete, given an oracle. While many algorithms such as the *pc* algorithm depend on an assumption that the set O be causally identifiable, where there are no latent confounding variables, the FCI and RFCI algorithms do not. These algorithms estimate a collection of conditional indpendence (CI) relations by performing a bunch of CI tests. If the data set is large, there is considerable computational cost in the CI tests. We would also like the tests to be nonparametric, which generally further increases the computer time. The parametric/nonparametric part is determined by the nature of the tests. If the multivariate normal assumption is used, we can use MVN-based tests.

In the nonparametric case, the CI tests are slow, and require a lot of computer time. If the data are catergorical or have been binned,the usual tests of independence for contingnecy tables are available. They are slow, however, and they break for large tables unless there are a lot of observations. I have found this to be the biggest hurdle. The FCI and RFCI algorithms are implemented in the *pcalg* package (Kalisch et al. 2012) in R (R Core Team 2016). Several faster nonparametric test are implemted (Verbyla, Desgranges, and

Wernisch 2017). Another faster CI test is proposed by (Strobl, Zhang, and Visweswaran 2017), of which I have a buggy implementation.

There is theory (Colombo et al. 2012) showing that these algorithms are consistent for the multivariate gaussian case, for sparse graphs in the large sample limit, provide the significance level of the tests goes to zero at an appropriate rate. Finding an appropriate significance for a given situation depends on quanities which must be estimated, however. I also consier the multivariate gaussian assumption very restrictive, but that assumption does make the algorithms run fast. It is also a very popular assumption among, for example the Structural Equations Modelling (SEM) folks. In general, for a fixed set of variables, here is an Argument I think we could make rigorous.

- Given an oracle for CI relationships, FCI and RFCI both return the 'correct' PAG
- For each CI relationship hypothesis, for example, H_0: $X \perp\!\!\!\perp Y|Z$

If the significance $\alpha \to 0$ and the power $1 - \beta \to 1$, so P(H_0 is accepted or rejected correctly) $\to 1$ for every $X, Y, \mathbf{Z}$, so the estimated PAG converges to the true PAG in probability. I don't need *almost surely.*

This is nice, but convergence seems like it might be pretty slow, so such an argument seems pretty academic. I would like a way to measure the reliability ofhe method for moderate or small data sets. The bootstrap seems like a useful method. In [jabbari2017obtaining] they go even a little further and calibrate bootstrap results to give probabilities for the presnce of arcs in a raphical model. I haven't read that one yet.

I include an example below. See (J. Zhang 2008a) for a detailed discussion of ancestral graphs and causal reasoning.

For other kinds of graphical models, there are algorithms which search for an optimum score, such as a penalized likelihood, but, until recently, I thought no such algorithm was available for the causal seach problem. A recent paper seems to provide one, though, based on the identification of Y-structures, and claims several nice properties. I don't know if there is an implementation of that algorithm.

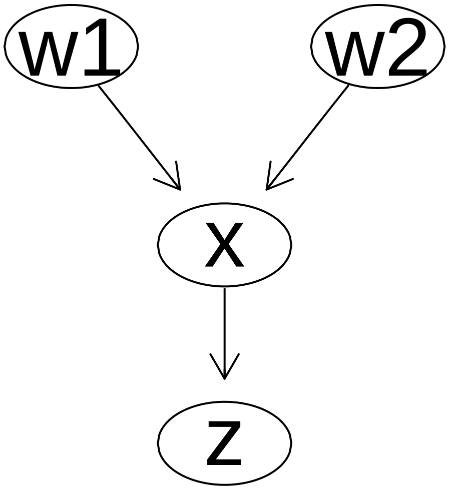
If we make the assumption that the dependence relationships can be expressed as a corelation matrix, it's pretty easy, but I would like to avoid that in spite of the fact that it is a common assumption. Tax data, for example, has a lot of highly skewed data so the normal distribution part has to be finessed, and the relationships often involve thresholds or *max/min*-type functions. I have tried to do it by binning the data, but the tables get to be infeasibly large.

If we are testing $X \perp\!\!\!\perp Y|Z$, where Z may be multivariate, There are some tests based on taking random functions of X, Y, and Z from a suitable set and testing the null hypothesis of no correlation. The idea is that if $X \perp\!\!\!\perp Y|Z$, then $cor(g_1(X), g_2(Y)|g_3(Z)) = 0$ for all g_1, g_2, g_3 and if $cor(g_1(X), g_2(Y)|g_3(Z)) = 0`\forall\ sets\{g_1, g_2, g_3\}$, then $X \perp\!\!\!\perp Y|Z$. Then if we do the test of correlation on enough sets $\{g_1, g_2, g_3\}$, our chance of missing dependencies is low. I don't know of any quantification of 'low', however, except for some simulation results. Anyway, applying the FCI or RFCI algorithms, with a suitable CI test, results in a PAG together with a list of seperating sets (*sep-sets*), which correspond to Z in the CI relationship $X \perp\!\!\!\perp Y|Z$.

Score-based Algorithms

I think I mentioned that score-based algorithms have not been as well-developed as constraint-based algorithms. It may be that it's possible to just plug in the results of the CI tests into algorithms based on oracles. The results are sensitive to each CI, test, though.

An alternative is to maximize some score which measures some overall fitness of the model to the data. (Mani, Spirtes, and Cooper 2012) and (Mani, Cooper, and Statnikov 2006) Approach this problem by looking for embedded Y-structures in the model. An example of a Y-structure is given by in Figure \ref{ystruc}.



The strategy is to choose subsets of the variables in search of *embedded pure Y-structures*, which are just Y-structures on the mariginal distributions of the subset. For any set of four variables, it's not very hard to find the score of all of them. They suggest pretending that the posterior distribution is proportional to the score (not so fanciful, IMO, for some choices of score, like the AIC).

It seems natural to me to proceed with an MCI (see below) algorithm and let the precedence be determined by the 'posterior' when there is a conflict. They use a different strategy, which is to . . . .

I am implementing this in R. The easiest way is to just use a search over the size-4 subset, and keep the score, then order the list by the score, and put that list into the MCI algorithm. Note my implementation ofthat algorithm sets precedence by the order of the list, so this should work conveniently. I need to be sure that the normalization constant isn't an issue. It shouldn't be if the subsets all come fromthe same data set. I'm not so sure if they don't.

I'm not sure about larger conditioning sets; It seems to me that these algorithms can only find a subset of those that the *FCI agorithms can find, so they would not be complete. That's an open question for me. ##Algorithms for finding Adjustment varaibles for Causal effects once the PAG is known.

The *gac* function in the *pcalg* package will identify adjustment variable for the causal effect of one vaiable on another, if it exists.

The Generalized Adjustment Criterion

Having the identifiable arc does not mean that the causal effect is identifiable, but there are algorithms that can find those that are. In particular, we can find visible edges, then, once we have a visible edge, we can use the Generalized Adjustment Criterion (GAC), see (Perković et al. 2015), to test whether a set **Z** is an adjustment set; it is implement in the function *gac* in the package *pcalg*. The function *adjustment* in *pcalg* provides a list of such adjustment sets, providied any exist.

If we soldier on trough this and find identifiable causal relationships, we get to estimate them using a regression method of some form or, more generally, we can estimate conditional distributions.

Finally, I think it is important to consider sensitivity of all of this to sampling or other errors. There is theoretical guidance on the topic, fortunately. See (Schulman and Srivastava 2016)

Comining PAGs

One way we can use the PAGs is through the combination of PAGs from separate data sets over non-disjoint sets of variables. The MCI algorithm uses the collection of sep-sets from the two data sets (because if it is true that $X \perp\!\!\!\perp Y | Z$ in either set, it's true in both) to form a new PAG over the union of he variables, and we may be able to detect new relationships. See (Claassen and Heskes 2010). I have an implementation of this algorithm.

More Examples

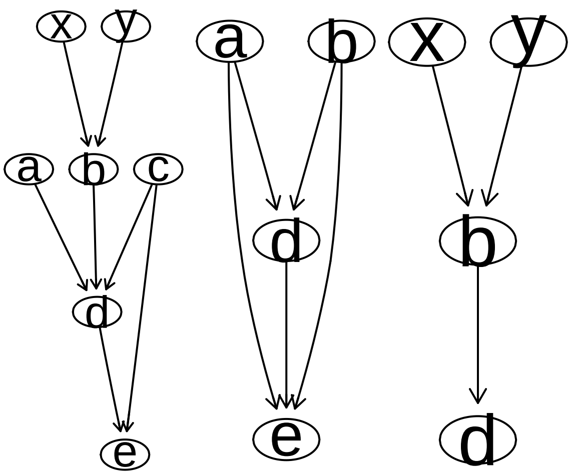


Figure 2: Underlying DAG for a data-generating process on the left; In the center, x,y, and c are marginalized out. It is not the case that $a \perp\!\!\!\perp e|d$, since a is not d-seperated from c by d, which is a direct cause of e, and the causal effect of a on e is apparent, but the presence of mediating or confounding variables is not. On the right, a, c, and e have been marginalized out.

Consider the following equations, which represent a model for which the graph in the left in Figure 2 is the true underlying DAG. U denotes a $U(0,1)$ random variable; Z is a standard normal random variable. There is nothing special about the parameters; I just chose some that made the equations nonlinear, but I did not try to make the conditional independence tests very challenging. In each case assume the random variables U and Z are drawn independently of the others.

$$\begin{aligned}
x &= Z_1/\sqrt{3} && (1)\\
y &= Z_2/\sqrt{3} && (2)\\
a &= U_1 - 0.5 && (3)\\
b &= x + y + Z_3/\sqrt{3} && (4)\\
c &= U_2 - 0.5 && (5)\\
d &= a + 0.08b^3 - 0.3b + 2|c| + (2U_3 - 1)/20 && (6)\\
e &= 0.3c^2 + 0.1d + Z_4/2 && (7)
\end{aligned}$$

I draw two samples, each of size 10000, independently from this system. In the first, I keep the values for a, c, d, and e. In the second I keep the values for x, y, b, and d. I estimate the PAGs for each using the RFCI algorithm from the *pcalg* package in R and the *hsic.gamma* conditional independence test from the *kpcalg* package. Figue 3 has the PAG for the observed variables $\{a, c, d, e\}$.

Here are the pairwise plots of the observed variables.

Figure 5 has the PAG for the observed variables in the second dataset.

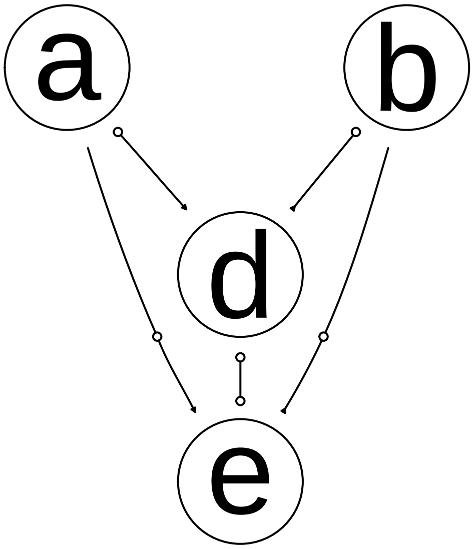


Figure 3: Estimated PAG for the simulated data from the equations above. I used the *rfci* algorithm, implemented in the *pcalg* package in R, using the *hsic.gamma* test of conditional independence from the *kpcalg* package.

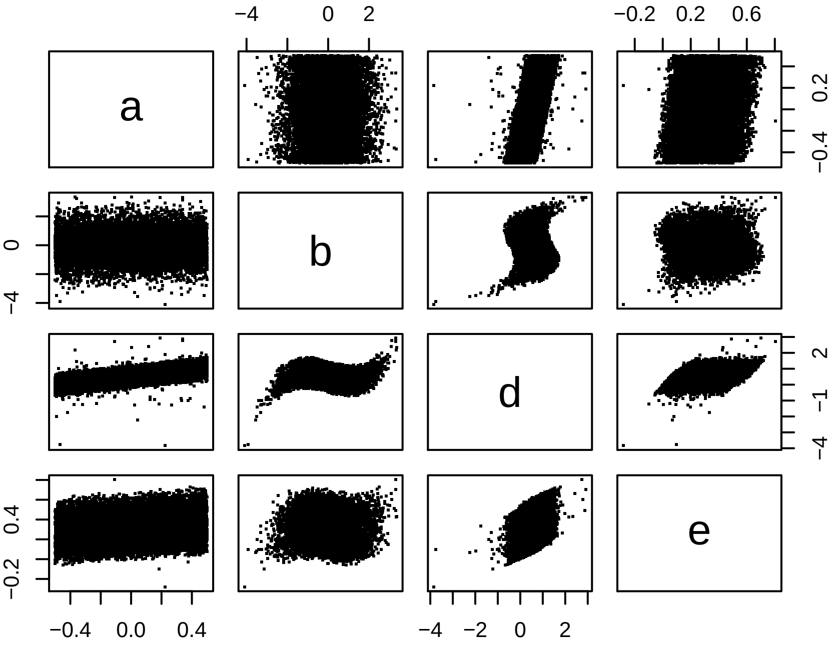


Figure 4: Pairwise scatterplots of the data generates from the functions in equations 1 through 5 above.

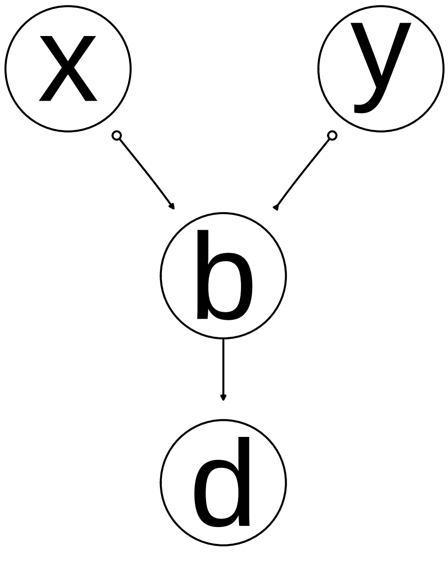


Figure 5: Estimated PAG for the simulated data from the equations above. I used the *rfci* algorithm, implemented in the *pcalg* package in R, using the *hsic.gamma* test of conditional independence from the *kpcalg* package.

Combining the results from models on overlapping sets of variables

```
##    a  b d e  x  y
## a  0  0 0 0  0  0
## b  0  0 1 0 -1 -1
## d -1 -1 0 0 -1  1
## e -1 -1 0 0 -1 -1
## x  0  0 0 0  0  0
## y  0  0 0 0  0  0
```

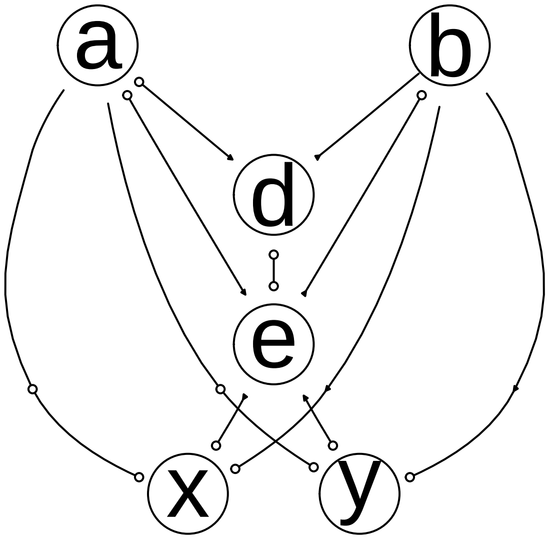


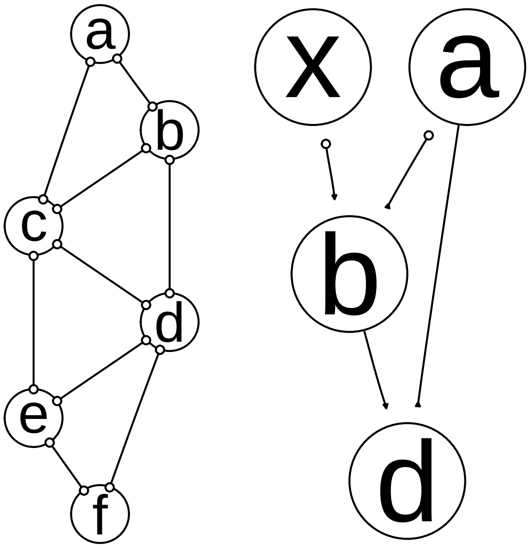
Figure 6: Combination of graphs from figures above using the MCI algorithm

This seems to be an I-map of the original model, but not a perfect one. We see, for example, that $a \perp\!\!\!\perp x|Z$

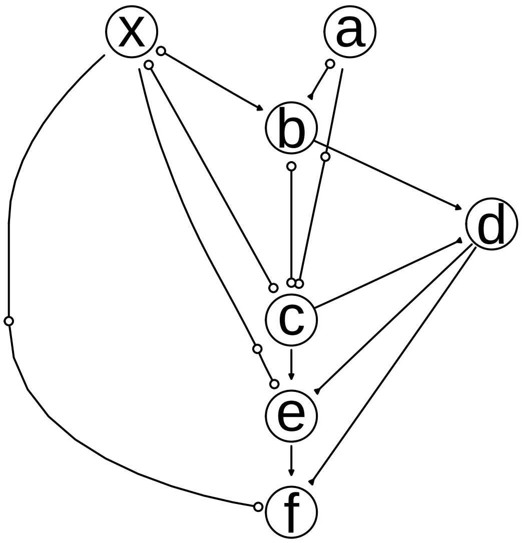
Example with MCI

```
##    x  a  b d c e f
## x  0  0  0 0 0 0 0
## a  0  0  0 1 0 0 0
## b -1 -1  0 1 0 1 1
## d -1 -1 -1 0 0 1 1
## c  0  0  0 1 0 1 1
## e  0  0  0 0 0 0 1
## f  0  0  0 0 0 0 0
```

Figure **??** shows two PAGs from different sets of simulated variables from the same overall model. On the left, the PAG is consistent with a triangulated undirected PGM, where there is no causal information. On the right, A causal model on four variables. We add one variable, x.



Using the MCI algorithm, we get PAG in Figure **??**. By adding the variable from the the second data set, we fully directed two arcs, $d \to f$ and $d \to e$. We also partially directed two arcs, indicating c does not cause e.



#Selection Bias

Consider the following underlying DAG. It is the same as in 2 on the right, but I'm using a different plotting function.

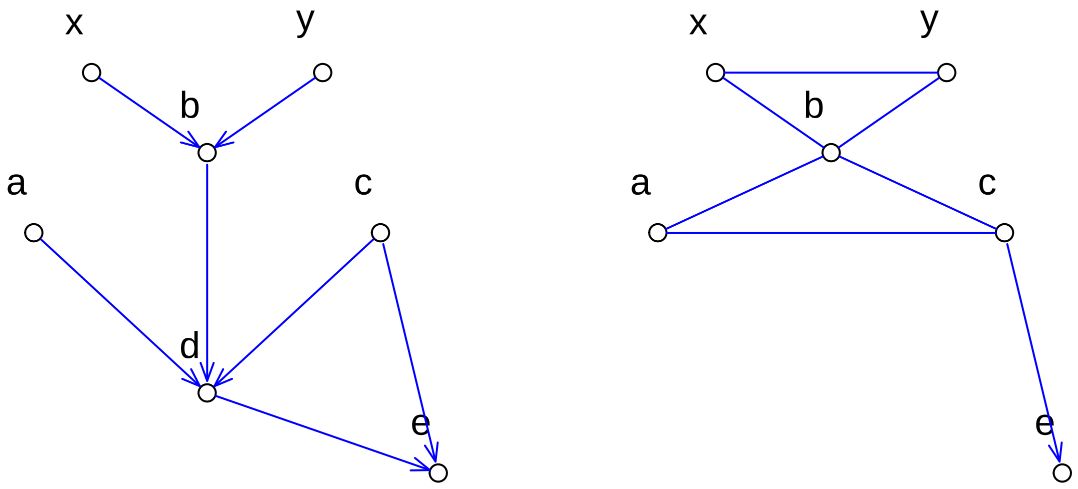


Figure 7: A DAG representing and underying causal DAG on the left. On the right, the MAG when we condition on the variable d.

The relationships between the other variables has really been muddied; b still separates x and y from the other variables, but, otherwise, the MAG has little resemblance to the DAG. Only one causal arrow is identifiable, but the causal effect would not be. If we can get $P(S = 1|V - S)$, we can do the weighting and fix the problem. But ignore sampling weihgts at your peril.

It's also interesting. Say we drew the sample with the knowledge d is always true or something. All it ancestors have "–" lines, so are causal ancestors of d.

Causal Models as Mininal PGMs and Imputation

I think there is value in the PAG as a description of the dependencies among variables, even if we can't get estimate the casual effects. The PAG keeps more information than is retained if we estimate either the BN or the UG models, as the BN and the UG models are special cases of the PAG; they are in fact special cases of the MAG, which subsumes those simpler graphical models. See (J. Zhang 2008b)

If there is an underlying DAG as described above, the methods we have described find, in the large sample, a PAG representing the class of faithful Markov MAGs of the indepencdence relationships of the corresponding joint distribution. Since the graphs are Markov with respect to that distribution, every *m-separation* relationship in the graph represents an independence relationship in the joint distribution. Since it is faithful, there are no edges in the graph between varaibles not in each others' markov blankets. The PGMs derived this way have the minimal numers of edges while remaining I-Maps of the distributions of the observed variables. Models discovered this way have these desireable characteristics without regard to the causal interpretation.

This is a desirable property when we do imputations; modeling either to many or too few CI relationships may contibute to errors in an analysis like a microsimulation. These considerations are in tension with a desire for speedy computations. Once a decomposable model has been constucted froma a PAG or MAG, the conditional distribution can be computed quickly at the cost of extra edges in the graph and extra members for some variables' Markov blankets. Consider the example if Figure 8. The underlying DG is on the left; in the center it had been *moralized* on the right, it has been *triangulated*, whcih yield the decoposable model. This is the easy one to work with, and it's the version we use, for example, on the state weights. The fast *sum-product* algorithm converges in one iteration on these models, and the sufficient statitics are functions of the variables in each clique, leading to low(er)-dimensional problems. While the sum-product algrotihm can be applied to the middle model, it may take a while to converge or may not converge and, if it does converge, may not converge to the right conditional probability. It 'usually' does, but it's easy to see the advantage of the better-behaved one. As an aside, other algorithms may be used, like Monte Carlo methods, but they are slow.

The other side of this is that, in the center, the MB of X_5 is $\{X_3, X_7\}$. In the decomposable model on the right, it is $\{X_1, X_3, X_4, X_6, X_7\}$. If we could, we would like to use the MB for the model in the center to avoid introducing artifacts. I don't know how to do this in general, though, without falling back to slow methods. In special cases, where we only care about one variable, we could just just take it and its MB out of the rest of the model and consider those variables in isolation. In Figure 8, if we just want to impute X_5, we could just condition on $\{X_3, X_7\}$. If we want to impute different variables for different rows in a data set, we might like to be more effifient and use the mode on the right.

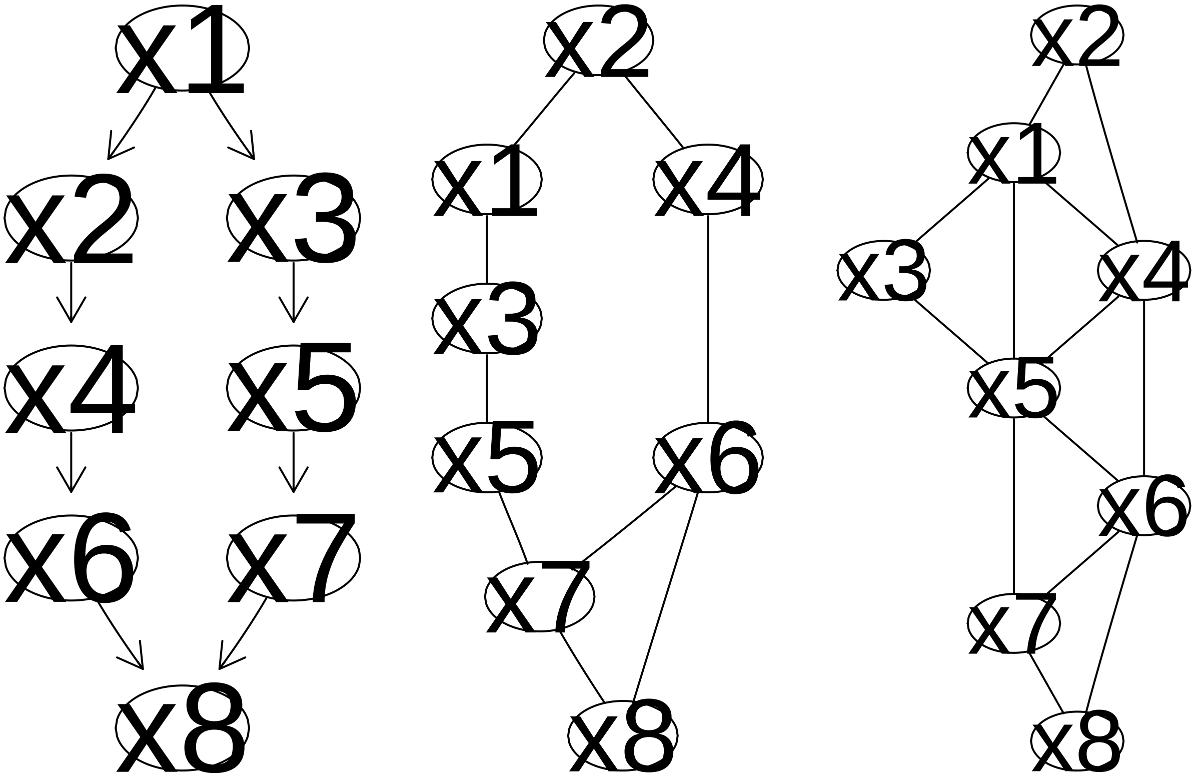


Figure 8:

There is another question I have not resolved. Consider a clique in the graph on the right, say $\{X_1, X_3, X_5\}$. The joint distribution of that is factorable as $P(x_1, x_3, x_5) = \phi(x_1, x_3)\phi(x_3, x_5)$, and we could specify it that way when the time comes to parameterize the model in the decomposable version. I don't know if that independence stucture is maintained when we find probabilities conditioned, for example, X_8. I' don't think so, because we need to calcualte $P(x_1, x_5)$, as the separator between $\{X_1, X_3, X_5\}$ and $\{X_1, X_4, X_5\}$ as the marginalization over X_3 and X_4, so I'm not clear on this part.

Multivariate Gaussian Distributions

I know I have been resistant to this kind of modeling assumption, but it is so common that I think it is worth some discussion. In the MultiVariate Normal (MVN) case, estimation and fitting is very easy. The sufficient statistic for fitting the PAG is the pair containing the sample size and the correlation matrix, $< n, C >$. The following illustrates the

```
## Time difference of 0.01673698 secs
```

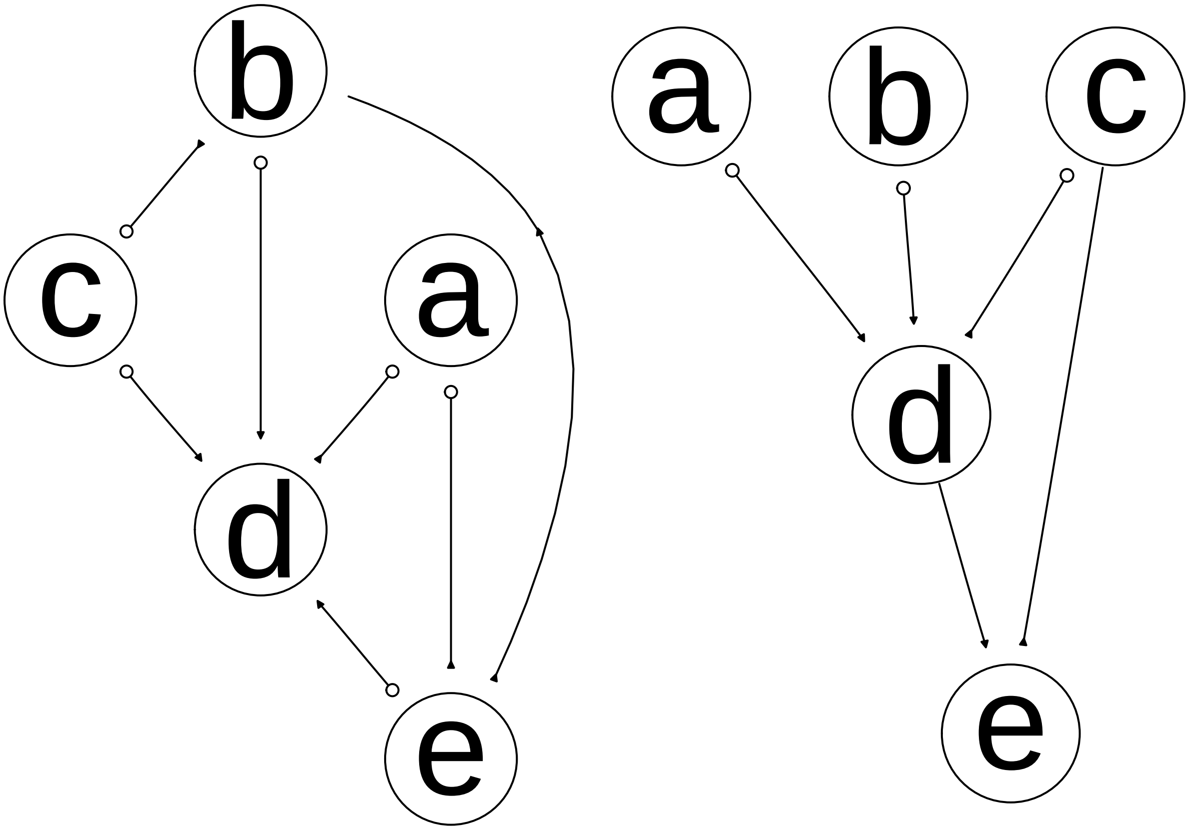


Figure 9: Causal Discovery Algrithm RFCI applied to Simulated data. We use the same data as above, generated from equations (1)-(7), not observing X or Y. On the left, we assume the data was generated from a MVN distribution; the estimated PAG is somewhat different from the true PAG. The estimation process took 0.03 secondson y laptop. On the right, we used a nonparametric set of CI tests in the RFCI algorithm. We got the true PAG, but the algorthm used more then 14 minutes on my laptop.

The graph on the right of Figure 9 is

Applications in Economics

(Moneta 2018) uses the PC algorithm on a data set of four variables, to investigate the existence and direction of causation between firm performance and exports. It's a suprisingly simple data set. He also noes that many economics graduate students are aware of these methods, but they are applied rarely. The data is derived from two years of longitudinal data on firms; the variables are changes in variables. *Expand this*

Examples

more to include

(Strobl 2018),*A Constraint-Based Algorithm For Causal Discovery with Cycles, Latent Variables and Selection Bias*

(Loftus et al. 2018), *Causal Reasoning for Algorithmic Fairness*

sokolova2016computing]

- Canonical DAGS for MAGS
  - MAGs, PAGs, Canonical DAGs, factor graphs
  - Do factor graphs have the necessary independence information? Is it generally possible to make a faithful factor graph? if so, could we fit a factor graph then use a separation criterion for the consraint-based test
- I would like to use a bootstrap or simulation method to test a result for a method in a given problem. Is there theory saying whether this is a posibility?
- How do we pick significance levels for costraint-based methods in finite data?
- Survey Data
-

References

Claassen, Tom, and Tom Heskes. 2010. "Causal Discovery in Multiple Models from Different Experiments." In *Advances in Neural Information Processing Systems*, 415–23.

Colombo, Diego, Marloes H Maathuis, Markus Kalisch, and Thomas S Richaerdson. 2012. "Learning High-Demensional Directed Acyclic Graphs with Latent and Selection Variables." *The Annls of Statistics*. JSTOR, 294–321.

Kalisch, Markus, Martin Mächler, Diego Colombo, Marloes H Maathuis, Peter Bühlmann, and others. 2012. "Causal Inference Using Graphical Models with the R Package Pcalg." *Journal of Statistical Software* 47 (11): 1–26.

Loftus, Joshua R, Chris Russell, Matt J Kusner, and Ricardo Silva. 2018. "Causal Reasoning for Algorithmic Fairness." *ArXiv Preprint ArXiv:1805.05859*.

Mani, Subramani, Gregory F Cooper, and A Statnikov. 2006. "Causal Discovery Algorithms Based on Y Structures." In *NIPS 2006 Workshop on Causality and Feature Selection*. Citeseer.

Mani, Subramani, Peter L Spirtes, and Gregory F Cooper. 2012. "A Theoretical Study of Y Structures for

Causal Discovery." *ArXiv Preprint ArXiv:1206.6853*.

Moneta, Alessio. 2018. *Causal Model Search Applied to Economics: Gains, Pitfalls and Challenges*. https://www.youtube.com/watch?v=Yn3IBcjdiFk&t=1633s.

Pearl, Judea. 2009. *Causality*. Cambridge university press.

Perković, Emilija, Johannes Textor, Markus Kalisch, and Marloes H Maathuis. 2015. "A Complete Generalized Adjustment Criterion." *ArXiv Preprint ArXiv:1507.01524*.

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Richardson, Thomas, Peter Spirtes, and others. 2002. "Ancestral Graph Markov Models." *The Annals of Statistics* 30 (4). Institute of Mathematical Statistics: 962–1030.

Schulman, Leonard J, and Piyush Srivastava. 2016. "Stability of Causal Inference."

Strobl, Eric V. 2018. "A Constraint-Based Algorithm for Causal Discovery with Cycles, Latent Variables and Selection Bias." *ArXiv Preprint ArXiv:1805.02087*.

Strobl, Eric V, Kun Zhang, and Shyam Visweswaran. 2017. "Approximate Kernel-Based Conditional Independence Tests for Fast Non-Parametric Causal Discovery." *ArXiv Preprint ArXiv:1702.03877*.

Verbyla, Petras, Nina Ines Bertille Desgranges, and Lorenz Wernisch. 2017. *Kpcalg: Kernel Pc Algorithm for Causal Structure Detection*. https://CRAN.R-project.org/package=kpcalg.

Zhang, Jiji. 2008a. "Causal Reasoning with Ancestral Graphs." *Journal of Machine Learning Research* 9 (Jul): 1437–74.

———. 2008b. "On the Completeness of Orientation Rules for Causal Discovery in the Presence of Latent Confounders and Selection Bias." *Artificial Intelligence* 172 (16-17). Elsevier: 1873–96.