# An Algorithm for Fast Recovery of Sparse Causal Graphs

**Peter Spirtes and Clark Glymour**

Previous asymptotically correct algorithms for recovering causal structure from sample probabilities have been limited even in sparse causal graphs to a few variables. We describe an asymptotically correct algorithm whose complexity for fixed graph connectivity increases polynomially in the number of vertices, and may in practice recover sparse graphs with several hundred variables. From sample data with $n$ = 20,000, an implementation of the algorithm on a DECStation 3100 recovers the edges in a linear version of the ALARM network with 37 vertices and 46 edges. Fewer than 8% of the undirected edges are incorrectly identified in the output. Without prior ordering information, the program also determines the direction of edges for the ALARM graph with an error rate of 14%. Processing time is less than 10 seconds. *Keywords*: DAGS, Causal Modelling.

Finding the causal relations between variables is necessary for both scientific explanation and policy making. For these purposes, it is insufficient to merely fit an empirical covariance matrix or find the best least squares linear estimator of a variable. The policy implications of empirical data can be completely reversed by alternative hypotheses about the causal relations of variables; furthermore, the estimates of a particular causal influence can be radically altered by changes in the assumptions made about other dependencies.[1] For these reasons, it is often the case that the aim of empirical research in the social sciences is to determine the causal relations among a set of variables, and to estimate the relative importance of various causal factors. Even when this aim is not explicitly acknowledged, it is often the tacit aim. Therefore, the question of how causal relations among variables can be discovered is of primary importance for social science research.

*Social Science Computer Review* 9:1, Spring 1991.  CCC 0894-4393/91/$1.50.

## The Difficulty of the Discovery Problem

Part of the difficulty in finding good causal models is due to the sheer number of possible causal models for a given set of variables. We will represent the direct causal dependence of one variable on another by a directed edge from a vertex representing the causal variable to a vertex representing the effect variable. Then the number of possible causal structures on $n$ variables is the number of directed graphs with $n$ vertices, or $4^{\binom{n}{2}}$. If causal cycles are forbidden, then the number of possible causal structures on $n$ variables is the number of acyclic directed graphs on $n$ variables. For 12 variables the number of directed graphs is approximately $5.4 \times 10^{39} \times$ and the number of acyclic graphs is 521,939,651,343,829,405,020,504,063 (Harary & Palmer, 1973). When the time order of the variables is also known, so that causal hypotheses in which later variables cause earlier variables can be eliminated, the number of alternatives remaining is still huge: for 12 variables, it is $7.4 \times 10^{19}$.

The social scientist who addresses a problem area where causal questions are of concern must therefore restrict the space of alternatives. There are three obvious avenues for restricting the space of alternatives: (1) use experimental controls, (2) use prior knowledge, and (3) use features of the sample data.

Experimental procedures for addressing social questions are often desirable but impractical. They are very expensive, and where quasi-experiments that control some variables but not others are used, the number of alternative causal structures possible a priori may remain very large. Methodology texts routinely recommend generating the set of admissible causal structures from "substantive theory" (see Joreskog & Sorbom, 1984; Duncan, 1975). The actual practice of publications in the social science literature is usually to restrict the number of alternatives considered to a very few; the restrictions are often justified by citing prior literature or by appealing to very broad theoretical frameworks. There is no evidence, however, that such appeals constitute a reliable discovery procedure. It seems at least as likely that appeals to theory introduce bias and often exclude the true causal relations among the variables of interest. What about the third avenue?

## Causal Inference from Statistical Samples

Statisticians routinely use sample data in systematic ways for parameter estimation in a parameterized family of probability distributions. However, they more rarely use sample data to explicitly or systematically infer causal structure. Indeed, methodologists often warn against using sample data to make causal inferences, and they routinely recommend that "substantive knowledge" rather than sample data should determine the causal structure of a model. Procedures

that use sample data are denounced as "data mining" or "data ransacking." Finding a textbook on statistical methodology for the social sciences that does not include these warnings would be difficult (see Loehlin, 1987; James, Mulaik, & Brett, 1982).

Despite the fervor of the denunciation of causal inference based on sample data, it is difficult to find any sober analysis that justifies the conviction that reliable inference of this kind is impossible. It is true that in the worst case, if the sample size is small compared to the number of variables, data-based inference will be unreliable. This can be avoided, however, by appropriate sampling. In addition, social scientists have experience with a number of exploratory factor analysis programs which are commonly judged to be quite unreliable. But part of the reason for the unreliability of these factor analysis programs in the contexts in which they are used is that they make very specific assumptions that are false in many domains. Among the assumptions made by many factor analytic programs is that the data functional dependencies between variables are linear, and that no measured variable directly causes either measured or latent variables (Loehlin, 1987). Each of these assumptions may be false in a given domain, but they are not essential to inferring causal structure from sample data.

The best way to show that the complaint against sample-based causal inference is simply an unfounded prejudice is to provide reliable procedures for using sample data to usefully narrow the class of causal structures that are, a priori, possible for the data, and to prove that the procedures are reliable. That is our aim.

### Recovering Causal Relations

Consider pairs $(g,P)$ for which $g$ is a directed acyclic graph and $P$ is a probability distribution on the vertices of $g$ such that (1) for every vertex $v$ and every set $S_v$ of vertices that are not descendants or parents of $v$, $v$ and $S_v$ are independent conditional on the parents of $v$; and (2) every independence relation in $P$ is a consequence of the independence relations in (1). Pairs satisfying these conditions can be viewed as causal structures in which the causal dependencies generate statistical dependencies. When the set of measured variables for which probabilities are provided in the data is such that every common cause of a measured variable is itself measured, we say the structure is causally sufficient.

Recovery problems occur when determining $g$, or features of $g$, from the distribution $P$ or from samples obtained from $P$. Spirtes, Glymour, and Scheines (1990) proposed the following SGS algorithm for the recovery problem with causally sufficient structures, using as input independence and conditional independence facts about $P$:[2]

1. Start with the complete undirected graph.
2. For each vertex pair $(a,b)$, remove the undirected edge between $a$

and $b$ if and only if $I(a,S,b)$ for some subset $S$ not containing $a$ or $b$. Call this undirected graph $G$.

3. For each triple $(a,b,c)$ of vertices such that $a$ and $b$ are adjacent in $G$, $b$ and $c$ are adjacent in $G$, and $a$ and $c$ are not adjacent in $G$, direct the edges $a$-$b$ and $b$-$c$ into $b$ if and only if for every set $S$ of vertices containing $b$ but not $a$ or $c$, $\sim I(a,S,c)$.
4. Output all orientations of the graph consistent with (2).

Verma and Pearl (1990) subsequently proved the correctness of the algorithm and offered a variant that outputs a pattern rather than a collection of graphs. The pattern has an undirected edge between two vertices if the SGS output contains graphs that orient the edge in different directions; the pattern contains a directed edge if every graph output by the SGS algorithm has the edge so oriented; and the pattern may have a bidirected edge: for example, $a \leftrightarrow b$ provided (2) determines that the $a$-$b$ edge collides with another edge at $a$ and also collides with another edge at $b$. When all common causes are measured and the data consist of the actual independence and conditional independence relations, the pattern is simply a representation of the class output by the SGS algorithm; but when there are unmeasured common causes or independence facts due to sampling variation rather than to $P$, the pattern is more general.

Two graphs $(g,g')$ are statistically indistinguishable provided that for every probability distribution $P$, $(g,P)$ satisfies the conditions (1) and (2) of the first paragraph if and only if $(g',P)$ does. From the independence facts of a distribution $P$ such that $(g,P)$ satisfies (1), and (2), the SGS algorithm returns all and only the graphs statistically indistinguishable from $g$.

In the worst case, the SGS algorithm requires a number of conditional independence facts that increase exponentially with the number of vertices, as must any algorithm based on conditional independence relations. But because for any undirected edge that is in the graph $g$, the number of conditional independence facts that must be generated and checked in stage (1) of the algorithm is unaffected by the connectivity of the true graph, even for sparse graphs, the algorithm rapidly becomes computationally infeasible as the number of vertices increases. Besides problems of computational feasibility, the algorithm has problems of reliability when applied to sample data. The determination of higher-order conditional independence relations from sample distributions is generally less reliable than is the determination of lower-order independence relations. With, say, 37 binary variables, the determination of the conditional independence of two variables on the set of all remaining variables requires considering the relations among the frequencies of $2^{35}$ distinct states, only a tiny fraction of which will be instantiated even in very large samples.

To illustrate the difficulty of recovering the graph $g$ (or a set of equivalent graphs) from the probability distribution $P$, consider an

example given by Herskovits and Cooper (1990). Their Kutató Algorithm is a heuristic entropy minimization procedure for recovering a directed graph given sample data and a total ordering of the vertices such that $v_1 > v_2$ implies that there is no directed edge from $v_2$ to $v_1$. The asymptotic reliability of the procedure is unknown. Nonetheless, from large sample data the algorithm recovers most of the connections on a sparse graph—the ALARM network (Beinlich, Suermondt, Chavez, & Cooper, 1989)—with 37 variables and 46 edges. In their example, the direction of the edges is not recovered from the data, but is determined by the prior ordering given to the computer (see Figure 1).[3]

Using 10,000 cases, an implementation on a Macintosh II required about 22½ hours, about a quarter of which was required to read the database. The output omitted two correct edges and included two false edges. By comparison, the SGS algorithm has been implemented in the *Tetrad II* program using partial correlation tests for conditional independence. *Tetrad II* is an experimental program for recovering causal structure from statistical data developed by the authors at Carnegie-Mellon University. The module of *Tetrad II* that implements the SGS algorithm takes a covariance matrix and any background causal information that the user has as input, and outputs a set of causal graphs compatible with the background knowledge that explain the conditional independencies true of the covariance matrix. The background knowledge can include information about casual relations among variables that are known to exist, causal relations that are known not to exist, and temporal information. We will implement tests for conditional independence relations that do not depend upon the assumption of linearity. We plan to make commercial versions of the program available by the end of the year that run on UNIX workstations or MS-DOS personal computers.

Run on a DEC workstation with 20MB RAM, the procedure stops at about 17 variables because of space requirements for storing the conditional independence facts. Space could be traded for time, but the ALARM case is "out of sight."

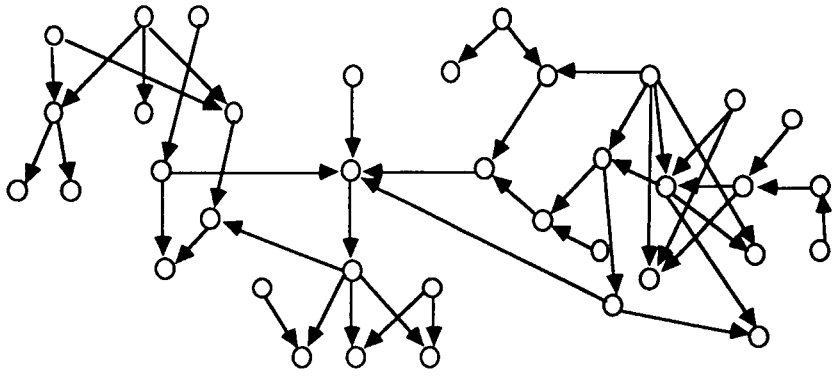

Figure 1 ALARM network

Verma and Pearl (1990) have suggested an improvement on the SGS algorithm. For each pair of variables $(a,b)$, introduce an undirected edge between them if they are dependent conditional on the set of all other variables. Call the resulting network $N$. (In the true graph, $G$, the parents of any variable form a maximal complete subgraph—a clique—in the network $N$.) Again, for each pair of variables $(a,b)$ adjacent in $N$, determine if $(a,b)$ are dependent conditional on all subsets of variables in the cliques in $N$ containing $a$ or $b$. If so, $a$ is adjacent to $b$ in $G$. The complexity is thus bounded by the size of the largest clique in $N$.

The practical value of the improvement is limited by the fact that it is still necessary to judge conditional independence relations of the order of the number of vertices of the graph (minus two). In the linear case, this may be possible. Discrete data judgements about such conditional independence relations are quite unreliable, however, since the great majority of the corresponding states will not be instantiated in the data.

We would like an algorithm that has the same input/output relations as the SGS procedure but for sparse graphs does not require the determination of higher-order independence relations, and in any case requires as few conditional independence relations as possible. The following procedure starts by forming the complete, undirected graph, then "thins" that graph by removing edges with zero-order conditional independence relations, thins again with first-order conditional independence relations, and so on. The set of variables conditioned on need only be a subset of the set of variables adjacent to one or the other of the variables conditioned, and can even be confined to adjacent variables on certain undirected paths.

### PC Algorithm

Let $A_C ab$ denote the set of vertices adjacent to $a$ or to $b$ in graph $C$, except for $a$ and $b$ themselves. Let $U_C ab$ denote the set of vertices in graph $C$ on (acyclic) undirected paths between $a$ and $b$, except for $a$ and $b$ themselves. (Since the algorithm is continually updating $C$, $A_C ab$ and $U_C ab$ are constantly changing as the algorithm progresses.)

A. Form the complete undirected graph $C$ on the vertex set $V$.
B. $n = 0$.
   repeat
      For each pair of variables $(a,b)$ adjacent in $C$, if $A_C ab \cap U_C ab$ has cardinality greater than or equal to $n$ and $a$, $b$ are independent conditional on any subsets of $A_C ab \cap U_C ab$ of cardinality $n$, delete $a$-$b$ from $C$.
      $n=n+1$.
   until for each pair of adjacent vertices $a$, $b$, $A_C ab \cap U_C ab$ is of cardinality less than $n$.
   Call the resulting undirected graph $F$.
C. For each triple of vertices $(a,b,c)$ such that the pair $(a,b)$ and the

pair $(b,c)$ are each adjacent in $F$ but the pair $(a,c)$ are not adjacent in $F$, orient $a$-$b$-$c$ as $a \rightarrow b \leftarrow C$ if and only if $a$ and $c$ are dependent on every subset of $A_F ac \cap U_F ac$ containing $b$. Output all graphs consistent with these orientations.

Note that $A_G ab \cap U_G ab$ is not in general the set of *parents* of $a$ or $b$ (in the oriented graph) on undirected paths between $(a,b)$, since descendents of $(a,b)$ may also occur.

An obvious modification of the algorithm will generate patterns rather than collections of graphs.

The complexity of the algorithm for a graph $G$ is bounded by max $(|A_G ab|)$ over all pairs of vertices $(a,b)$, which is never more than the sum of the two largest degrees in $G$. Generally stage $B$ of the algorithm continues testing for some steps after the correct undirected graph has been identified. The number of steps required before the true graph is found (but not necessarily until the algorithm halts) depends on the maximal number of treks[4] between a pair of variables, say $(a,b)$, that share no vertices adjacent to $a$ or $b$. If these maximal numbers are held constant as the number of vertices increases, so that $k$, the maximal order of the conditional independence relations that need to be tested, does not change, then the worst case computational demands of the algorithm are bounded by $n^3$. It should be possible to recover sparse graphs with as many as several hundred variables. Of course the computational requirements increase exponentially with $k$.

In many cases it may be more efficient to perform conditional independence tests on all subsets of $A_G ab$ rather than to compute $U_G ab$. We have not yet theoretically determined the trade-off.

The structure of the algorithm and the fact that it continues to test even after having found the correct graph suggest a natural heuristic for very large variable sets whose causal connections are expected to be sparse, namely to set a fixed bound on the order of conditional independence relations that will be considered.

*Proposition: The PC and SGS algorithms give the same output.*

Proof: Let $P_G ab$ denote the set of vertices in directed graph $G$ that are parents of $a$ or of $b$, except for $a$ and $b$ themselves. We note a lemma.

Lemma: In any pair $(G,P)$ meeting conditions (1) and (2), if vertices $(a,b)$ are not adjacent then they are independent conditional on $P_G ab \cap U_G ab$.

The proof is a trivial modification of the argument Verma and Pearl (1990, pp. 221–222) give for their Lemma 1.

Now we show that steps (A) and (B) of the PC algorithm produce the correct undirected graph. Let $G$ be a directed graph produced by the SGS algorithm. (Every graph produced by the SGS algorithm

shares the same underlying undirected graph.) First, we will show that every edge in the undirected graph of $G$ is also in the undirected graph $C$ at every stage of construction. The PC algorithm starts with a complete graph, and only removes an edge between $a$ and $b$ if $a$ and $b$ are independent on some subset of $A_C ab \cap U_C ab$. However, if the edge between $a$ and $b$ is in the undirected graph of $G$, then $a$ and $b$ are not independent on any subset of variables not containing $a$ or $b$. Hence, every edge in the undirected graph of $G$ is also in $F$, the final graph produced by the PC algorithm.

We must now show that if $a$ and $b$ are not adjacent in the undirected graph of $G$, then $a$ and $b$ are not adjacent in $F$. If $a$ and $b$ are not adjacent in $G$, then $a$ and $b$ are independent on some subset of variables not containing $a$ or $b$. By the lemma, then, $a$ and $b$ are independent conditional on the set $P_G ab \cap U_G ba$. Since every edge in the undirected graph of $G$ is in $C$, $P_G ab \cap U_G ba$ is a subset of $A_C ab \cap U_C ab$, and hence $a$ and $b$ are independent conditional on some subset of $A_C ab \cap U_C ab$.

It remains only to show that step $C$ of the algorithm orients the graph correctly. Assume that in $G$, $\{a,c\}$ are not adjacent but $a$ is adjacent to $b$ and $b$ is adjacent to $c$. In $G$, the $a$-$b$ and $b$-$c$ edges collide at $b$ if and only if there is no set $S$ containing $b$ and not $a$ or $c$ such that $\{a,c\}$ are independent conditional on $S$. Since $\{a,c\}$ are not adjacent in $G$, they are independent conditional on the set $P_G ac \cap U_G ac$. If the edges in $G$ do not collide at $b$, then $b$ is a parent of $a$ or of $c$, so $b$ is in $P_G ac \cap U_G ac$, which is a subset of $A_F ac \cap U_F ac$ containing $b$. If the edges do collide at $b$ in $G$, then $\{a,c\}$ are dependent on every set containing $b$ and not $\{a,c\}$, and hence dependent on every subset of $A_F ac \cap U_F ac$ that contains $b$.

### An Application of the PC Algorithm

We have applied the PC algorithm to a linear version of the ALARM network. Using the same directed graph, linear coefficients with values between 0.5 and 1.0 were randomly assigned to each directed edge in the graph. Using a joint normal distribution on the variables of zero indegree, three sets of simulated data were generated, each with a sample size of 20,000. The covariance matrix and sample size were given to a version of the *Tetrad II* program with an implementation of the PC algorithm. This implementation takes as input a covariance matrix, and it outputs a pattern. It does not check to determine whether variables adjacent to vertices $\{v_1, v_2\}$ lie on an undirected path between $v_1$ and $v_2$. No information about the orientation of the variables was given to the program. Run on a DECstation 3100, for each data set the program required less than 10 seconds to return a pattern. In each trial, the output pattern omitted three edges in the ALARM network. Of the remaining 43 edges, the orientation of 3 of them is indeterminable in principle from the probabilities, and in the first two trials the program so reported, while in the third it or-

Table 1

| | Omitted undirected edges | False undirected edges | Orientation errors |
|---|---|---|---|
| Trial 1 | 3 | 0 | 5 |
| Trial 2 | 3 | 0 | 6 |
| Trial 3 | 3 | 2 | 7 |

ienting one of the three. Of the remaining 40 edges, the trials misoriented 5, 6, and 7 edges respectively, always by judging that an edge was directed into *both* of its vertices (as the pattern output allows) when in the ALARM graph it is directed into only one. The results are summarized in Table 1.

The implementation used did not determine the adjacency sets lying on undirected paths between two variables because in this case with correlation data it was computationally cheaper to determine the partial correlations for all subsets of $A_C ab$ than to keep track of $A_C ab \cap U_C ab$. With discrete count data for which the determination of conditional independence relations is more computationally demanding, the alternative procedure described in our statement of the algorithm might be faster. For example, for one pair of vertices in the network, $Aab$ consists of 8 vertices while $Aab \cap Uab$ consists of only 2 vertices.

The comparison of 10 seconds for the PC algorithm with 22½ hours for the Kutató algorithm should not be taken as a direct comparison of the efficiencies of the algorithms, since the DECStation 3100 is much faster than a Macintosh, and without the assumption of linearity considerably more time would be required in numerical operations to determine conditional independence.[5] Nonetheless, the PC algorithm appears to be very fast and reliable for sparse graphs. For similar data from a similarly connected graph with 100 variables, the present implementation should require less than 2 minutes.

## Appendix

Fung and Crawford (1990) have independently proposed an algorithm similar in spirit to the PC algorithm for constructing undirected graphs.

In addition, Pearl and Verma (1990) describe an algorithm that shows how step (C) of the PC algorithm can be improved in the following way (which also requires a slight modification to step (B):

B.

$n = 0$.

repeat

For each pair of variables $(a,b)$ adjacent in $C$, if $A_C ab \cap U_C ab$ has cardinality greater than or equal to $n$ and $(a,b)$ are independent condi-

tional on some set $S(a,b)$ that is a subset of $A_C ab \cap U_C ab$ of cardinality $n$, delete $a$-$b$ from $C$, and record $S(a,b)$.
$n = n + 1$.

until for each pair of vertices $(a,b)$ adjacent in $C$, $A_C ab \cap U_C ab$ is of cardinality less than $n$.

C. Let $F$ be the graph resulting from step (B). For each triple of vertices $(a,b,c)$ such that the pair $(a,b)$ and the pair $(b,c)$ are each adjacent in $F$ but the pair $(a,c)$ are not adjacent in $F$, orient $a$-$b$-$c$ as $a \rightarrow b \leftarrow c$ if and only if $b$ is not in $S(a,c)$.

Output all graphs consistent with these orientations.

## Notes

Peter Spirtes and Clark Glymour, Department of Philosophy, Carnegie-Mellon University, Pittsburgh, PA 15217. We thank Gregory Cooper for a conversation that stimulated this work.

1. See our discussion of the causal relations between foreign capital on political repression in Glymour (1987).

2. We denote by "$I(a,S,b)$" the claim that variables $a$ and $b$ are independent conditional on the set of variables in $S$, and by "$\neg I(a,S,b)$ the denial of that claim.

3. Herskovits and Cooper say that a variant of the Kutató algorithm can determine the orientation of edges without a prior ordering of the variables, but they do not describe the properties of the application or give an example. They are also investigating Bayesian alternatives that are much faster than the Kutató procedure.

4. A trek is a pair of directed paths from some vertex $z$ to $(a, b)$ respectively, intersecting only at $z$, or a directed path from $a$ to $b$ or a directed path from $b$ to $a$.

5. It may in fact be the case that for large samples and variable sets the errors introduced by assessing conditional independence through partial correlations or other aggregate measures are adequately repaid in time savings.

## References

Beinlich, I., Suermondt, H., Chavez, R., & Cooper, G. (1989). *The ALARM monitoring system.* (Technical Report KSL 88-84). Stanford, CA: Knowledge Systems Laboratory, Medical Computer Science, Stanford University.

Duncan, O. D. (1975). *Introduction to structural equation models.* New York: Academic Press.

Fung, R., & Crawford, S. (1990). Constructor: A system for the induction of probabilistic models. In *Proceedings of the American Association for Artificial Intelligence* (pp. 762-769). Boston.

Herskovits, E., & Cooper, G. (1990). Kutató: An entropy-driven system for construction of probabilistic expert systems from databases. In *Proceedings of the Sixth Conference on Uncertainty in AI* (pp. 54–62). Cambridge, MA.

James, L. R., Mulaik, S. A., & Brett, J. (1982). *Causal analysis.* Newbury Park, CA: Sage.

Joreskog, K., & Sorbom, D. (1984). *LISREL VI: User's guide.* Mooresville, IN: Scientific Software.

Loehlin, J. (1987). *Latent variable models.* Hillsdale, NJ: Lawrence Ehrlbaum Assoc.

Pearl, J., & Verma, T. (1990, October). *A formal theory of inductive causation.* (Technical Report R-135). Cognitive Science Laboratory, Department of Computer Science, University of California, Los Angeles.

Spirtes, P., Glymour, C., & Scheines, R. (1990). Causality from probability. In J. Tiles, G. McKee, and G. Dean (Eds.), *Evolving knowledge in the natural and behavioral sciences* (pp. 181–199). London: Pitman.

Verma, T., & Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in AI* (pp. 220–227). Cambridge, MA.