Genetic Algorithm Search Over Causal Models

Shane Harwood and Richard Scheines

March 29, 2002

Technical Report No. CMU-PHIL-131

Philosophy

Methodology

Logic

Carnegie Mellon

Pittsburgh, Pennsylvania 15213

Genetic Algorithm Search over Causal Models

Shane Harwood
Dept of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213

Richard Scheines
Dept of Philosophy
Carnegie Mellon University

Abstract

In this paper, we use a genetic algorithm in combination with a form of simulated annealing to search for DAGs representing linear causal systems. In previous work (Harwood and Scheines, 2002), we showed that a genetic algorithm search based on the Bayes Information Criterion can substantially improve on the constraint based approach of Spirtes, Glymour, and Scheines (2001), especially at sample sizes as low as 100. Here we ultimately score models with the BIC but vary the penalty for model complexity to increase population diversity. We do a form of simulated annealing on the complexity penalty itself, and show that the results surpass those obtained previously, in both speed and expected accuracy. The complexity penalty is initially set very low to produce a set of models perhaps more complex than the true model. We then restrict the search space to models with only adjacencies in the union of the models in the top tier from the previous generation. This exponentially reduces the remaining search space, allowing us to search it much more exhaustively. Over time, the algorithm periodically makes the complexity penalty stricter which further specializes the search space. We present the algorithm and show its reliability on simulation studies of models with up to 30 variables, and sample sizes as small as 100.

0. Introduction

Bayesian Networks and Structural Equation Models are powerful tools used throughout statistics, economics, and the social sciences to parameterize causal hypotheses. Searching for the set of causal Directed Acyclic Graphs (DAGs) that best explain the observed data is difficult due to the exponential growth of the search space as a function of the number of variables measured. When an appropriate score is available and the search space is small enough to search thoroughly, scoring based searches (Cooper, 1999) can be highly accurate. When the search space is large, these tend to be thwarted by local maxima. By using local decisions about independence (or some other constraint) to sequentially cut down the search space, constraint-based searches (Spirtes, Glymour, and Scheines, 2001) are dramatically more efficient, but susceptible to small early mistakes that can expand into large mistakes that cannot be repaired without backtracking, a move which tends to undermine the gain in computational efficiency. In this paper, we present an algorithm that combines the advantages of both constraint propagation and scoring based search. We use a genetic algorithm to do a massive parallel scoring search over patterns (equivalence classes of DAGs; Pearl, 2000), but we also use a form of simulated annealing to achieve dramatic increases in speed similar in spirit to those achieved by constraint-based searches. Our method of annealing sequentially increases the penalty on model complexity in the BIC score, while pruning the search space by eliminating models with adjacencies that we can safely assume do not exist in the real model. Because the algorithm begins with a score more lenient toward complex models than the BIC score, it starts out very conservative with respect to the adjacencies thrown out. Nevertheless, each adjacency thrown out reduces the search space exponentially.

In previous work (Harwood and Scheines, 2002), we presented the SEMGA, a genetic algorithm for Structural Equation Models with a BIC scoring metric. The SEMGA relied on multiple independent searches, each with a slightly different complexity penalty, and dramatically improved on the accuracy of the PC algorithm (Spirtes, Glymour, and Scheines, 2002) for samples of size 100 and models with 20 to 30 variables. We sketch the SEMGA and show how applying a form of simulated annealing on model complexity improves the speed substantially without any loss in performance.

In the remainder of the paper, we first give the briefest possible overview of SEMs, causal search, genetic search, and simulated annealing. Second, we give a brief summary of the basic operation of the SEMGA to set the stage for the annealing-SEMGA adaptation. Third, we explain our version of annealing in more detail, and

explain the exponential savings in search obtained. Fourth, the annealing-SEMGA is compared on a basis of speed and accuracy to the standard SEMGA and the PC algorithm on simulated data. Finally, we speculate on improving the performance of the algorithm in the future.

1. Background

SEMs

Restricting ourselves to searching acyclic causal structures with no latent common causes, a structural equation model (SEM) is a parameterization of a directed acyclic graph (DAG) in which the vertices are variables, and in which each variable is assumed to be a linear function of its direct causes and Gaussian noise. Linear SEMs with variables expressed as a deviation from their mean use the simple function $Y_i = b^T X_i + \varepsilon_i$, where Y_i represents response variable Y's value at individual i, X_i represents a vector of Y's parents values at individual i, b represents a vector of the linear contributions to Y of each of Y's parents. Finally, ε_i represents the residual or error term. For a detailed explanation of DAGs, SEMs, and linear SEMs see (Pearl 2000, or Bollen, 1989)).

Causal Search

The search space involved in DAG exploration is astronomical. For a given number of variables n, there are $\frac{n(n-1)}{2}$ possible adjacencies that can exist in a graph. Each of these adjacencies can be present in a given graph or not making the number of possible adjacency structures $= 2^{\frac{n(n-1)}{2}}$. Each of these adjacencies can then be directed as long as no cycles are generated in the graph. The size of the hypothesis space can be computed using a recurrence relation (Harary, p.19) that sits between $2^{\frac{n(n-1)}{2}}$ and $3^{\frac{n(n-1)}{2}}$. Over a set of 30 variables, 2.7149E158 unique DAGs can be constructed. In order to deal with this space, two basic methods have been proposed: Constraint based search and Scoring based search. Constraint based search (Spirtes, Glymour, Scheines, 2001) uses independence relationships and conditional independence relationships inferred from observed data to determine the adjacencies and then through constraint propagation orient as much of the graph as possible. The main advantage constraint based search algorithms offer is their relative speed, their ability to handle latent variables, and the availability of asymptotic consistency proofs. Two main drawbacks to constraint based search: First, early decisions about independence relations influence which independence relations are even calculated later, so an incorrectly assigned independence relation early has the potential to propagate errors in the graph construction algorithm via the set of later independence relations even considered.[1] Second, in cases in which no DAG exactly entails the independence relations judged to hold in the data, the algorithm has no way to search for the DAG that sits "closest" to the independence structure judged to hold. Scoring based search has nearly the complementary set of advantages and disadvantages. Because proximity among DAGS, at least with respect to small perturbations in adjacencies and/or orientation, translates terribly into scoring proximity over DAGs, local maxima abound and a search for the model or models with the best "score" is hard. As a result, scoring based searches are typically very slow.

Genetic Search

Genetic Algorithms, a subset of scoring algorithms, search for multiple solutions simultaneously. Over time, these solutions are blended with each other and are maintained in a population based primarily on their fitness. The hope is that traits found in the real model improve fitness when included in an organism and are thus imported into the population through probabilistic discovery using crossover and mutation. All genetic algorithms follow some sequence of decisions that can be transformed into the following format, to mimic natural selection.

1) Generate initial population
2) Select a "fit" subset of the organisms from the present population
3) Produce offspring from crossing different "fit" organisms
4) Mutate the current population
5) Return to 2

Simulated Annealing

Annealing is the physical process of cooling a solution slowly to allow crystalline structure to be uniform or maintain super saturation. The standard machine learning definition of simulated annealing denotes a probabilistic hill climbing search where a temperature variable, that decreases over time, governs the probability of moving from one explored state to another. As temperature decreases, the probability of exploring models scoring less then the present state decreases. The motivation is to avoid local maxima by allowing the search to be less strict then hill climbing. The process of adjacency pruning we

[1] A variety of approaches have been taken to make a constraint-based search more robust to early errors. In particular, see Shipley (2000).

describe here simulates simulated annealing (we couldn't resist). In the process we describe, the set of adjacencies included in the possible edge set is analogous to the temperature variable in a standard simulated annealing. The set of possible edges both decreases monotonically and constrains the range of possible mutations of any given solution.

2. SEMGA

Overview

The SEMGA performs a causal search that can be parameterized to search for causal graphs of varying complexity by adjusting the complexity penalty in the usual BIC score for a SEM. The remainder of this section describes: the genomic representation of the model, the scoring metric, and the form of the results returned by an individual search. For a more detailed description of the SEMGA, see (Harwood and Scheines, 2002).

Genomic Representation

Each DAG is represented by a genome or sequence of traits, one for each possible edge between any two variables found in the graph. The alleles are represented by the values of the individual traits, where each trait represents an edge's orientation or absence from the model. For each variable in the model, a singular value decomposition regression calculates a regression intercept and a linear coefficient for each of the variable's direct causes. Therefore, each genome/graph uniquely generates a SEM given the observed data allowing each representation to be scored.

Scoring (modified BIC)

The scoring function we employ is a modification of the Bayesian Information Criterion (BIC), an approximation of the posterior probability of a SEM given observed data. The BIC score can be broken into two parts: First, a measure of how similar the covariance matrix of the data is to the covariance matrix implied by the model at the ML parameter estimate. Second, a penalty based on the complexity of the model, which in acyclic no-latent variable SEMs is proportional to the number of adjacencies in the graph, or pattern. Our modification to improve diversity is to treat the penalty contribution as a parameter of the search. The benefit is that searches can be biased away from edge omission, which allows us to more conservatively generate sets of partial information, i.e., sets of adjacencies to include in the search and their complement to ignore.

Frontiers (result format)

Frontiers represent our set of best estimates of the model that generated the observed data. Varied levels of aversion to complexity are represented by each model in the frontier. We can store frontiers representing an individual search result or representing results for all searches ever performed. The parameterization of the individual searches influences what region of the frontier will be specialized. A combined or general frontier of organisms will thus contain models with a range in complexity, i.e., the number of adjacencies, roughly proportional to the range of the penalty factors used.

3. Annealing on Model Complexity

Consider the set of models in a general frontier, and consider the set of adjacencies that occur in any model in this frontier. We call this the set of edges included in the general frontier, and its complement, i.e., the set of edges that occur in no model in the general frontier, the set of edges restricted by a general frontier. The algorithm slowly makes this general frontier more specific as it discovers information.

Initialization

In the beginning of our modified annealing search, no partial information exists, and all edges are possible, that is, included in the initial frontier. The complexity penalty for edges is initially set very low (a third of the standard edge penalty found in the BIC) in-order to produce a frontier very generous with respect to the set of edges included. We execute three SEMGA searches, each seeded differently, and construct our initial general frontier from the union of the frontiers produced by these three searches. This initial search is computationally expensive, precisely because the search space has not yet been pruned at all. At the completion of this initial stage, we have a general frontier that is very conservative with respect to the edges restricted. These edges are never searched again; this restriction reduces the search space exponentially in the number of edges restricted by the frontier.

Iteration

We then continue the three SEMGA searches, but with an edge complexity penalty increased from the previous iteration. Again, after the iteration is complete, we recalculate the general frontier, and add any edges to the set of restricted edges we found from previous iterations. At each iteration the set of restricted edges grows, and each successive search faces a dramatically smaller search space, which can be searched more exhaustively than in

previous iterations. We are thus essentially annealing on the complexity of the model, or the set of restricted edges.

Final Phase

Once the initial phase and several iterations have been executed, we arrive at a single independent SEMGA search with the standard BIC scoring metric. We shrink the already small general frontier by removing the most complicated element until we simplify the model. The algorithm stops at the point when the most complicated model found in the general frontier is the model with the highest BIC score.

4. Simulation Results

20 Variable Datasets

The first set of graphs (figures 1 and 2) represent the results of search run on 43 simulated data files. Each of the data files had a sample of 100 drawn for 20 variables produced from a randomly generated causal graph and randomly parameterized SEM interpretation of that graph. We divided the 43 studies into 4 groups based on the average indegree of the randomly generated causal graph in order to achieve meaningful results upon averaging. The breakdown of the groups is approximately 11 models from the following average indegree ranges: (0.25 - 1), (1.05 - 1.3), (1.35 - 1.75), and (1.8 - 2.4)

Figure 1

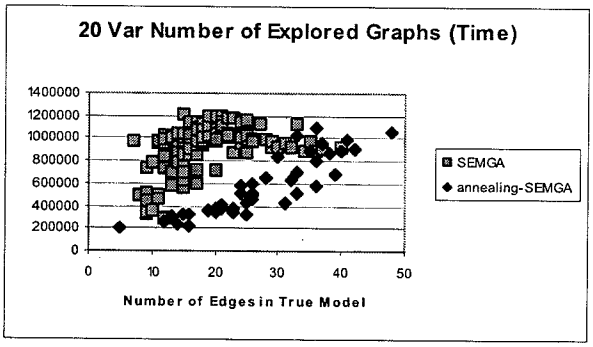


Figure 1 concerns the relative speed of the two searches. The speed of the search depends on the number of models visited, since the time for search is dominated by the time to spent scoring models. The average quantity of models scored in the annealing-SEMGA was significantly less then in the standard SEMGA, even though the accuracy of the procedure is as good or better. The number of models scored by the annealing-SEMGA rises approximately linearly with the model complexity, but the number scored by the SEMGA plateaus after approximately 20 edges. This was primarily due to SEMGA exhausting the bounded computational resources we had allowed. The SEMGA calculates roughly the same quantity of graph scores each execution unless the data was generated by an extremely sparse graph. Our method of annealing takes advantage of the reduction in the search space that results from the number of edges restricted by the initial frontier. For complex graphs, the annealing-SEMGA's control of its own annealing process allows it to dictate how hard it will work on a problem, hence the increasing number of models searched as the true model gets more complicated.

The four graphs in Figure 2 compare the accuracy of the PC algorithm (Spirtes, Glymour, and Scheines, 2001), standard SEMGA and annealing SEMGA. It plots the accuracy of each algorithm on edge commission, edge omission, orientation commission, and orientation omission. In each case, the results list the percentage error with respect to true pattern determined by the true graph. If, for example, the output pattern contained 20 edges, 10 of which did not exist in the true pattern, then the error percentage is the 10 divided by the number of non-adjacencies in the true pattern, which is the number the algorithm could have committed.

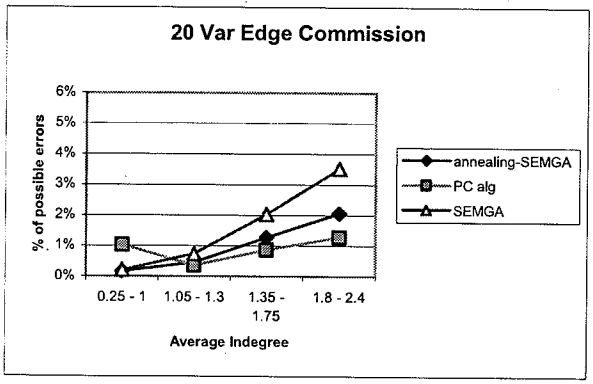
Ultimately, the annealing-SEMGA search returned results as good or better than the SEMGA search, in much less time. Both genetic algorithm searches radically outperform the PC algorithm.

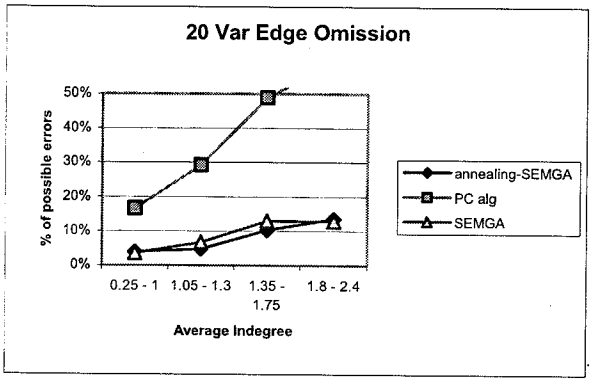
30 Variable Datasets

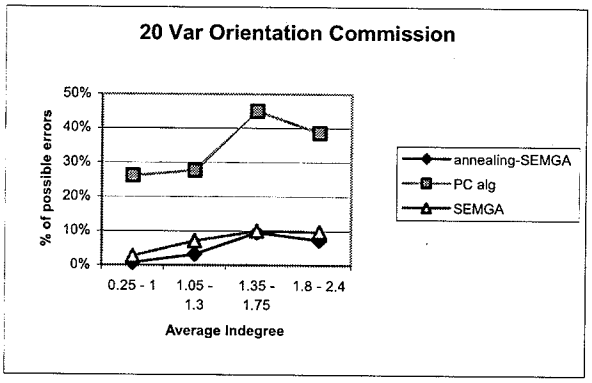
The second set of graphs (figure 3) represent the results of search ran on 42 simulated data files, again with N=100. The searches have been divided into 4 groups based on average indegree in order to achieve reliable results upon averaging. The breakdown of the groups is approximately 10 models from the following average indegree ranges: (0.63 – 0.9), (0.93 – 1.07), (1.1 - 1.27), and (1.33 – 1.5).

Figure 3 shows a comparison between the annealing – SEMGA and the PC algorithm on accuracy of edge commission, edge omission, orientation commission, and orientation omission. The annealing-SEMGA dominates the PC in all forms of error except for edge commission on the most complex set of models; this error of 0.5% translates to approximately two committed edges on average. We were unable to obtain results for the SEMGA algorithm at 30 variables, because the search had difficulty terminating with the limited resources we gave it.

Figure 2







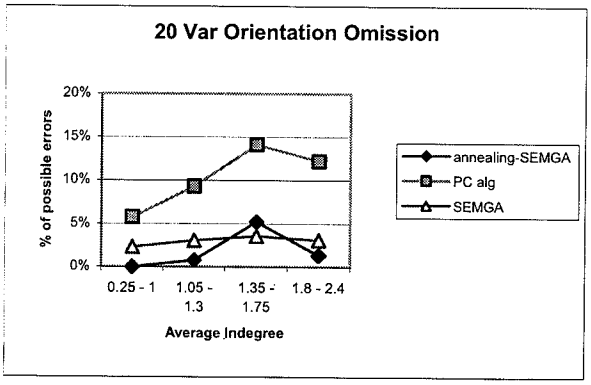
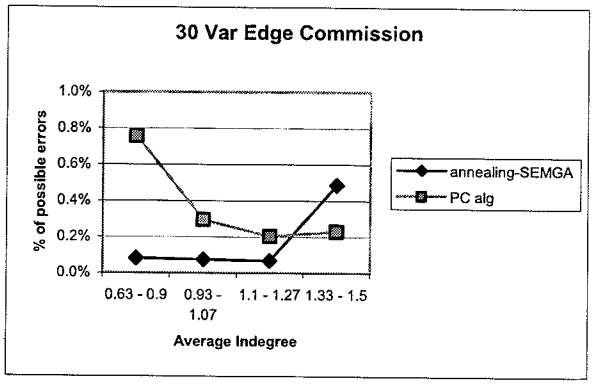
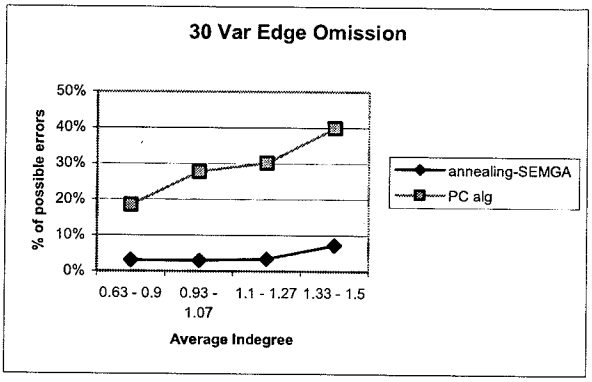
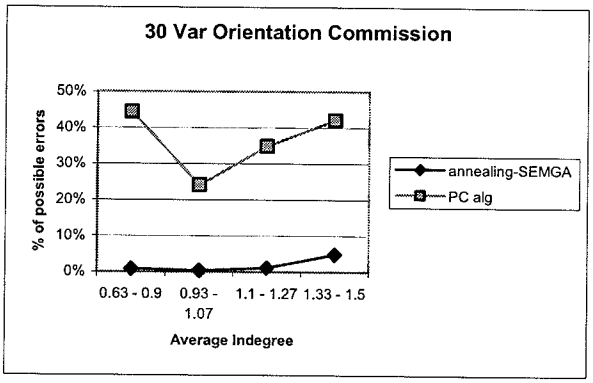
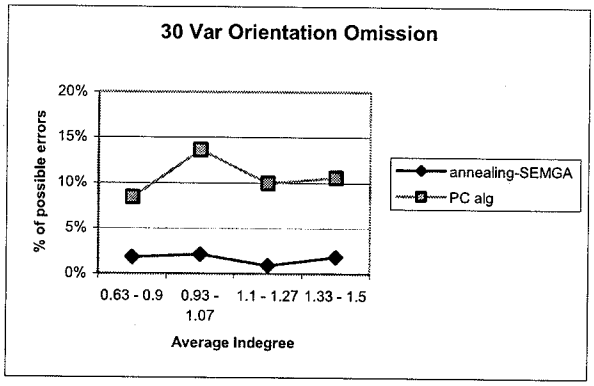


Figure 3









5. Future Work

The plans for the SEMGA's future are threefold: First, the search should extended to cyclic graphs and graphs that contain latent variables. Second, new genomic representations are being considered. There is a chance that a representation of a causal ordering mapped over an adjacency structure may encode the data better, allowing for more intelligent crossover and mutation. This representation has been used frequently in Bayesian Network structure learning (deCampos, 1999). Third, the algorithm assumes that the data we've seen is derived from a linear system. Linear systems are not ubiquitous in nature, and they fail to represent some of the simplest relations such as conjunction and disjunction. Recently, genetic algorithms have been used to specify the functional form of the structural equations that parameterize a causal graph (Marcoulides and Drezner, 2001). Incorporating functional form specification into our annealing search for causal models could well improve the overall reliability of our method for finding causal models.

6. References

Bollen, K.A. (1989), *Structural Equations with Latent Variables*, New York: John Wiley and Sons.

deCampos, Jose A., Gamez, Serafin Moral (1999). *Computation Approach by using Problem Specific Genetic Operators.* University of Spain, Grenada.

Glymour, C., and Cooper. G., (1999). *Computation, Causation, and Discovery*, Boston: AAAI Press and MIT Press.

Harary, Frank (1973). *Graphical Enumeration.* Academic press, New York and London

Harwood S., and Scheines R. (2002) *Learning Linear Causal Structure Equation Models with Genetic Algorithms,* Technical Report CMU-PHIL-128, Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA.

Marcoulides, George A., Drezner Z. (2001). "Specification Searches in Structural Equation Models With a Genetic Algorithm," Chapter 9 in *New Developments and Techniques in Structural Equation Modeling,* Lawrence Erlbaum and Associates, Mahwah, NJ.

Pearl, Judea. (2000). *Causality: Models, Reasoning, and Inference*, Cambridge University Press.

Spirtes, P., Glymour, C., & Scheines, R. (2001). *Causation, Prediction, and Search*. MIT Press.