Path Analysis Models of an Autonomous Agent in a Complex Environment

Paul R. Cohen, David M. Hart, Robert St. Amant,
Lisa A. Ballesteros and Adam Carlson

Computer Science Technical Report 94-33

Experimental Knowledge Systems Laboratory
Department of Computer Science, Box 34610
Lederle Graduate Research Center
University of Massachusetts
Amherst, MA 01003-4610

Abstract

We seek explanatory models of how and why AI systems work in particular environments. We are not satisfied to demonstrate performance, we want to understand it. In terms of data and models, this means we are not satisfied with descriptive summaries, nor even with predictive models. We want causal models. In this brief abstract we will present descriptive, predictive and causal models of the behavior of agents that fight simulated forest fires. We will describe the shortcomings of descriptive and predictive models, and summarize path analysis, a common technique for inducing causal models.

This research was supported by ARPA under contracts F30602-91-C-0076 and F30602-93-C-0100; and by AFOSR under the Intelligent Real-time Problem Solving Initiative, contract AFOSR-91-0067.

1 Phoenix

Phoenix is a simulated environment populated by autonomous agents. It is a simulation of forest fires in Yellowstone National Park and the agents that fight the fires [Cohen et al. 89]. Agents include watchtowers, fuel trucks, helicopters, bulldozers and, coordinating (but not controlling) the efforts of all, a fireboss. Fires burn in unpredictable ways due to wind speed and direction, terrain and elevation, fuel type and moisture content, and natural boundaries such as rivers, roads and lakes. Agents behave unpredictably, too, because they instantiate plans as they proceed, and they react to immediate, local situations like encroaching fires.

Phoenix *appears* to work very well, in that bulldozers are expeditiously dispatched to fires and appear to coordinate their efforts, and fires are usually contained. But as with any complex system, one cannot tell much from a demonstration. (In fact, the demo looked fine even though, as we later discovered, a critical algorithm was projecting events minutes instead of hours in the future!) For this reason we turned to an extensive statistical analysis of specific aspects of behavior and the factors that affect them.

2 Modeling Phoenix

We created a straightforward fire fighting scenario that controlled for many of the variables known to affect the planner's performance. In each of 343 trials, one fire of a known initial size was set at the same location (an area with no natural boundaries) at the same time (relative to the start of the simulation). Four bulldozers were used to fight it. The wind's speed and direction were set initially and not varied during the trial. Thus, in each trial, the Fireboss receives the same fire report, chooses a fire-fighting plan, and dispatches the bulldozers to implement it. A trial ends when the bulldozers have successfully surrounded the fire or after 120 simulation-time hours without success.

For exploratory purposes we collected over 40 measurements on each trial, but we will concentrate on eight here.

Success. If the fire is contained in less than 120 simulated hours, the trial is a success, otherwise it is a failure.

Wind speed. We set the wind speed to be either 3, 6 or 9 kilometers per hour and didn't change it during a trial.

Realtime knob. This parameter controls the ratio of the fireboss's "thinking time" to the elapsed "environment time." The normal setting allows the fireboss one simulated minute of thinking time for one minute of environment time, during which fires burn. Halving this parameter means the fireboss takes twice as long, relative to the environment, to think.

Utilization. The proportion of a trial that the fireboss spends thinking, as opposed to idle, or waiting for requests, etc.

First plan. Phoenix adopts one of three general plans to fight a fire.

Number of plans. Phoenix sometimes replans when its first plan doesn't work. If number of plans = 1, then no replanning occurred.

Fireline built. This measures the number of meters of fireline cut by the bulldozers in a successful trial.

Shutdown time. This measures the number of hours, from the first sighting to the return to base of the last bulldozer, required to contain the fire.

2.1 Descriptive models

We wanted to know how these factors interacted; specifically, we wanted to know how success and shutdown time depended on the other factors. (We will focus on shutdown time, here, but see [Hart and Cohen 92]). A descriptive model is simply a correlation matrix of the factors. We don't recommend this kind of model. We suggest it only to point out its shortcomings, the most glaring of which is that a correlation represents the endpoints of a causal story, not the story itself. For example, we were very surprised to find in our data that wind speed was essentially uncorrelated with shutdown time ($r = -.053$). We knew that higher wind speed increased the spread rate of the fire, which would require more fireline, which would take more time to cut. If this causal story is right, then $r = -.053$ must represent the sum of two influences, the one we just described and another that cancels it out and also depends on wind speed. We traced the problem to a sampling bias in our experiment: For high wind speeds, if a fire isn't contained relatively quickly, then it might not be contained at all. For example, if a fire has been burning for 60 hours or more, and wind speed is 3, then the probability of the fire being contained eventually is .375. But if wind speed is 6, the probability of eventually containing an old fire is only .2, and if wind speed is 9, the probability drops to .13. We measured shutdown time for successful trials only. But successful containment of old fires is relatively unlikely at higher wind speeds, so as wind speed increases, we see fewer older fires contained, thus fewer high values of shutdown time. This results in a negative relationship between wind speed and shutdown time. Note that this relationship represents an effect of missing data, not a true negative causal relationship.

The two influences are shown in a *path diagram* in Figure 1. The path from wind speed to fireline built to shutdown time is the first causal story. We hypothesized another path, corresponding to the second story, from wind speed to shutdown time. By the rules for estimating correlations from path diagrams (described later) the estimated correlation between wind speed and shutdown time is the sum of the two stories: first, the effect of wind speed on fireline built multiplied by the effect of fireline built on shutdown time ($.363 \times .892$); second, the effect that the sampling bias must have ($-.377$) to offset the first story to the extent that the overall correlation is $-.053$. This model does not tell us that the second story is correct—only how strong the hypothesized sampling bias must be. Still, we prefer path models such as Figure 1 to raw correlations because they can tease apart the causal stories that are often confounded in correlations.

2.2 Predictive regression models

We wanted a predictive model of how factors influence shutdown time in Phoenix, so we turned to multiple linear regression. Again, we do not recommend this technique, but present it to highlight its limitations. Here is a regression equation for shutdown time; the coefficients are standardized:

$$\begin{aligned} est.Shutdown &= .762 FirelineBuilt + .144 NumPlans + .036 FirstPlan - \\ &\quad .192 RealTimeKnob - .283 WindSpeed \end{aligned}$$

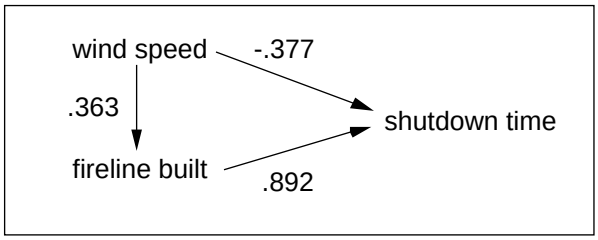

Figure 1: A path model of the influences of wind speed and fireline built on shutdown time.

Unfortunately, while predictive, this model is hardly explanatory. It is a completely "flat" causal story. All it says is that five factors influence the dependent variable to different degrees. If our goal is to understand the interactions of factors in a complex environment (i.e., wind speed and real time knob) with decisions and actions of a complex agent (e.g., which plan to use first, how often to replan, and fireline built), then this model is useless. It is also unparsimonious, as described in [Hart and Cohen 92].

3 Causal path models

Path analysis is an exploratory modeling technique that has helped us incrementally develop a causal understanding of Phoenix. It is based on multiple regression, but it can be used to fit "multilevel" models in which some predictors point to the dependent variables and are in turn predicted by other variables. One way to run a path analysis is to specify a path model and see how well data fit it; another approach is to let the data suggest path models. The latter approach is causal induction, similar to the work of [Pearl and Verma 91], and [Spirtes et al. 93]. An algorithm is described later. Here, let us focus on the former approach, where the analyst specifies a path model. Figure 2 is a model we derived from the experiment described above. Note that it is neither flat nor unparsimonious: some causal influences are mediated by one or two endogenous variables, and the model includes only a fraction of the influences that are implicit in a multiple regression model. Consequently, the coefficient of determination R^2 of this model is lower than for the regression model. The trick in path analysis is to get a good degree of fit to the data while preserving the causal structure that you think exists, and elucidating causal structure that you didn't suspect. Path analysis has three steps:

1. Propose a path model. The model represents causal influences with directed arrows (e.g., $FirelineBuilt \rightarrow ShutdownTime$) and correlations with undirected links (see Figure 2).

2. Derive path coefficients, the magnitude of which are interpreted as a measure of causal influence.

3. Estimate the correlation between two factors by multiplying path coefficients along paths between the factors and summing the products over all legal paths between the factors.

The three steps are repeated, modifying the model each time, until we have a model that is good by some criteria (more on this shortly). The second step is accomplished by multiple regression: A variable that is pointed to by one or more arrows is regressed on the variables at the tails, and the

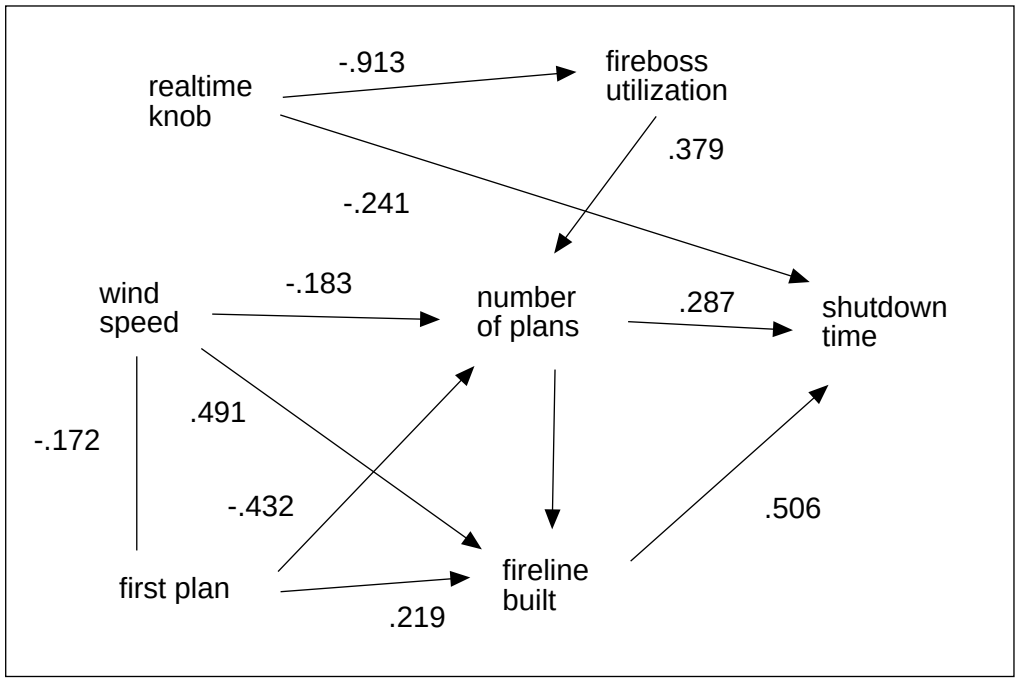

Figure 2: A path model derived from Phoenix data.

standardized regression coefficients are the path coefficients. For example, the regression of fireline built on first plan, wind speed and number of plans produces the following standardized regression model: $est.FirelineBuilt = .219(FirstPlan) + .491(WindSpeed) + .849)(NumPlans)$.

Two common statistical criteria of goodness are that the model accounts for a lot of the variance in the dependent variable and the model does a good job of estimating empirical correlations. The first criterion is the familiar R^2; the second requires some explanation. From the normal equations that underlie regression, we can derive a chain rule that says the estimated correlation between two variables is the sum of the products of the path coefficients on all the legal paths that connect the variables. Legal paths are those that don't visit a node more than once, don't follow arrows backward after they have gone forward, and don't include more than one undirected arc (see [Asher 83];[Li 75]; [Loehlin 87]; [Cohen 94] for details). For example, the estimated correlation between wind speed and shutdown time is $(-.183)(.287) + (-.183)(.843)(.506) + (.491)(.506) + (-.172)(.219)(.506) + (-.172)(-.432)(.287) + (-.172)(.432)(.843)(.506) = .089$. We have already seen that the empirical correlation between wind speed and shutdown time is -.053. This is the largest disparity between estimated and empirical coefficients in the entire model, so, by and large, the model does a pretty good job of explaining the empirical correlations. At the same time, the model accounts for 73% of the variance in shutdown time, 69% of the variance in fireline built, and 31% of the variance in the number of plans. This latter figure seems low and makes us wonder whether the variables in the model are "connected right." Perhaps another model does as good a job estimating correlations and also accounts for more of the variance in the dependent variables. We haven't found it.

4 An algorithm for automating path analysis

Estimating the fit of a path model in terms of R^2 and estimated correlations is straightforward, but proposing path models is not. Or rather, it is easy to propose models, but because the number of ways to connect N nodes with single- or double-headed arrows is 2^{N^2-N}, it is wise to propose them intelligently. We have developed an algorithm called PA that automatically searches the space of path models for those that fit the data and satisfy other heuristic criteria. The algorithm explores the search space by gradually refining models until a satisfactory model is reached or no more possibilities remain. Model refinement is by means of modifications. A modification to a model adds a direct causal influence (i.e., a new directed edge) between two unrelated variables, or a correlation between two directly related variables. The algorithm builds complex causal models incrementally, by modifying simpler models. Through successive modifications the algorithm can generate and evaluate all possible models, although this is to be avoided in most cases.

In general terms, the algorithm maintains a "best model" and a pool of further possibilities, in the form of potential modifications to models that have already been examined. The algorithm selects the best of these potential modifications and executes it, producing a new model. All the legal modifications of this new model are generated, scored, and returned to the pool of modifications. The new model itself is scored and compared to the best model. If the new model scores higher, it becomes the best model. The process repeats until the modification pool is empty. The best model encountered is finally returned.

Because of the size of the search space (e.g., 1,048,576 for five variables) the algorithm must be constrained. We limit consideration of modifications to those that are syntactically correct and meet particular heuristic criteria. Further, we guide search by considering the most promising modifications first. Syntactic criteria ensure that a model meets the requirements entailed by structural equations. There are several: We specify exactly one of the model variables as a dependent variable, which has no outgoing edges. Some model variables may be specified as independent variables, which have no incoming edges (bidirectional or correlation edges between independent variables and other variables are allowed, however.) The model must consist of a single connected component, not two or more unconnected substructures. A model may not contain a cycle. These constraints reduce the search space a good deal, but still only by a polynomial factor.

If a modification to a model will produce a new model that meets these syntactic requirements, it is then tested against heuristic criteria. There are two distinct kinds of heuristics: those that apply to the introduction of an new edge in a model, and those that evaluate the model as a whole.

Consider a modification that involves introducing an edge in a model from variable S to variable D. First, if variables S and D have some common ancestor, i.e., a common cause, a correlation edge directly between S and D is not allowed. Second, if the partial correlation between S and D with some third variable V held constant is below a threshold, then a direct edge between S and D is not allowed. The rationale for this heuristic is that the effect of S on D may be mediated by an intermediate variable, which is indicated by a low partial. Variations on this second heuristic constrain those variables considered as intermediate variables.

The second set of heuristic criteria deals with the model produced by a prospective modification. First, we may specify an upper limit on the number of edges in a model. Second, we may similarly specify an upper limit on the number of correlation edges in a model. Third, we may set a maximum on the branching factor of outgoing edges from any variable. All these heuristics limit the complexity of the models our algorithm produces; they are one means of encouraging parsimony.

What remains from this pruning process is a set we call the legal modifications to a model. These modifications are then evaluated for comparison with other possible modifications in the pool. The algorithm evaluates each modification by summing two component scores:

- $R^2_{new} - R^2_{old}$ is the difference between the determination coefficient for the existing model and the model produced by the modification. This component of the score prefers modifications that produce the largest increase in the statistical explanatory power of the model.
- $\sum_{i,j \in \text{vars}}(|r_{i,j} - \text{e\_old}_{ij}| - |r_{i,j} - \text{e\_new}_{ij}|)$ is more complex. For any two variables in a model we can calculate the estimated correlation based on the topology of the model, and the actual correlation produced by the data. If the difference between these two values is small, then this is a good model with respect to those two variables. The sum of this difference over all variable combinations in the model is one measurement of how well the model fits the data. The total value is the difference between the value of the existing model and the value of the model produced by the modification.

Once the highest scoring modification is selected and applied, the algorithm must evaluate the model that is produced. This model is scored on two counts: the determination coefficient for the dependent variable, and the summed difference between actual and estimated correlations. The highest scoring model is save d.

Besides limiting the search space, the algorithm must deal with two other efficiency concerns, the calculation of betas and of estimated correlations. Because any partial regression coefficient may be called for in evaluating a model, it may be necessary in the worst case to calculate all possible regressions during the course of the search. This requires an exponential number of calculations. Estimated correlations pose a larger problem. They are calculated by following paths in the model, and again in the worst case the process is exponential. The algorithm relies on the pruning process, in particular on the limitations on branching factor and the number of edges, to reduce these calculations to polynomial time.

4.1 Illustration of PA

The PA algorithm has been tested with artificial data [Cohen et al. 93] and also with the Phoenix dataset. We are currently conducting a study in which it is compared with Pearl's algorithm [Pearl and Verma 91] and the TETRAD II algorithm [Spirtes et al. 93]. We are also improving the algorithm. Thus, detailed evaluation of PA is forthcoming, and this section must be read more as an illustration than an evaluation.

We ran PA on a subset of the Phoenix dataset, leaving out wind speed and fireboss utilization. We told PA that realtime knob and first plan were independent variables, and shutdown time was a dependent variable. These facts reduced the size of PA's search space to 4096 possible models. Syntactic and heuristic pruning reduced the space further to 311 models. The desired model (the one that we built ourselves as a good model of Phoenix) is shown in Figure 3. When we ran PA on what we will call the *full* dataset, its best-ranked model was almost identical to Figure 3, missing only the link between first plan and fireline built. This link was pruned by the partial correlation heuristic, mentioned earlier.

We were concerned that we had somehow tuned PA to perform well with the Phoenix dataset, so we partitioned this dataset by wind speed into three smaller sets of 85, 67 and 63 trials at

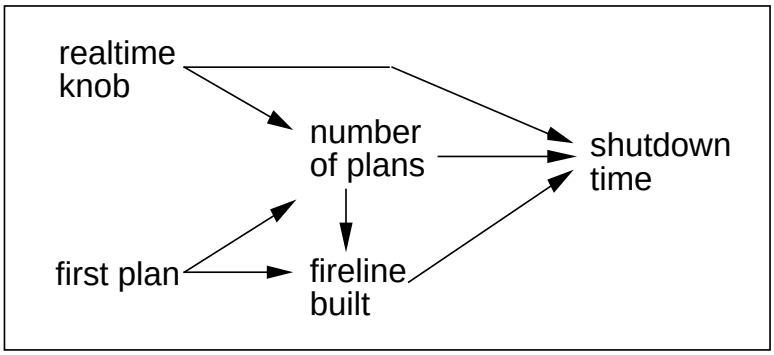

Figure 3: The desired model.

wind speed of 3, 6 and 9 kph, respectively. We thought that PA ought to produce "essentially the same" model in these three conditions, recognizing, of course, that the covariances between wind speed and the other variables would be somewhat different in each condition. At the same time, we would be disappointed if the models were very different. After all, the "core" of the Phoenix planner should operate in pretty much the same way whether the wind speed is low, medium or high. For instance, the realtime knob should still influence the number of plans, which should in turn influence the amount of fireline built, and so on. If data from the three wind speed conditions produced very different models, it might mean that PA is too sensitive to unimportant differences in the covariance structure among the variables.

As you can see in Figure 4, many aspects of the models are common, and the differences are interesting "near misses." First, all the models have shutdown time directly influenced by realtime knob, number of plans, and fireline. All of the models have realtime knob influencing fireline and number of plans: in the 3 kph model, realtime knob influences fireline indirectly through number of plans; in the 6 kph model, both influences are direct; in the 9 kph model, fireline is directly influenced and in turn influences number of plans. The influence of first plan is hardest to pin down: it influences shutdown time, number of plans and fireline, respectively, in the three conditions. It appears that as wind speed increases, the influence of the first plan on shutdown time becomes more indirect. In retrospect, this is exactly what we should have expected: at low wind speed, Phoenix replans very infrequently, hence the influence of the first plan on shutdown time is greatest. At higher wind speeds, replanning is generally necessary, unless Phoenix's first plan happened to be one called "model," which was very conservative and built such a large perimeter of fireline that replanning was rarely required. Thus, at higher wind speed, the choice of the first plan should directly influence either the number of plans or the amount of fireline built.

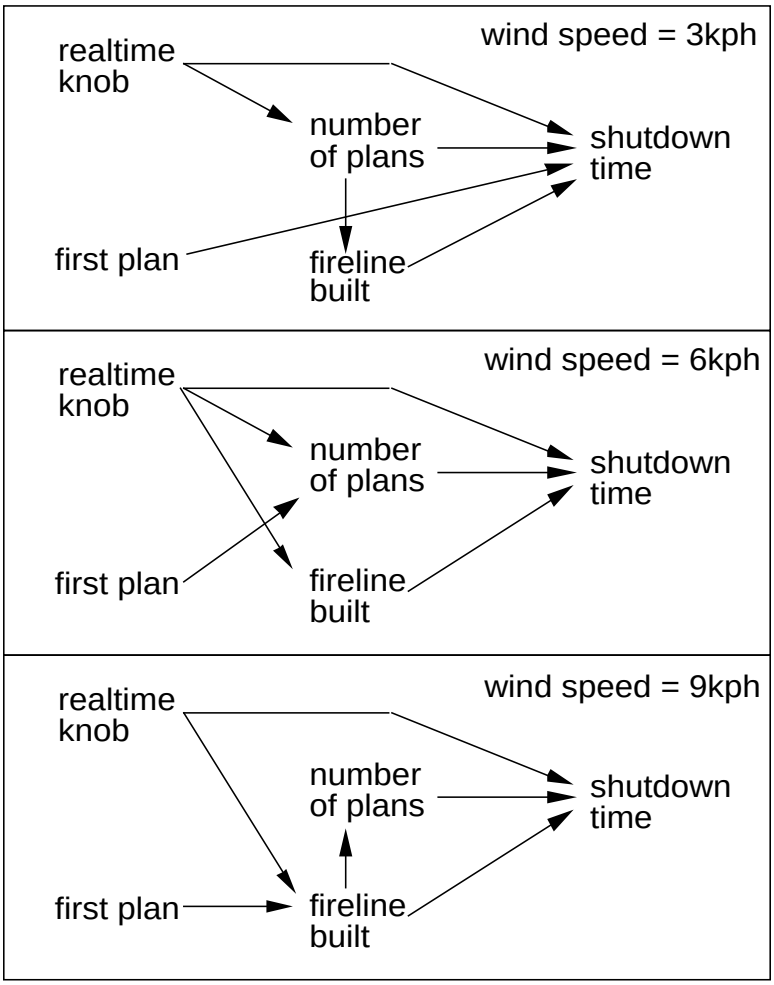

Figure 4: The top-ranked models produced by PA for partitions of the Phoenix dataset by wind speed.

5 Conclusion

Path analysis appears to be a promising technique for modeling complex AI systems, although our studies to date have been small and our results preliminary. Although one can build and estimate path models by hand, it would be better to give the job to an intelligent algorithm. Ours appears to work well on the Phoenix dataset, and also on artificial datasets, although a larger, more discriminating study is clearly needed.

Acknowledgments

This research was supported by ARPA/Rome Laboratory under contracts F30602-91-C-0076 and F30602-93-C-0100; and by AFOSR under the Intelligent Real-Time Problem Solving Initiative, contract AFOSR-91-0067. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

References

[Asher 83] Asher, H.B. (1983) *Causal Modeling.* Sage Publications.

[Cohen et al. 89] Cohen, P.R., Greenberg, M.L., Hart, D.M., and Howe, A.E. (1989) "Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments", *AI Magazine* **10(3)**, 32-48.

[Cohen et al. 93] Cohen, P.R., Carlson, A., and Ballesteros, L.A. (1993) "Automating Path Analysis for Building Causal Models from Data", *Proc. 10th Intl. Conf. on Machine Learning*, Amherst, MA, in press.

[Cohen 94] Cohen, P.R. (1994) *Empirical Methods for Artificial Intelligence.* The MIT Press, forthcoming.

[Hart and Cohen 92] Hart, D.M. and Cohen, P.R. (1992) "Predicting and Explaining Success and Task Duration in the Phoenix Planner", *Proc. 1st Intl. Conf. on AI Planning Systems, College Park, MD*, 106-115.

[Li 75] Li, C.C. (1975) *Path Analysis - A Primer.* Boxwood Press.

[Loehlin 87] Loehlin, J.C. (1987) *Latent Variable Models.* Lawrence Erlbaum Associates.

[Pearl and Verma 91] Pearl, J. and Verma, T.S. (1991) "A Theory of Inferred Causation", *Principles of Knowledge Representation and Reasoning: Proc. of 2nd Intl. Conf*, Morgan Kaufmann, 441-452.

[Spirtes et al. 93] Spirtes, P., Glymour, C. and Scheines, R. (1993) *Causation, Prediction and Search.* Springer-Verlag, New York.

Availability of Code

CLASP, our Common Lisp statistics package, is available. The path analysis code is part of CLASP. CLASP uses CLIM and will run on most platforms. Please contact David Hart (dhart@cs.umass.edu) if you want any of this code.