On the Application of The Bootstrap for Computing Confidence Measures on Features of Induced Bayesian Networks

Nir Friedman
The Institute of Computer Science
The Hebrew University
Jerusalem 91904 ISRAEL
nir@cs.huji.ac.il

Moises Goldszmidt
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
moises@erg.sri.com

Abraham Wyner
Department of Statistics, Wharton School
University of Pennsylvania
Philadelphia, PA
ajw@stat.wharton.upenn.edu

Abstract

In the context of learning Bayesian networks from data, very little work has been published on methods for assessing the *quality* of an induced model. This issue, however, has received a great deal of attention in the statistics literature. In this paper, we take a well-known method from statistics, Efron's Bootstrap, and examine its applicability for assessing a confidence measure on features of the learned network structure. We also compare this method to assessments based on a practical realization of the Bayesian methodology.

1 Introduction

In the last decade there has been a great deal of research focused on the issue of learning Bayesian networks from data. With few exceptions, these results have concentrated on issues of computationally efficient induction methods and, more recently, on the issue of hidden variables and missing data. Very little work (but see below) has been published on methods or on a methodology for assessing the quality of an induced model.

In this paper, we are interested in the following questions: With what *confidence* can we establish that a given feature of the induced network is part of the golden model that generated the data (if such model exists)? Is the existence of a directed arc in our induced model the product of chance/noise, or is it a statistically valid conclusion from the data?

There have been few attempts to answer such questions in the literature: Cowell et al. [4] present a method based on the log-loss scoring function to *monitor* each variable in a given network. These monitors check the deviation of the predictions by these variables from the observations in the data. Heckerman et al. [7] present an approach, based on Bayesian considerations, to establish the belief that a causal edge is part of the underlying generating model.

The problem we study in this paper, is similar in spirit to the one investigated by Heckerman et al. We are concerned with being able to assess a confidence measure to *features* of an induced network structure. To this end, we examine an approach based on the Bootstrap method of Efron [5]. The Bootstrap is a computer-based method for assigning measures of accuracy to statistics estimates and performing statistical inference. Although the Bootstrap is conceptually easy to implement and apply, there are issues about the convergence of the (confidence) estimates computed with it in the the domain of Bayesian networks. Convergence require first asymptotic consistency of the (Bayesian network) induction algorithm, and second a continuity condition on the feature being examined. This paper provides empirical evidence that, in practice, high confidence estimates on structural features are indicative of the existence of these features in the generating model.

There many possible features we might examine such as whether there is an edge from X to Y, or whether X is in Y's Markov blanket. In general, however, we must be careful about the features we select, since observational data alone cannot distinguish between *equivalent* networks, namely networks encoding the same independence statements. In particular, we are better off if we limit our attention to distinguishing features amongst equivalent classes of networks than to features of the particular induced network (which can be an arbitrary choice from the equivalence class).

The features we examine in this paper are edges in the *partially directed graph* (PDAG) that describes the equivalence class of the learned network. These edges can be either directed, denoting that the edge direction is the same in all equivalent networks, or undirected, denoting that either direction is possible in some equivalent network. Results of [2, 8] describe the relationship between such PDAGs and equivalence classes of Bayesian networks. In particular, every equivalence class can be represented by a unique PDAG.

The rest of the paper is organized as follows: In Section 2, we briefly review the definition of Bayesian networks and the methods for learning them. In Section 3, we suggest two methods, based on the Bootstrap, for assessing our confidence in a partially directed edges in a Bayesian network. In Section 4 we suggest a variant of the method of Heckerman et al. for Bayesian estimation of our beliefs about such edges. In Section 5, we present results on experiments with the three methods and compare their behavior. In this section, we also suggest several ways of visualizing our confidence estimation. Finally, future work and the further application of these techniques are discussed in Section 6.

2 Learning Bayesian Networks

Consider a finite set $\mathbf{X} = \{X_1, \ldots, X_n\}$ of discrete random variables where each variable X_i may take on values from a finite set. We use capital letters, such as X, Y, Z, for variable names and lowercase letters x, y, z to denote specific values taken by those variables. Sets of variables are denoted by boldface capital letters $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$, and assignments of values to the variables in these sets are denoted by boldface lowercase letters $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

A *Bayesian network* is an annotated directed acyclic graph that encodes a joint probability distribution of a set of random variables $\mathbf{X}$. Formally, a Bayesian network for $\mathbf{X}$ is a pair $B = \langle G, \Theta \rangle$. The first component, namely G, is a directed acyclic graph whose vertices correspond to the random variables $X_1, \ldots, X_n$, and whose edges represent direct dependencies between the variables. The graph G encodes the following set of independence statements: each variable X_i is independent of its non-descendants given its parents in G. The second component of the pair, namely Θ, represents the set of parameters that quantifies the network. It contains a parameter $\theta_{x_i|\mathbf{pa}(x_i)} = P_B(x_i \mid \mathbf{pa}(x_i))$ for each possible value x_i of X_i, and $\mathbf{pa}(x_i)$ of $\mathbf{pa}(X_i)$, where $\mathbf{pa}(X_i)$ denotes the set of parents of X_i in G. A Bayesian network B defines a unique joint probability distribution over $\mathbf{X}$ given by:

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} P_B(X_i \mid \mathbf{pa}(X_i)).$$

The problem of learning a Bayesian network structure can be stated as follows. Given a *training set* $D = \{\mathbf{x}[1], \ldots, \mathbf{x}[N]\}$ of instances of $\mathbf{X}$, find a network B that *best matches* D. The common approach to this problem is to introduce a scoring function (or a *score*) that evaluates the "fitness" of networks with respect to the training data, and then to search for the best network (according to this score). In this paper we use the score proposed in [6] which is based on Bayesian considerations, and which scores a network structure according to the posterior probability of the graph structure given the training data (up to a constant). We will take advantage of this fact in Section 4 for computing a Bayesian estimation of the belief of the existence of an edge in a model.

Finding the structure that maximizes the score is usually an intractable problem [3]. Thus, we need to resort to heuristic search to find a high-scoring structure. Standard proposals for such search include greedy hill-climbing, stochastic hill-climbing, and simulated annealing; see [6]. In this paper we will focus on a greedy hill-climbing strategy.

In our experiments, we will not assess directly the confidence on the existence of an arc in the induced network, but rather, on the existence of an arc in the *equivalent class* of networks. As discussed in the introduction, two Bayesian network structures G and G' are *equivalent*, if they imply exactly the same set of independence statements. The results in [2, 8] establish that two equivalent networks structures must agree on the connectivity between variables, but might disagree on the direction of the arcs. These results also show that each equivalent class of network structures can be represented by a *partially directed graph* (PDAG), where a directed $X \rightarrow Y$ denotes that all members of the equivalence class contain the arc $X \rightarrow Y$; and, an undirected edge X—Y denotes that some members of the class contain the arc $X \rightarrow Y$, and some contain the arc $Y \rightarrow X$. The score in [6] is structure equivalent in the sense that equivalent networks receive equal scores. In our experiments, we learn network structures and then use the procedure described in [2] to convert them to to PDAGs.

3 Using Bootstrap for Confidence Estimation

Suppose we are given N observations $D = \{\mathbf{x}[1], \ldots, \mathbf{x}[N]\}$, each an assignment of values to $\mathbf{X}$. Moreover, assume that these assignments where sampled independently from a probabilistic network B. Let G be the PDAG graph corresponding to B and let E be the set of edges in G. (Note that edges can be either directed edges $X \rightarrow Y$ or undirected edges X—Y.) Let $\hat{B}(D)$ be the induced network returned by some induction algorithm invoked with data D as input, and let $\hat{G}(D)$ be the corresponding PDAG structure. For any edge e consider the following quantity

$$p_N(e) = \Pr\{e \in \hat{G}(D) \mid |D| = N\}.$$

This is the probability of inducing a network that contains e among all possible datasets of size N that can be sampled from B.[1] If our induction procedure is *consistent*, then we expect that as N grows larger, $p_N(e)$ will converge to 1 if $e \in G$, and to 0 if $e \notin G$. The quantity $p_N(e)$ is a natural measure of the power of any induction algorithm. Our goal is to find estimates of $p_N(e)$ given only a single set of observations D of size N. This would mimic the usual induction situation when we want to learn a model from data.

We describe two possible algorithms: the non-parametric bootstrap and the parametric bootstrap. Application of the non-parametric bootstrap begins by re-sampling N times, with replacement, from the dataset D. This results in a sequence of instances which we label $D_1^* = \{\mathbf{x}_1^*[1], \ldots, \mathbf{x}_1^*[N]\}$. We then repeat this procedure m times, labeling the i^{th} replicate D_i^*. For each replicate we induce a PDAG $\hat{G}(D_i^*)$. We then define

$$p_N^{*,n}(e) = \frac{1}{m} \sum_{i=1}^{m} 1\{e \in \hat{G}(D_i^*)\}.$$

The parametric bootstrap is a similar process. Instead of re-sampling the data with replacement from the training

[1] Of course, there are nontrivial relationships between confidence estimates for different edges. For example $p_N(X \rightarrow Y) + p_N(Y \rightarrow X) + p_N(X—Y) \leq 1$.

data, we sample new datasets from the induced network $\hat{B}(D)$. We define $p_N^{*,p}(e)$ as above with respect to the m induced datasets. The parametric bootstrap estimates of $p_N(e)$ will converge under more general conditions than the non-parametric bootstrap, provided, of course, that the parameterization converges to the true underlying model at least asymptotically. On the other hand, if this last condition is not satisfied then no consistency claim can be made. The non-parametric bootstrap estimates require no such model consistency.

The consistency of the non-parametric bootstrap, however, requires uniform convergence in distribution of the bootstrap statistic as well as a continuity condition (in the parameters). Convergence of the parametric bootstrap will hinge on a continuity condition, and most importantly on the asymptotic consistency of the induction algorithm. Under these conditions, we claim that as m and n tend to ∞ then $|p_N(e) - p_N^{*,n}(e)|$ tends to 0 for all edges e. We are currently working on providing a thorough theoretical analysis of these conditions in the context of Bayesian network induction. One of the purposes of the preliminary experiments in the section below is to provide empirical support for this claim.

4 Bayesian Confidence Estimation

The Bayesian perspective on confidence estimation is quite different than the "frequentist" measures we discussed above. A Bayesian would compute (or estimate) the posterior probability of each feature. Via reasoning by cases this is simply:

$$\Pr(e \mid D) = \sum_G \Pr(G \mid D) 1\{e \in G\}. \qquad (1)$$

The term $\Pr(G \mid D)$ is the posterior of a structure given the training data, and for certain classes of priors, can be computed up to a multiplicative constant (where the constant is the same for all graphs G) [6].

A serious obstacle in computing this posterior is that it requires summing over a large (potentially exponential) number of equivalence classes. Heckerman et al. [7] suggest to approximate (1) by finding a set $\mathcal{G}$ of high scoring structures, and then estimating the relative mass of the structures in $\mathcal{G}$ that contains e.

$$\Pr(e \mid D) \approx \frac{\sum_{G \in \mathcal{G}} \Pr(G \mid D) 1\{e \in G\}}{\sum_{G \in \mathcal{G}} \Pr(G \mid D)}.$$

This raises the question of how we construct G. One simple approach for finding such a set is to record all the structures examined during the search, and return the high scoring ones. The set of structures found in this way is quite sensitive to the search procedure we use. For example, if we use greedy hill-climbing, then the set of structures we will collect will all be quite similar. Such a restricted set of candidates also show up when we consider multiple restarts of greedy hill-climbing and beam-search. This is a serious problem since we run the risk of getting estimates of confidence that are based on a biased sample of structures.

Method		Edges	All edges μ	All edges σ^2	$e \in G$ μ	$e \in G$ σ^2	$e \in G$, $p_N(e) > 0.75$ μ	$e \in G$, $p_N(e) > 0.75$ σ^2
1K	P	G	.062	.007	.067	.008	.051	.009
1K	N	G	.076	.005	.064	.004	.052	.004
1K	P	B	.069	.007	.067	.007	.046	.007
1K	N	B	.093	.006	.091	.006	.068	.006
10K	P	G	.042	.004	.036	.003	.021	.001
10K	N	G	.070	.006	.062	.006	.048	.005
10K	P	B	.075	.008	.073	.007	.047	.004
10K	N	B	.084	.005	.081	.005	.065	.003

Table 1: Expected difference $|p_N(e) - p_N^*(e)|$ and variance of these differences for the various experiments conducted. Experiments are designated by P or N for parametric or non-parametric bootstrap, and by G or B for greedy hill climbing or beam search, respectively.

One way of avoiding this problem is to run a fairly extensive MCMC simulation of the posterior of G. Then we might expect to get a more representative group of structures. This, procedure, however, can be quite expensive in terms of computation time. The bootstrap approach suggests a relatively cheap alternative—we can use the structures $\hat{G}(D_1^*), \ldots, \hat{G}(D_m^*)$ from the non-parametric bootstrap as our representative set of structures in the Bayesian approximation. In this proposal we use the re-sampling in the Bootstrap processes as way of widening the set candidates we examine. The confidence estimate is now quite similar to the non-parametric bootstrap of Section 3, except that structures in the bootstrap samples are weighted in proportion to their posterior probability.

5 Experimental Results

We used the "alarm" network [1] which has 37 random variables and 46 edges (only 4 of which are not compelled) as the data generating distribution B. We performed experiments with N (the number of instances in our data set) being $1,000$ and $10,000$. We choose $1,000$, since it is too small to recover the true network, but large enough to recover some of the more pronounced relations. Thus, we can examine how well the confidence estimates separate the parts of the network that are accurate from those that are not. On the other hand, $10,000$ is usually enough to recover a good approximation to the generating network. In this case we expect to be more confident about the true structure.

We used two search procedures in our tests. The first search procedure is greedy hill-climbing with random restarts. This procedure attempts to apply the best scoring change to the current network until no further improvement can be made. Once the hill-climbing procedure was stuck at a local maxima, it applied 20 random arc changes (addition/deletion/reversal) and restarted the search. The procedure was terminated after a fixed number of restarts. The second search procedure is beam search. This procedure keeps a queue of k candidates (in our experiments we choose $k = 90$). At each iteration the procedure se-

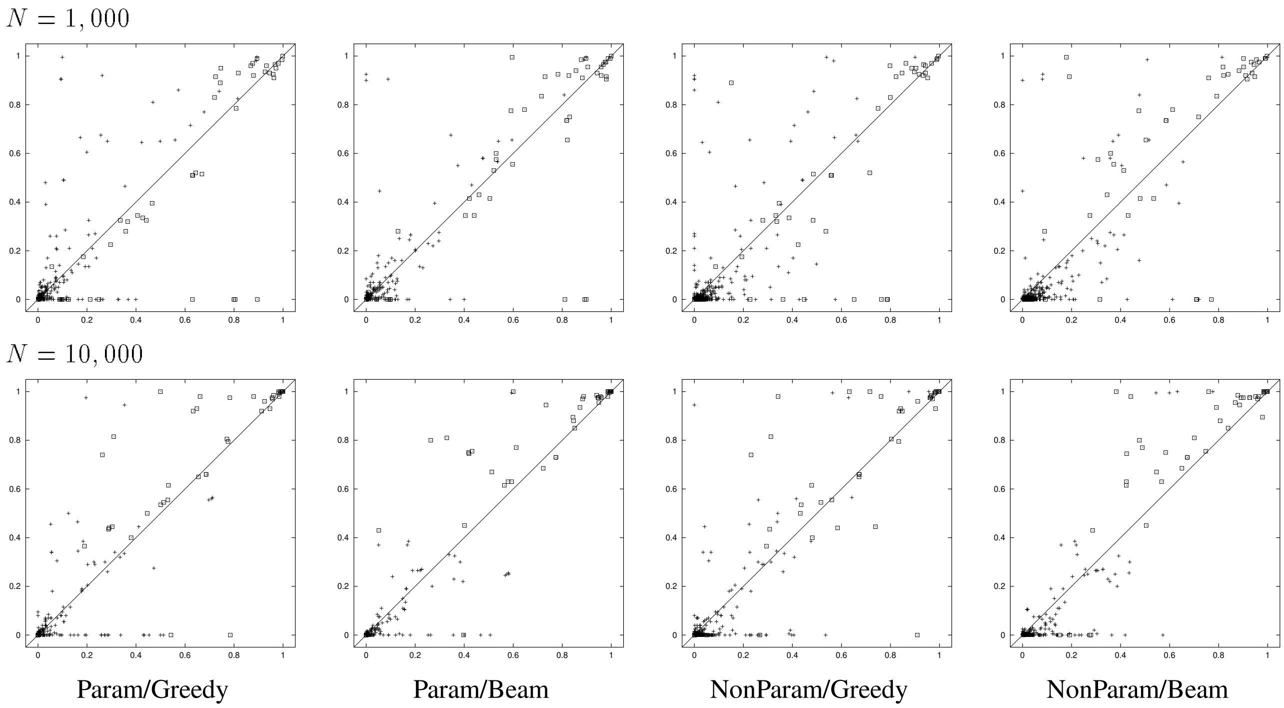

Figure 1: Scatter plots comparing the confidence reported by the bootstrap against the confidence from the golden model. The x-axis is the bootstrap confidence, and the y-axis is the estimate of the golden model confidence. Each edge is represented by a mark with the appropriate x and y coordinates. Edges that appear in the generating PDAG are marked as a box, while other edges are marked as a plus sign.

lects the best candidate on the queue, generate all it neighbors, and inserts ones that were not expanded before into the queue (keeping the best k among these new candidates and the previous ones on the queue). The procedure was terminated after 300 iterations without improvement. The main difference between these procedures resides then on the strategies used to escape local minima.

To estimate $p_N(e)$ we sampled 200 data sets of size N from the "alarm" network, induced 200 networks B', and computed $p_N(e) = \frac{1}{200}\sum_{i=1}^{200} 1\{e \in G'_i\}$. (The choice of $m = 200$ is based on the recommendations in [5].) We generated estimates of $p_N(e)$ for the two values of N and for the two search procedures.

We then compared the bootstrap estimates with (our approximation of) $p_N(e)$. We generated 10 data sets from "alarm", and applied both parametric and non-parametric bootstrap procedures to each dataset separately (for each search procedures). Each of these bootstrap procedures generated 200 networks, from which we computed our estimate $p^*_N(e)$. Figure 1 shows plots comparing $p^*_N(e)$ (averaged over the 10 repetitions) $p_N(e)$. The closer the points to the diagonal line the smaller the difference between the estimates. As we can see, edges with high confidence (above 0.75) have a general tendency to cluster around the diagonal line. There is bigger dispersion in the region between 0.2 and 0.6. Also, the estimates of $p^*_N(e)$ appear to be slightly pessimistic when $N = 10k$. In general, and specially for the case of $N = 10k$, if $p^*_N(e) > 0.75$ then $p_N(e) > 0.75$ which is reassuring.

To quantitatively summarize the errors in estimation and the variance among different bootstrap runs, we computed the expected difference $|p_N(e) - p^*_N(e)|$ and the variance of this difference across different training data sets. We report these computations in Table 1 for several cases: All edges, all edges in G (the generating model), and edges in G with $p_N(e) > 0.75$. The expectation is weighted by the actual confidence we need to assign to edges (i.e., $p_n(e)$), since errors in estimating the confidence of low-confidence edges is less important. We compute this weighted average according to $\frac{\sum_{i=1}^{i=v} |p_N(e_i) - p^*_N(e_i)| \times p_N(e_i)}{\sum_{i=1}^{i=v} p_N(e_i)}$, where v is the number of edges considered in the computation. Note that for edges with $p^*(e) > 0.75$, the difference is as low as 0.021 in the parametric case with $N = 10k$. Moreover, Figure 1 indicates that most of the $p^*(e) > 0.75$ estimates when $N = 10k$ are "pessimistic," guaranteeing a better confidence $p(e)$ in the generating model. As can be seen in Figure 2 even in the non-parametric case (where the difference is 0.047, all edges in this class (bold lines) also appear in the generating model.

Motivated by these observations, we decided to investigate whether we can use the estimate of confidence to decide whether an edge belongs to the generating model or not. A simple decision procedure is to rely on a threshold value of the confidence. That is, we classify an edge e as "true" if $p^*_N(e) > t$ for some threshold value. We can then examine the number of false positives (i.e., $p^*_N(e) > t$ but $e \notin G$) and false negatives (i.e., $p^*_N(e) < t$ but $e \in G$) for dif-

Threshold			0.60		0.75		0.90	
Method			fp	fn	fp	fn	fp	fn
1K	P	G	11.0	19.5	7.5	23.5	4.0	27.6
1K	P	B	8.8	14.5	5.9	18.8	3.5	25.9
1K	N	G	8.8	19.9	3.4	25.1	1.5	32.9
1K	N	B	8.4	19.8	3.7	24.7	1.1	33.0
1K	B	G	22.1	19.7	21.9	19.7	21.9	19.7
1K	B	B	13.3	12.3	13.1	12.5	12.8	12.8
1K	B*	G	7.8	22.4	3.1	26.9	1.4	34.0
1K	B*	B	8.1	19.8	3.6	24.8	1.1	33.0
1K	G	G	19.0	26.0	11.0	26.0	6.0	29.0
1K	G	B	9.0	18.0	6.0	22.0	5.0	26.0
10K	P	G	9.3	17.0	7.9	21.2	6.1	24.5
10K	P	B	8.7	14.1	7.4	18.0	6.5	21.1
10K	N	G	7.7	16.6	4.9	20.3	3.1	27.0
10K	N	B	6.4	14.4	4.2	19.3	2.2	27.3
10K	B	G	22.4	16.9	22.4	16.9	22.1	16.9
10K	B	B	13.0	10.5	13.0	10.5	12.7	10.7
10K	B*	G	7.2	16.7	4.7	20.4	2.7	27.7
10K	B*	B	5.1	17.6	3.0	21.5	2.0	29.0
10K	G	G	5.0	15.0	5.0	20.0	5.0	23.0
10K	G	B	4.0	7.0	4.0	15.0	4.0	23.0

Table 2: An analysis of the average number of misclassifications as a function of the threshold value. Reported numbers are "fp" for false positive, and "fn" for false negative. Methods are described by: number of instances, "1K" for $N = 1000$ and "10K" for $N = 10000$; type of estimation, "P" for parametric, "N" for non-parametric, "B" for Bayesian, "B*" for Bayesian weighting of non-parametric bootstrap, and "G" for the estimate of the "golden" confidence $p_N(e)$; and search method, "G" for greedy hill-climbing, and "B" for beam search.

ferent bootstrap runs. Table 2 reports the number of false positives and negatives when we classify based on the estimates. These numbers are averaged over the 10 bootstrap runs. The table contains the classification errors for both the frequentist estimates and the two Bayesian methods we described above. The Bayesian methods differ on the networks considered in their respective computation: the first collects networks during the search, and the other relies on the non-parametric bootstrap for the networks.

There are several things to note. First, these results indicate that some of the misclassification are due to problems with the learning procedures (i.e., either the scoring metric and prior we used or the search procedures are directly responsible). To see this, note that the classification errors based on $p_N(e)$, the estimate based on the generating model, involve at least 4 false positives at all threshold levels and both training set sizes. This means that such edges appeared in most of the networks learned from samples of the domain. We stress that missing a single edge in the induction can cause this phenomena, since it can cause the arc direction of other edges not to be compelled. We are currently investigating the source of this error. These results also indicate that beam search usually performs better than greedy hill-climbing.

Second, these results demonstrate the Bayesian approach that collects networks during the search is not sensitive to the threshold value used in the classification process. A closer inspection of the estimates generated by this approach showed that it assigns extreme values, either close to 1 or close to 0 to the edges, as predicted by our discussion above.

Third, by having more data, we usually reduce the number of errors. However, for a fixed threshold value the number of false positive increases in several cases. This is mainly due to the fact that with ten times more data, we are more confidence in edges that appear in all the networks. Although, the increased number of false positives seems counterintuitive, we note that the relative number of misclassified edges among the edges with confidence higher than a threshold is smaller in most of these cases.

These results indicate that there is nontrivial correlation between high confidence in an edge, say above 0.75, and it being in the generating model. However, we must caution that counting edges can be misleading, since we are ignoring the strength of the dependency between variables and other features of the distributions. We also suspect that other features, such as membership in the Markov blanket of a variable, or neighborhood relations in the graph, might be more reliably predicated based on the bootstrap methods. We are currently investigating this issue.

6 Discussion

The purpose of this paper was to examine the applicability of the Bootstrap to estimate confidence in learned networks. We focused on the structural properties of the networks, specifically on the confidence of each edge on the related PDAG. Our preliminary results are encouraging in that they indicate that the bootstrap confidences are correlated with estimates on the generating model. Moreover, the great majority of edges that were labeled with high confidence (greater than 0.75) were indeed present in the generating network. Our experiments also uncovered important differences in performance between the search methods, when we examine the structural properties of the induced networks. Beam search was consistently better in the classification experiments depicted in Table 2. Yet, the confidence estimates for the case of $N = 10k$ where more accurate when a greedy hill-climbing strategy was used (Table 1.

There are several directions for future work. We are interested in a theoretical analysis of the Bootstrap in the context of Bayesian network induction. Also, our results are preliminary and much more experimentation is needed on different features of the structure of the network, and on the search methods. An important direction is to find principled methods for incorporating the cues provided by the Bootstrap confidence measures into the search procedure. For example, in some of our examples edges that were not present in the learned model received high confidence in the bootstrap test.

Our final goal is twofold: first, to be able to provide guarantees about the properties of the induced model, and second, to find better ways to guide the process of structure discovery. Although current induction methods for learning Bayesian networks are reliable for density estimation [6],

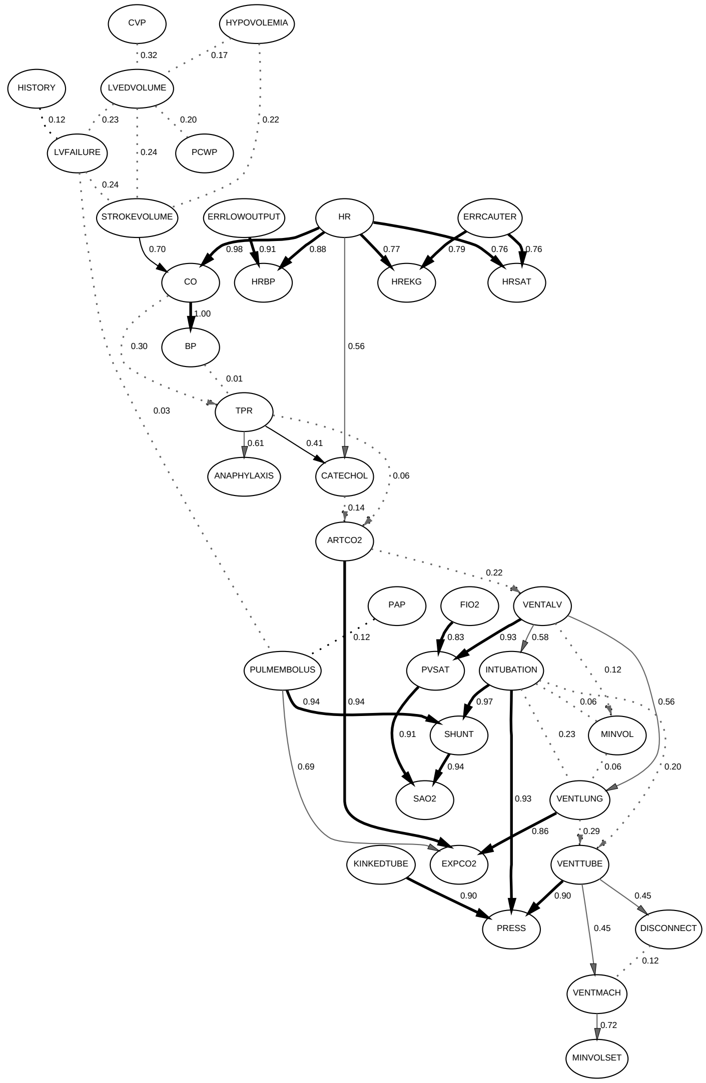

Figure 2: PDAG of network induced with greedy hill-climbing annotated with confidence measures computed by non-parametric bootstrap. The drawing style of edges (e.g., bold, plain, and dashed) correspond to the estimated level of confidence. The color of edges denotes whether they appear in the "generating" network (black) or not (gray).

taken as a whole, the empirical results in this paper indicate that there is definite room for improvement on their performance for structure identification.

Acknowledgements

Some of this work was done while Nir Friedman and Abraham Wyner were at the University of California at Berkeley. Part of the experiments reported here were run on the NOW cluster at UC Berkeley. We thank the NOW group for allowing us to use their resources.

References

[1] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. 2'nd European Conf. on AI and Medicine*, 1989.

[2] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *UAI '95*, pages 87–98. 1995.

[3] D. M. Chickering. Learning Bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, eds., *Learning from Data: Artificial Intelligence and Statistics V*, 1996.

[4] R. G. Cowell, A. P. Dawid, and D. J. Spiegelhalter. Sequential model criticism in probabilistic expert systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15:209–219, 1993.

[5] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*, 1993.

[6] D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[7] D. Heckerman, C. Meek, and G. Cooper. A Bayesian approach to causal discovery. Technical Report MSR-TR-97-05, Microsoft Research, 1997.

[8] C. Meek. Causal inference and causal explanation with background knowledge. In *UAI '95*, pages 403–410. 1995.