Improving Accuracy of Constraint-Based Structure Learning

Andrew Fast, Michael Hay and David Jensen
University of Massachusetts Amherst
Department of Computer Science
140 Governors Drive, Amherst MA 01002
{afast,mhay,jensen}@cs.umass.edu

Abstract

Hybrid algorithms for learning the structure of Bayesian networks combine techniques from both the constraint-based and search-and-score paradigms of structure learning. One class of hybrid approaches uses a constraint-based algorithm to learn an undirected skeleton identifying edges that should appear in the final network. This skeleton is used to constrain the model space considered by a search-and-score algorithm to orient the edges and produce a final model structure. At small sample sizes, the performance of models learned using this hybrid approach do not achieve likelihood as high as models learned by unconstrained search. Low performance is a result of errors made by the skeleton identification algorithm, particularly false negative errors, which lead to an over-constrained search space. These errors are often attributed to "noisy" hypothesis tests that are run during skeleton identification. However, at least three specific sources of error have been identified in the literature: unsuitable hypothesis tests, low-power hypothesis tests, and unexplained d-separation. No previous work has considered these sources of error in combination. We determine the relative importance of each source individually and in combination. We identify that low-power tests are the primary source of false negative errors, and show that these errors can be corrected by a novel application of statistical power analysis. The result is a new hybrid algorithm for learning the structure of Bayesian networks which produces models with equivalent likelihood to models produced by unconstrained greedy search, using only a fraction of the time.

1 Introduction

Algorithms for learning the structure of Bayesian networks usually fall into one of two broad categories: *constraint-based* algorithms and *search-and-score*. Constraint-based algorithms use a series of statistical decisions to identify structures that are consistent with the conditional independencies entailed by the training data. Search-and-score techniques treat structure learning as an optimization problem using a heuristic search technique to find structures that maximize the desired scoring metric. Tsarmardinos et al. [14] recently introduced a two-phase hybrid approach for structure learning to combine the efficiency of constraint-based algorithms with the consistent high performance of search-and-score algorithms. Under this approach, during the first phase, called *skeleton identification*, a constraint-based algorithm is used to identify the *skeleton* of the learned Bayesian network. A skeleton is set of undirected edges indicating possible dependence among variables in the final network. During the second phase, or *heuristic search phase*, a search-and-score algorithm is used to determine whether edges appearing in the skeleton will be included in the final model, and if so, the orientation of that edge.

Skeleton identification algorithms use a series of local statistical decisions to efficiently identify conditional independence relations among variables appearing in the training data [5, 11, 14]. If two variables can be shown to be conditionally independent, then there should not be an edge connecting those variables in the final model structure, and possible structures that contain that edge can be safely excluded from further consideration. When two variables cannot be shown to be conditionally independent, skeleton identification algorithms add an edge to the skeleton between those variables.

Since skeleton identification algorithms reduce the number of possible structures considered by the heuristic search phase, skeleton identification can be viewed as providing constraints limiting the heuristic search. Ideally, if skeleton identification is able to identify many conditional independence relations, then the search space considered by heuristic search is reduced, leading to dramatic improvements in search efficiency. However, if too many edges are removed from the skeleton, then the search can become over-constrained, making it impossible to identify high-scoring model structures.

Over-constrained search is often the result of errors made during skeleton identification, particularly *false negative errors*. Such errors occur when an edge appearing in the true network is erroneously removed from the skeleton during skeleton identification. Although often attributed to noise in training samples [2, 12], false negative errors can arise from one of three causes identified in prior work on learning the structure of Bayesian networks. *Unsuitable hypothesis tests* occur when the data being tested does not conform to the requirements and assumptions of the hypothesis test used to determine independence [11, 14]. *Low-power statistical tests* can fail to detect a dependence in the data even when the dependence exists in the true model [11, 14]. And *Unexplained d-separation* results from hypothesis tests that produce inconsistent results, often as a result of errors in previous tests [5, 12].

No work has examined these sources of error in combination to identify their relative importance and the interactions among possible solutions. In this paper, we determine the relative importance of each of these sources of error and evaluate possible corrections for false negative errors, including the first correction for low-power statistical tests based on statistical power analysis. We show that low-power tests are the primary source of false negative errors produced by skeleton identification algorithms and that these errors can be corrected with a novel application of statistical power analysis. Including this correction during skeleton identification results in a hybrid algorithm that learns models with likelihoods that are statistically indistinguishable from models learned by unconstrained greedy search, but in significantly less time on most datasets.

2 Background

2.1 Two-Phase Hybrid Algorithms

The goal of structure learning algorithms is to identify from data the presence and orientation of edges in the Bayesian network. A Bayesian network is a directed, acyclic, graphical model of a joint probability distribution over the variables appearing in a dataset. The edges in the graph represent probabilistic dependence between variables. Structure learning for Bayesian networks has been extensively studied; for additional background information see Pearl [10] and Buntine [3].

We focus on two-phase hybrid algorithms. These hybrid algorithms differ from full constraint-based algorithms only in the choice of edge orientation; constraint-based algorithms use deterministic edge orientation rules whereas hybrid algorithms use a heuristic search procedure to produce a final model [5, 11, 14]. Therefore, any constraint-based algorithm (without edge orientation) can be paired with a generic constrained search procedure to create a hybrid algorithm.

2.1.1 Skeleton Identification

There are many different varieties of skeleton identification algorithms appearing in the literature. To describe some of the main design decisions of these algorithms we highlight the differences between three prototypical skeleton algorithms: PC [11], Max-Min Parents Children (MMPC) [14], and Three-Phase Dependency Analysis (TPDA) [5]. The PC[1] algorithm is a prototypical skeleton algorithm that runs hypothesis tests in increasing order of conditioning set size trying all pairwise tests first, followed by tests conditioned on a single variable, and so on until no more tests can be run. For categorical data, the PC algorithm uses a G^2 statistic to determine independence [11]. G^2 is asymptotically distributed as χ^2. PC uses a *rule of thumb* to determine whether to continue running tests. The rule of thumb states that the G^2 test is reliable if there are five or more instances per degree of freedom of the test. If the test is not reliable, PC makes a *default decision* to include the edge in the skeleton. MMPC was recently introduced as the skeleton phase of the Max-Min Hill Climbing algorithm, the first example of a two-phase hybrid algorithm [14]. MMPC is similar to PC in every way except that it runs all reliable tests for a single target variable before considering other variables, choosing variables to add to the conditioning set with the Max-Min Heuristic [14]. TPDA uses a different approach to learn a skeleton than either PC or MMPC [5]. Rather than using classical hypothesis tests, TPDA relies on tests of mutual information to determine independence. As its name implies, TPDA operates in three phases. At each phase, the algorithm considers pairs of variables and either adds or removes an edge depending on the conditional mutual information score. TPDA restricts its conditioning variables to those variables that appear on an undirected path between the variables being tested. Unlike PC and MMPC, TPDA does not make any determination of test reliability, instead choosing to run every hypothesis test.

We chose to consider these three algorithms because they represent three different strategies for learning undirected skeletons from data and all three are widely used or have been shown to perform well in comparison with other structure learning algorithms. The PC algorithm is widely used; the textbook describing the PC algorithm currently has been cited over 1400 times [2]. In addition, many variants of the PC algorithm appear in the literature [1, 15]. A hybrid algorithm using MMPC has been shown to outperform six leading non-hybrid structure learning algorithms on many datasets with varying characteristics [14]. The TPDA algo-

[1] PC is named for its creators Peter (Spirtes) and Clark (Glymour)

[2] Citations according to http://scholar.google.com as of July 7, 2008

rithm is also widely used; the software package containing the TPDA algorithm has been downloaded over 2000 times [5].

2.1.2 Heuristic Search Phase

For the heuristic search phase, our experiments use a greedy hill-climbing algorithm with a tabu list. The search operators search all possible edge additions, deletions, and reversals from the current network. We use a BDeu score as the metric, though any likelihood or penalized likelihood metric could be used. Greedy hill-climbing is simple, easy to implement, and generally performs quite well; in fact, it is often considered to be state-of-the-art for Bayesian network learning [13]. Any search algorithm that can be constrained to the skeleton could be used for the heuristic search phase. If skeleton identification produces a fully-connected graph, then the heuristic search phase is equivalent to unconstrained greedy search.

2.2 Hypothesis Tests

By definition, constraint-based skeleton algorithms use a series of hypothesis tests to determine whether an edge should be added to the skeleton. A hypothesis test specifies a null hypothesis defining the distribution we would expect if the variables were truly independent. The significance threshold is defined in terms of the probability of the observed correlation or of a more extreme value being observed under the null hypothesis. For classical hypothesis tests, the standard significance threshold is $p <= 0.05$; if the probability of the observed value is 0.05 or less then the null hypothesis is rejected. The significance threshold bounds the rate of type I errors, called α, of a hypothesis test. Type I errors are incorrect rejections of the null hypothesis when it is true. The type II error rate, β, is of particular interest for skeleton identification. A type II error occurs when the null hypothesis is incorrectly accepted when it is false.

The probability of detecting a significant effect given that it exists in the data is $1 - \beta$. This value is also called the *statistical power* of the test and is useful for determining whether to run a test in structure learning for Bayesian networks. Statistical power depends on the sample size N, the degrees of freedom of the test, the significance threshold α, and the expected effect size w [6]. The *effect size* of a test defines a specific alternative hypothesis to compare against the null and indicates the minimum strength of correlation that is detectable by the hypothesis test. Since w is unfamiliar to most researchers, Cohen [6] suggests values of w for small ($w = 0.1$), medium ($w = 0.3$), and large ($w = 0.5$) effects.

3 Errors in Skeleton Identification

Since skeleton identification algorithms rarely operate in the sample limit, errors are bound to occur. As with hypothesis tests, there are two kinds of errors made during skeleton identification: false positive and false negative errors. A false positive error is made when an edge not appearing in the true network is added to the skeleton. This type of skeleton error could be caused by a type 1 error of the hypothesis test or by other means, such as a default decision to add an edge if the hypothesis test is determined to be unreliable due to insufficient data. False negative errors are most frequently due to type II errors in hypothesis tests but could also be caused by inconsistencies between the results of hypothesis tests. The potential outcomes of assessing both test reliability and statistical significance are shown in Figure 1.

In hybrid algorithms, false negative errors are much more costly than false positive errors. Once an edge has been erroneously removed from the skeleton it cannot be corrected by the heuristic search phase. In contrast, a false positive error can still be corrected by the heuristic search phase. In general, the goal of hybrid algorithms is to produce a high-likelihood model while reducing the runtime of heuristic search by constraining the possible search space. If skeleton identification produces too many false negative errors combined with few false positive errors, then the search space becomes over-constrained and may exclude high-quality networks. In contrast, if too few edges are removed, then the network is under-constrained and the hybrid approach does not lead to decreased runtimes. The ideal skeleton identification algorithm would be a “conservative” approach that would add a superset of the correct edges to the skeleton to avoid over-constrained search but not so many edges as to increase the runtime of heuristic search.

Prior work in structure learning has identified three sources of false negative errors: (1) unsuitable hypothesis tests, (2) low-power hypothesis tests, and (3) unexplained d-separation. In categorical data, *unsuitable hypothesis tests* occur when the expected frequencies in some of the cells of the contingency table are small, either due to small sample sizes or large contingency tables [11, 14]. When this occurs,the G^2 statistic is known to deviate from the χ^2-distribution resulting in inaccurate p-values [8]. Even if the hypothesis test is suitable for the data, *low-power statistical tests* may result in false negative errors. The power of a hypothesis test depends on a combination of the degrees of freedom of the test, the sample size, and the effect size appearing in the data (See Section 2.2). *Unexplained d-separation* produces a false negative error when a variable (or set of variables) can be used to show that two variables are independent, but that variable does not appear on

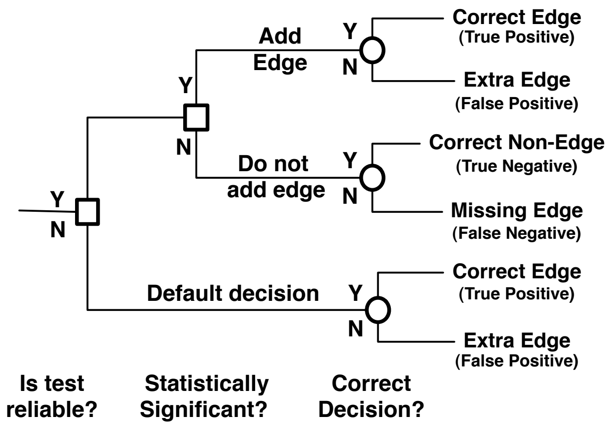

Figure 1. Determining which edges are included in the skeleton. A hypothesis test can be unreliable if the test is unsuitable for the data or has low statistical power.

a path between the variables being tested [5, 12]. This follows from d-separation rules that define conditional independence relationships is terms of the structure of the graphical model. Pearl [10] provides more information about the possible d-separation relationships.

Based on an empirical comparison of existing hybrid algorithms on data generated from a standard benchmark dataset, we observe that hybrid algorithms almost always produce models with lower likelihood than models produced by greedy search, although the runtimes are much faster than greedy search. This provides evidence that existing skeleton identification algorithms over-constrain search. Figure 2 shows the comparison of the performance of our implementation of the PC, MMPC, and TPDA skeleton algorithms learning the structure of the ALARM network. The ALARM network is a standard benchmark dataset for Bayesian network structure learning available from the Bayesian Network repository[3]. We conjecture that the apparent over-constrained search produced by existing skeleton algorithms is a result of failing to control for all sources of false negative errors. On the ALARM network, skeleton identification algorithms exclude between 20% and 33% of the true edges from the skeleton with only 500 samples.

4 Corrections for False Negative Errors

The goal of this work is to improve the likelihood of the hybrid approaches by reducing the number of false negative errors produced by skeleton identification algorithms. There are two primary approaches appearing in the literature to correct false negative errors: the rule of thumb and the necessary path condition. In addition, we describe a novel correction, called the POWER correction, to prevent false negative errors due to low-power hypothesis tests.

4.1 The Rule of Thumb Correction

The rule of thumb correction used in both the PC and and MMPC skeleton identification algorithms is sufficient to account for errors due to unsuitable hypothesis tests. The correction is motivated by Monte Carlo studies comparing the p-value of the G^2 statistic with a χ^2 test against the exact p-value produced using computationally intensive statistics to generate the sampling distribution [8]. These studies showed that the G^2 statistic is not a suitable approximation of the χ^2 distribution and does not produce accurate p-values if ratio of the sample size and the degrees of freedom falls below 5. Consequently, the rule of thumb correction is based on this ratio and prevents the running of a hypothesis test if the test is unsuitable, that is, if the ratio of the sample size to the degrees of freedom of the test is less than 5.

Although the rule of thumb correction is intended to also prevent errors due to low-power statistical tests, it can provide only a weak bound on the statistical power. The rule of thumb and other approaches of limiting the size of the contingency table, such as limiting the size of the conditioning set [15], do not account for all of the factors that determine statistical power. In particular, they do not account for the possible effect sizes present in the data. If the effect size we wish to detect is small, then the test could have low statistical power and produce false negative errors despite using the rule of thumb correction. The minimum power permitted under the rule of thumb for small effect sizes are shown in Figure 3. Our experiments show that effect sizes actually occurring in the benchmark data result in low-power statistical tests under the rule of thumb (See Section 5.3).

4.2 The POWER Correction

To account for potentially small effect sizes in the data, we developed a novel correction for low-power statistical tests based on statistical power analysis [6]. Statistical power analysis provides an analytical framework for computing the exact statistical power of the test $(1 - \beta)$, given an accurate estimate of the expected effect sizes in the data. For the POWER correction, the estimated effect size can be supplied by the user or estimated using cross-validation. The POWER correction is easy to implement and can be used in any skeleton identification algorithm that uses the rule of thumb, including PC and MMPC. In addition, the effect size parameter provides a knob for varying the performance of skeleton identification algorithms between constrained and unconstrained skeletons.

[3] http://compbio.cs.huji.ac.il/Repository/

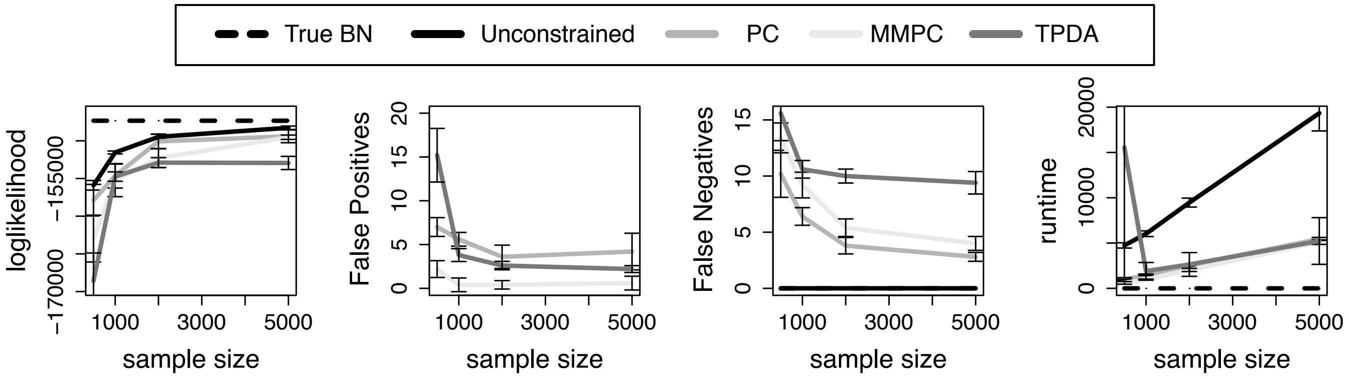

Figure 2. Comparison of hybrid algorithms using PC, MMPC and TPDA skeleton algorithms on the Alarm network. There are 46 edges in the true network. Error bars indicate a 95% confidence interval around the mean. The likelihood of the hybrid algorithms is less than the likelihood of greedy search (with the exception of the PC algorithm at 2000 samples), but runtimes of the hybrid algorithms are also significantly less.

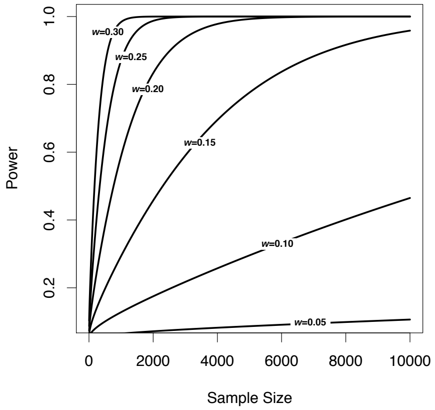

Figure 3. Minimum statistical power permitted under the rule of thumb.

4.2.1 Statistical Power Threshold

Like the rule of thumb, the POWER correction defines a limit on the acceptable ratio between the degrees of freedom of the test and the sample size. Since the sample size N is fixed for a particular test, the POWER correction determines whether a test with the given degrees of freedom has sufficient statistical power. The desired level of statistical power can be determined by the user. We set the desired power level to be 0.95. This corresponds to β of 0.05, matching the standard level for α. When N, α, the degrees of freedom, and effect size w are specified, it is possible to compute the statistical power of the test [6]. In skeleton identification algorithms, the value of N and the degrees of freedom are determined by the data and the specific test, respectively. The values of α (typically 0.05) and w are set by the user before running the algorithm. Recall, the power of a statistical test is $1-\beta$, where β is the probability of rejecting the alternative when it is true. The statistical power corresponds to the area under the alternative distribution that exceeds the critical value specified by α. For categorical data, the alternative distribution is specified by a noncentral χ^2 distribution with noncentrality parameter, $\lambda = w^2N$ [6]. The evaluation of $1-\beta$ requires an infinite summation of the noncentral χ^2 distribution [9]. We use the efficient two-stage approximation identified by Milligan when computing power [9]. Other implementations of power calculations are found in many common statistical packages (e.g., R [4]).

In practice, rather than computing statistical power for every test, we determine a range of degrees of freedom corresponding to high-power tests. Since power is inversely proportional to the degrees of freedom, we identify the threshold by computing the statistical power for every possible degree of freedom starting at one and increasing until the statistical power falls below our desired level. The computation is efficient and can be performed quickly before starting structure learning. We chose this approach due to its simplicity of implementation; many existing algorithms already make reliability decisions based the degrees of the freedom of the test, we simply substitute a new threshold based on statistical power. For categorical data, the degrees of freedom is $(r-1)(c-1)d$, where r and c are the number of distinct values taken on by the two variables whose dependence is being tested, and d is the number of possible joint values of the set of conditioning variables.

4.2.2 Choosing Effect Size with Cross-Validation

For the power threshold to be effective, the value of w must be set as close to the true minimum effect size as possible. If w is set too high, then the algorithm risks false negative errors due to low-power tests. If w is set too low, then the search may not be as efficient as possible as few edges can be safely removed from the skeleton. Fortunately, there are boundaries on the useful values of w. If the goal is constraining search at all, the minimum value of w should be large enough to run tests with a single degree of freedom. This minimum varies with the available sample size. To ensure that our new threshold is an improvement over the existing approaches, the maximum value of w must fall below the value corresponding to the effect detectable by the rule of thumb with the desired level of power.

We use ten-fold cross-validation to determine the optimal value of w for each skeleton identification algorithm. Following the standard cross-validation procedure, we divide each training set into ten folds. We then run both phases of the hybrid algorithm (skeleton identification and heuristic search) to learn a Bayesian network from training data composed of nine of the folds, and then compute the likelihood of data contained in the last fold given the learned model. We repeat this procedure for a range of w at each sample size.

4.3 Necessary Path Correction

The necessary path condition requires that for a variable z to d-separate x and y then z must fall on an undirected path between x and y [12]. Enforcing this condition leads to fewer variables being considered as possible conditioning variables, which results (at least in theory) in fewer edges being removed from the skeleton. Abellan et al. [1] use a stricter form of the necessary path condition that only conditions on variables in the minimum cut set appearing between the two variables.

The necessary path condition is used by the TPDA algorithms and at least two variants of the PC algorithm (but not the original) to prevent false negative errors due to unexplained d-separation [1, 5, 12]. To enforce the path condition, the skeleton identification algorithm must maintain a superset of all the edges that could be included in the skeleton. Both TPDA and PC maintain a superset of the possible skeleton edges. MMPC uses a depth-first approach which does not maintain a superset of the paths between two variables. Steck and Tresp [12] use the necessary path condition to identify inconsistent regions produced by the PC algorithm, but do not return a single model.

4.4 Other Approaches

An alternative approach for correcting false negative errors is to use a different type of hypothesis test such as tests of mutual information or tests using a Bayesian score such as BDeu [1, 5]. These tests typically use a weak significance threshold, such as determining whether the score is greater than zero, to determine independence. Unlike the POWER correction, these approaches do not permit a statistical bound on false negative errors. A statistical bound is necessary for determining the expected rate of false negative errors, which is critical for improving performance. Hutter [7] shows that point estimates of mutual information, such as those used by TPDA, are inaccurate and the that consideration of the second-order distribution is necessary to improve accuracy.

5 Experimental Results

We ran a series of experiments to determine the importance of each correction individually and in combination. The corrections we considered are listed in Table 1. Each correction was applied to both the PC and MMPC skeleton identification algorithms, with the exception of the necessary path correction, which was only applied to the PC algorithm. The weak correction was included as a baseline correction. If no correction was applied then every possible test would be run. This is neither feasible due to runtime considerations nor sensible as conditioning on many variables would likely result in a conclusion of independence for all pairs of variables.

Table 1. Corrections for false negative errors.

Weak Correction	Permit tests with at least 1 instance per degree of freedom.
Rule of Thumb	Permit tests with at least 5 instances per degree of freedom.
POWER	Run tests with sufficient power (with estimated effect size parameter).
Necessary Path	Only condition on variables on a path (PC Only).

5.1 Benchmark Data Considered

To evaluate the performance of the various corrections, we consider five networks from the Bayesian Network Repository. These five networks cover a number of different domains such as medical diagnosis, insurance risk, meteorology, and agriculture (see Table 2). For each network, we generated random samples of 500, 1000, 2000, and 5000 samples for use in training. To assess how well the learned structure approximates the generating distribution, we measured the log-likelihood of a large test sample (500,000 instances). Performance measurements were averaged over five training samples at each sample size.

Table 2. A summary of the Bayesian networks used.

Network	No. Vars.	No. Edges	In/Out Degree	Domain Size (Avg.)
Alarm	37	46	4/5	2-4 (2.8)
Barley	48	84	4/5	2-67 (8.8)
Hailfinder	56	66	4/16	2-11 (4.0)
Insurance	27	52	3/7	2-5 (3.3)
Mildew	35	46	3/3	3-100 (17.6)

5.2 Implementation Details

We re-implemented both the PC and MMPC skeleton algorithms (and heuristic search with tabu lists for the search phase) in our own Java package so that we could easily incorporate the corrections to both algorithms. Every effort was made to reproduce the original algorithms as described. When possible, we compared the results of the original software to our re-implementation and found no substantial differences in performance.

5.3 Determining the Effect Size Parameter Via Cross-Validation

Although cross-validation produces good estimates of the minimum effect size w, it not does not permit fast learning of Bayesian network structure. To avoid running cross-validation every time we wish to run structure learning, we learned the optimal values of w for each sample size on two randomly selected benchmark datasets: INSURANCE and MILDEW.

We then used cross-validation to determine the optimal setting of w for each of those datasets at each sample size. We considered values of w between the largest effect size that results in an unconstrained skeleton (i.e., no hypothesis will be run) and the effect that corresponds to the rule of thumb. In addition to these endpoints, we considered three values equally spaced between the minimum and maximum effect sizes. The best effect size at each sample size is shown in Figure 4. To compute a suggested effect size for other datasets, we averaged the results at each sample size over the two algorithms. The suggested values of w are shown in Table 3.

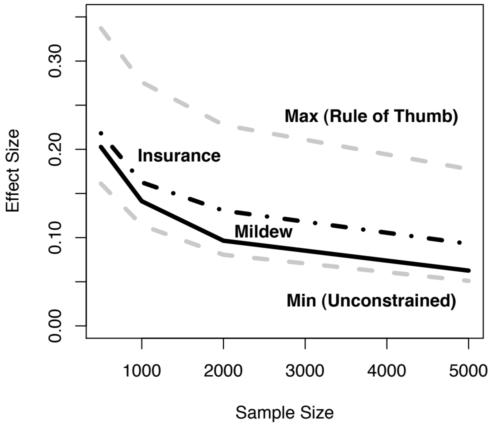

Figure 4. Results of cross-validation to select the best effect size parameter. The dashed grey lines indicate the outer limits considered during cross-validation. The minimum indicates the largest effect size where no tests are run and the maximum indicates the tests that are run under the rule of thumb threshold.

5.4 Evaluating Corrections

We then compared the number of false negative errors resulting from each correction on the remaining three datasets (see Table 4). We found that applying the POWER correction using the suggested effect size parameters resulted in a significant decrease in false negative errors across all three

Table 3. The effect size parameters chosen via cross-validation.

Sample Size	Suggested Effect Size
500	0.2183
1000	0.1518
2000	0.1204
5000	0.0766

datasets when compared to the rule of thumb. Using the necessary path correction resulted in a significant decrease in false negatives on only the ALARM dataset. Using the rule of thumb resulted in a significant decrease in false negatives over the weak correction on two of the three datasets. Since the suggested effect size parameters are consistently below the effect size parameter corresponding to the rule of thumb, the POWER correction subsumes the rule of thumb correction.

Table 4. Number of false negative errors after applying corrections. Combined results of PC and MMPC skeleton identification. Results are averaged over five training sets at each sample size. Bold text indicates a significant reduction compared to the rule of thumb. Italics indicate a significant increase from the rule of thumb. The necessary path correction can only be applied using the PC algorithm. Differences were significant at the 0.05 level using a t-test.

Dataset	Correction	500	1000	2000	5000
Alarm	Weak	13.1	8.5	4.9	3.4
	Rule of Thumb	11.8	7.8	4.6	3.4
	POWER	**6.1**	**3.9**	**3.0**	**2.1**
	RoT + Path	**7.4**	**4.6**	**2.6**	**2.0**
	POWER + Path	**5.4**	**3.6**	**2.4**	**2.0**
Barley	Weak	*30.0*	*23.4*	*22.6*	*20.4*
	Rule of Thumb	15.0	13.8	16.4	14.4
	POWER	**1.2**	**1.6**	**4.4**	**3.8**
	RoT + Path	12.2	11.8	15.2	14.4
	POWER + Path	**1.2**	**1.6**	**4.4**	**3.8**
Hailfinder	Weak	*15.6*	*13.6*	*10.8*	*6.7*
	Rule of Thumb	13.3	9.6	8.0	4.3
	POWER	**2.9**	**1.6**	**2.8**	**1.6**
	RoT + Path	12.8	9.0	7.4	4.0
	POWER + Path	**3.0**	**1.6**	**2.8**	**1.6**

We also compared a two-phase hybrid algorithm using the PC algorithm with a combination of the POWER and necessary path corrections to unconstrained greedy search and the PC algorithm using only the rule of thumb (see Figure 5). In addition to a significant reduction in false negative errors, we found that using the POWER correction also resulted in models with significant increases in likelihood in two datasets over models produced using only the rule of thumb. The likelihood of the models learned with the POWER correction are statistically indistinguishable from the likelihood of model learned with unconstrained greedy search. On the ALARM and HAILFINDER datasets, using the POWER correction also resulted in a significant decrease in runtime. The number of false positive errors shown in Figure 5 indicates that using the POWER correction does not result in a completely connected skeleton. On the BARLEY network, using the POWER correction does result in an increase in likelihood of the final model, but does not constrain the model space enough to result in a decrease in runtime over unconstrained search. To achieve faster runtimes at the expense of likelihood, it would be necessary to increase the minimum effect size parameter from the suggested values.

6 Conclusions

In this paper, we show that low-power statistical tests are the largest source of false negative errors from skeleton identification algorithms. We describe POWER, a correction for false negative errors that uses a novel application of statistical power analysis to correct for errors due to low-power tests. This correction results in a significant reduction in false negative errors caused by skeleton identification, subsumes the rule of thumb threshold previously used to correct these errors, and can be combined with other corrections such as the necessary path correction. The POWER correction is easy to implement and improves any skeleton identification algorithm that previously used the rule of thumb.

The POWER correction relies on the proper setting of an effect size parameter w to achieve these improvements. We present the results of cross-validation experiments to learn suggested values of w. We also show that these suggested values of w generalize across datasets. In addition, the w parameter provides a principled parameter for determining the trade-off between runtime and likelihood of two-phase hybrid algorithms. Although the suggested values of w can result in a significant decrease in runtime, it is possible that the suggested values may not provide many constraints on search. If long runtimes are a concern, increasing the suggested effect sizes would result in stronger constraints and improved runtimes, at the expense of the likelihood of the models.

We show that a two-phase hybrid algorithm incorporating the POWER and necessary path corrections is able to produce models with equivalent likelihood to models produced by unconstrained greedy search. These corrections resulted in a significant reduction in runtime on two of the three test datasets. These results indicate that hybrid approaches which constrain search using skeleton identification algorithms are an efficient alternative to unconstrained greedy search.

The suggested effect size values presented in this paper work well as a general starting point when considering applying the POWER correction. However, we would like to explore efficient alternatives to cross-validation for determining a setting of w that is best for a particular dataset. An alternative would likely consider the cardinality of vari-

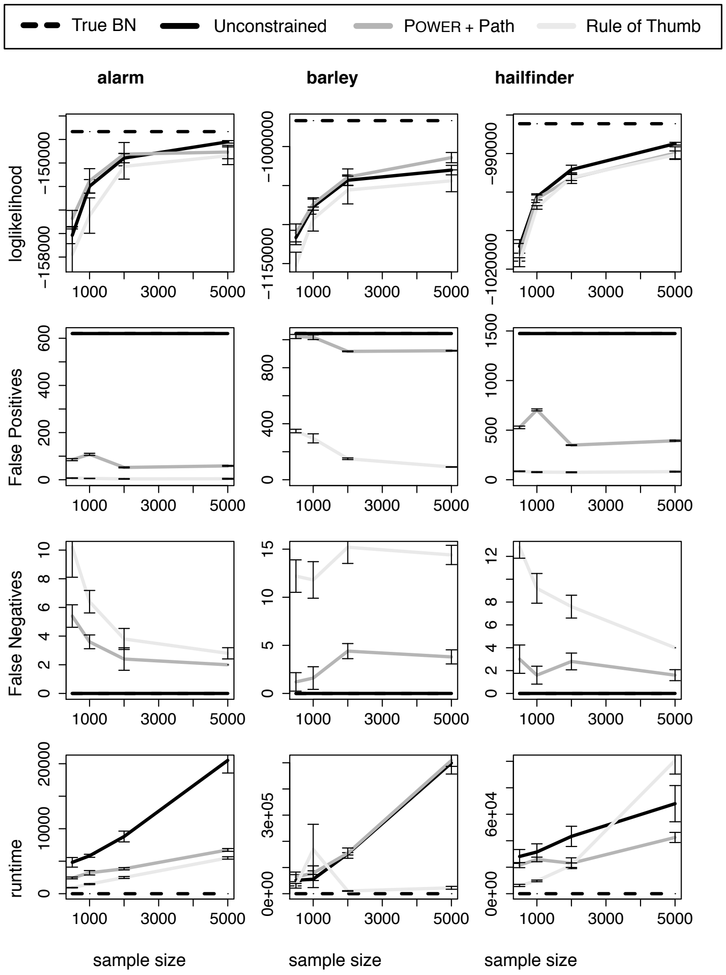

Figure 5. Comparing the combination of the Power and necessary path correction with the rule of thumb correction on the PC algorithm. Error bars indicate a 95% confidence interval about the mean.

ables to determine the number of possible tests that could be run at each level of w. This would also allow an easy method for determining how many tests will be affected by changing the w parameter.

7 Acknowldgements

This material is based on research sponsored by the Air Force Research Laboratory and the Intelligence Advanced Research Projects Activity (IARPA), under agreement number FA8750-07-2-0158. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusion contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory, the Intelligence Advanced Research Projects Activity, or the U.S. Government.

References

[1] J. Abellan, M. Gomez-Olmedo, and S. Moral. Some variations on the PC algorithm. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models*, 2006.

[2] L. Badea. Determining the direction of causal influence in large probabilistic networks: A constraint-based approach. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 263–267, 2004.

[3] W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(2):195–210, April 1996.

[4] S. Champely. *The Pwr Package for R*. UCB Lyon 1, France, February 2007.

[5] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1-2):43–90, 2002.

[6] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Inc., 2nd edition, 1988.

[7] M. Hutter. Distribution of mutual information. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[8] K. J. Koehler. Goodness-of-fit tests for log-linear models in sparse contingency tables. *Journal of the American Statistical Association*, 81(394):483–493, June 1986.

[9] G. W. Milligan. A computer program for calculating power of the chi-square test. *Educational and Psychological Measurement*, 39(3):681–684, 1979.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kauffman, 1988.

[11] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. MIT Press, 2nd edition, 2000.

[12] H. Steck and V. Tresp. Bayesian belief networks for data mining. *Proceedings of the 2nd Workshop on Data Mining und Data Warehousing als Grundlage Moderner Entscheidungsunterstutzender Systeme*, pages 145–154, 1996.

[13] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *Proc. of the Twenty-first Conference on Uncertainty in AI*, pages 584–590, 2005.

[14] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, March 2006.

[15] S. van Dijk, L. van der Gaag, and D. Thierens. A skeleton-based approach to learning Bayesian networks from data. In *Proc. of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 2838, 2003.