Some Variations on the PC Algorithm.

Manuel Gómez-Olmedo
University of Granada
71 PUBLICATIONS **573** CITATIONS
SEE PROFILE

Serafín Moral
University of Granada
216 PUBLICATIONS **4,692** CITATIONS
SEE PROFILE

Some Variations on the PC Algorithm

J. Abellán, M. Gómez-Olmedo, and S. Moral
Department of Computer Science and Artificial Intelligence
University of Granada
18071 - Granada, Spain

Abstract

This paper proposes some possible modifications on the PC basic learning algorithm and makes some experiments to study their behaviour. The variations are: to determine minimum size cut sets between two nodes to study the deletion of a link, to make statistical decisions taking into account a Bayesian score instead of a classical Chi-square test, to study the refinement of the learned network by a greedy optimization of a Bayesian score, and to solve link ambiguities taking into account a measure of their strength. It will be shown that some of these modifications can improve PC performance, depending of the objective of the learning task: discovering the causal structure or approximating the joint probability distribution for the problem variables.

1 Introduction

There are two main approaches to learning Bayesian networks from data. One is based on scoring and searching (Cooper and Herskovits, 1992; Heckerman, 1995; Buntine, 1991). Its main idea is to define a global measure (score) which evaluates a given Bayesian network model as a function of the data. The problem is solved by searching in the space of possible Bayesian network models trying to find the network with optimal score. The other approach (constraint learning) is based on carrying out several independence tests on the database and building a Bayesian network in agreement with tests results. The main example of this approach is PC algorithm (Spirtes et al., 1993). It can be applied to any source providing information about whether a given conditional independence relation is verified.

In the past years, searching and scoring procedures have received more attention, due to some clear advantages (Heckerman et al., 1999). One is that constraint based learning makes categorical decisions from the very beginning. These decisions are based on statistical tests that may be erroneous and these errors will affect all the future algorithm bahaviour. Another one is that scoring and search procedures allow to compare very different models by a score that can be be interpreted as the probability of being the true model. As a consequence, we can also follow a Bayesian approach considering several alternative models, each one of them with its corresponding probability, and using them to determine posterior decisions (model averaging). Finally, in score and searching approaches different combinatorial optimization techniques (de Campos et al., 2002; Blanco et al., 2003) can be applied to maximize the evaluation of the learned network. On the other hand, the PC algorithm has some advantages. One of them is that it has an intuitive basis and under some ideal conditions it has guarantee of recovering a graph equivalent to the one being a true model for the data. It can be considered as an smart selection and ordering of the questions that have to be done in order to recover a causal structure.

The basic point of this paper is that PC algorithm provides a set of strategies that can be combined with other ideas to produce good learning algorithms which can be adapted to different situations. An example of this is when van Dijk et al. (2003) propose a combination of order 0 and 1 tests of PC algorithm with an scoring and searching procedure. Here, we propose several variations about the original PC algorithm. The first one will be a generalization of the necessary path condition (Steck and Tresp, 1999); the second will be to change the statistical tests for independence by considering decisions based on a Bayesian score; the third will be to allow the possibility of refining the network learned with PC by applying a greedy optimization of a Bayesian score; and finally the last proposal will be to delete edges from triangles in the graph following an order given by a Bayesian score (removing weaker edges first). We will show the intuitive basis for all of them and we will make some experiments showing their performance when learning Alarm network (Beinlich et al., 1989). The quality of the learned networks will be measured by the number or missing-added links and the Kullback-Leibler distance of the probability distribution associated to the learned network to the original one.

The paper is organized as follows: Section 2 is devoted to describe the fundamentals of PC algorithm; Section 3 introduces the four variations of PC algorithm; in Section 4 the results of the experiments are reported and discussed; Section 5 is devoted to the conclusions.

2 The PC Algorithm

Assume that we have a set of variables $\mathbf{X} = (X_1, \ldots, X_n)$ with a global probability distribution about them P. By an uppercase bold letter $\mathbf{A}$ we will represent a subset of variables of $\mathbf{X}$. By $I(\mathbf{A}, \mathbf{B}|\mathbf{C})$ we will denote that sets $\mathbf{A}$ and $\mathbf{B}$ are conditionally independent given $\mathbf{C}$.

PC algorithm assumes *faithfulness*. This means that there is a directed acyclic graph, G, such that the independence relationships among the variables in $\mathbf{X}$ are exactly those represented by G by means of the d-separation criterion (Pearl, 1988). PC algorithm is based on the existence of a procedure which is able of saying when $I(\mathbf{A}, \mathbf{B}|\mathbf{C})$ is verified in graph G. It first tries to find the skeleton (underlying undirected graph) and on a posterior step makes the orientation of the edges. Our variations will be mainly applied to the first part (determining the skeleton). So we shall describe it with some detail:

```
Start with a complete undirected graph $G'$
$i = 0$
Repeat
  For each $X \in \mathbf{X}$
    For each $Y \in ADJ_X$
      Test whether $\exists \mathbf{S} \subseteq ADJ_X - \{Y\}$
        with $|\mathbf{S}| = i$ and $I(X, Y|\mathbf{S})$
      If this set exists
        Make $S_{XY} = \mathbf{S}$
        Remove $X - Y$ link from $G'$
  $i = i + 1$
Until $|ADJ_X| \leq i, \quad \forall X$
```

In this algorithm, ADJ_X is the set of nodes adjacent to X in graph G'. The basis is that if the set of independencies is faithful to a graph, then there is not a link between X and Y, if and only if there is a subset $\mathbf{S}$ of the adjacent nodes of X such that $I(X, Y|\mathbf{S})$. For each pair of variables, S_{XY} will contain such a set, if it is found. This set will be used in the posterior orientation stage.

The orientation step will proceed by looking for sets of three variables $\{X, Y, Z\}$ such that edges $X - Z, Y - Z$ are in the graph by not the edge $X - Y$. Then, if $Z \notin S_{XY}$, it orients the edges from X to Z and from Y to Z creating a v-structure: $X \rightarrow Z \leftarrow Y$. Once, these orientations are done, then it tries to orient the rest of the edges following two basic principles: not to create cycles and not to create new v-structures. It is possible that the orientation of some of the edges has to be arbitrarily selected.

If the set of independencies is faithful to a graph and we have a perfect way of determining whether $I(X, Y|\mathbf{S})$, then the algorithm has guarantee of producing a graph equivalent (represents the same set of independencies) to the original one.

However, in practice none of these conditions is verified. Independencies are decided at the light of independence statistical tests based on a set of data $\mathcal{D}$. The usual way of doing these tests is by means of a chi-square test based on the cross entropy statistic measured in the sample (Spirtes et al., 1993). Statistical tests have errors and then, even if faithfulness hypothesis is verified, it is possible that we do not recover the original graph. The number of errors of statistical tests increases when the sample is small or the cardinality of the conditioning set $\mathbf{S}$ is large (Spirtes et al., 1993, p. 116). In both cases, due to the nature of frequentist statistical tests, there is a tendency to always decide independence (Cohen, 1988). This is one reason of doing statistical tests in increasing order of the cardinality of the sets to which we are conditioning.

Apart from no recovering the original graph, we can have another effects, as the possibility of finding cycles when orienting v-structures. In our implementation, we have always avoided cycles by reversing the arrows if necessary.

3 The Variations

3.1 Necessary Path Condition

In PC algorithm it is possible that we delete the link between X and Y by testing the independence $I(X,Y|\mathbf{S})$, when $\mathbf{S}$ is a set containing nodes that do not appear in a path (without cycles) from X to Y. The inclusion of these nodes is not theoretically wrong, but statistical tests make more errors when the size of the conditioning set increases, then it can be a source of problems in practice. For this reason, Steck and Tresp (1999) proposed to reduce $ADJ_X - \{Y\}$ in Step 6, by removing all the nodes that are not in a path from X to Y. In this paper, we will go an step further by considering any subset $CUT_{X,Y}$ disconnecting X and Y in the graph in which the link $X - Y$ has been deleted, playing the role of $ADJ_X - \{Y\}$. Consider that in the skeleton, we want to see whether link $X - Y$ can be deleted, then we first remove it, and if the situation is the one in Figure 1, we could consider $CUT_{X,Y} = \{Z\}$. However, in the actual algorithm (even with the necessary path condition) we consider the set $ADJ_X - \{Y\}$, which is larger, and therefore with an increased possibility of error.

Our proposal is to apply PC algorithm, but by considering in step 6 a cut set of minimum size in the graph without $X - Y$ link, as Acid and de Campos (2001) did in a different context. The computation of this set will need some extra time, but it can be done in polynomial time with a modification of Ford-Fulkerson algorithm (Acid and de Campos, 1996).

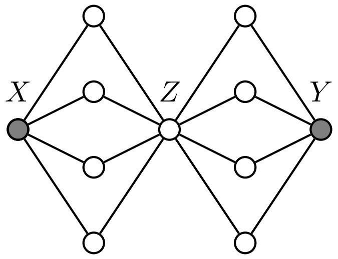

Figure 1: An small cut set

3.2 Bayesian Statistical Tests

PC algorithm performs a chi-square statistical test to decide about independence. However, as shown by Moral (2004), sometimes statistical tests make too many errors. They try to keep the Type I error (deciding dependence when there is independence) constant to the significance level. However, if the sample is large enough this error can be much lower by using a different decision procedure, without an important increase in Type II error (deciding independence when there is dependence). Margaritis (2003) has proposed to make statistical tests of independence for continuous variables by using a Bayesian score after discretizing them. Previously, Cooper (1997) proposed a different independence test based on a Bayesian score, but only when conditioning to 0 or 1 variable. Here we propose to do all the statistical tests by using a Bayesian Dirichlet score[1] (Heckerman, 1995) with a global sample size s equal to 1.0. The test $I(X,Y|\mathbf{S})$ is carried out by comparing the scores of X with $\mathbf{S}$ as parents and of X with $\mathbf{S} \cup \{Y\}$ as parents. If the former is larger than the later, the variables are considered independent, and in the other case, they

[1] We have chosen this score instead of the original K2 score (Cooper and Herskovits, 1992) because this is considered more correct from a theoretical point of view (Heckerman et al., 1995).

are considered dependent. The score of X with a set of parents $Pa(X) = \mathbf{Z}$ is the logarithm of:

$$\prod_{\mathbf{z}} \left(\frac{\Gamma(s')}{\Gamma(N_{\mathbf{z}} + s')} \prod_{x} \frac{\Gamma(N_{\mathbf{z},x} + s'')}{\Gamma(s'')} \right)$$

where $N_{\mathbf{z}}$ is the number of occurrences of $[\mathbf{Z} = \mathbf{z}]$ in the sample, $N_{\mathbf{z},x}$ is the number of occurrences of $[\mathbf{Z} = \mathbf{z}, X = x]$ in the sample, s' is s divided by the number of possible values of $\mathbf{Z}$, and s'' is equal to s' divided by the number of values of X.

3.3 Refinement

If the statistical tests do not make errors and the faithfulness hypothesis is verified, then PC algorithm will recover a graph equivalent to the original one, but this can never be assured with finite samples. Also, even if we recover the original graph, when our objective is to approximate the joint distribution for all the variables, then depending of the sample size, it can be more convenient to use a simpler graph than the true one. Imagine that the variables follow the graph of Figure 2. This graph can be recovered by PC algorithm by doing only statistical independence tests of order 0 and 1 (conditioning to none or 1 variable). However, when we are going to estimate the parameters of the network we have to estimate a high number of probability values. This can be a too complex model (too many parameters) if the database is not large enough. In this situation, it can be reasonable to try to refine this network, taking into account the actual orientation and the size of the model. In this sense, the result of PC algorithm can be used as an starting point for a greedy search algorithm to optimize a concrete metric.

In particular, our proposal is based on the following steps:

1. Obtain an order compatible with the graph learned by PC algorithm.

2. For each node, try to delete each one of its parents or to add some of the non parents preceding nodes as parent, measuring the resulting Bayesian Dirichlet score. We make the movement with highest score difference while this is positive.

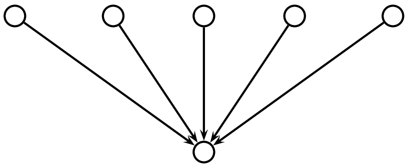

Figure 2: A too complex network.

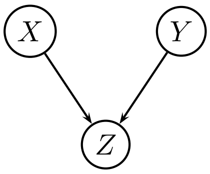

Figure 3: A simple network

Refinement can also solve some of the problems associated with the non verification of the faithfulness hypothesis. Assume for example, that we have a problem with 3 variables, X, Y, Z, and that the set of independencies is given by the independencies of the graph in Figure 3 plus the independence $I(Y, Z|\emptyset)$. PC algorithm will estimate a network, where the link between Y and Z is lost. Even if the sample is large we will estimate a too simple network which is not an I-map (Pearl, 1988). If we orient the link $X \rightarrow Z$ in PC algorithm, refinement can produce the network in Figure 3, by checking that the Bayesian score is increased (as it should be the case if $I(Z, Y|X)$ is not verified).

The idea of refining a learned Bayesian network by means of a greedy optimization of a Bayesian score has been used in a different context by Dash and Druzdzel (1999).

3.4 Triangles Resolution

Imagine that we have 3 variables, X, Y, Z, and that no independence relationship involving them is verified: each pair of variables is dependent and conditionally dependent giving the third one. As there is a tendency to decide for independence when the size of the conditioning set is larger, then it is possible that all order 0 tests produce dependence, but when we test the independence of two variables with respect to a third one, we obtain independence. In this situation, the result of PC algorithm, will depend of the order in which tests are carried out. For example, if we ask first for the independence, $I(X,Y|Z)$, then the link $X-Y$ is deleted, but not the other two links, which will be oriented in a posterior step without creating a v-structure. If we test first $I(X,Z|Y)$, then the deleted link will be $X-Z$, but not the other two.

It seems reasonable that if one of the links is going to be removed, we should choose the weakest one. In this paper, for each 3 variables that are a triangle (the graph contains the 3 links) after order 0 tests, we measure the strength of link $X-Y$ as the Bayesian score of X with Y,Z as parent, minus the Bayesian score of X with Z as parents. For each triangle we delete the link with lowest strength (if this value is lower than 0). This is done as an intermediate step, between order 0 and order 1 conditional independence tests.

In this paper, it has been implemented only in the case in which independence tests are based on a Bayesian score, but it could be also considered in the case of Chi-square tests by considering the strength of a link equal to the p-value of the statistical independence test.

A deeper study of this type of interdependencies between the deletion of links (the presence of a link depends of the absence of other one, and vice versa) has been carried out by Steck and Tresp (1999), but the resolution of these ambiguities is not done. Hugin system (Madsen et al., 2003) allows to decide between the different possibilities by asking to the user. Our procedure could be extended to this more general setting, but at this stage the implementation has been limited to triangles, as it is, at the sample time, the most usual and simplest situation.

4 Experiments

We have done some experiments with the Alarm network (Beinlich et al., 1989) for testing the PC variations. In all of them, we have started with the original network and we have generated samples of different sizes by logic sampling. Then, we have tried to recover the original network from the samples by using the different variations of the PC algorithm including the orientation step. We have considered the following measures of error in this process: number of missing links, number of added links, and the Kullback-Leibler distance (Kullback, 1968) of the learned probability distribution to the original one[2]. Kullback-Leibler distance is a more appropriate measure of error when the objective is to approximate the joint probability distribution for all the variables and the measures of number of differences in links is more appropriate when our objective is to recover the causal structure of the problem. We do not consider the number of wrong orientations as our variations are mainly focused in the skeleton discovery phase of PC algorithm. The number of added or deleted links only depend of the first part of the learning algorithm (selection of the skeleton).

Experiments have been carried out in Elvira environment (Consortium, 2002), where a local computation of Kullback-Leibler distance is implemented. The different sample sizes we have used are: 100, 500, 1000, 5000, 10000, and for each sample size, we have repeated the experiment 100 times. The combinations of algorithms we have tested are the following:

Alg1 This is the algorithm with minimal separating sets, score based tests, no refinement, and triangle resolution.

Alg2 Algorithm with minimal separating sets, score based tests, refinement, and triangle resolution.

Alg3 Algorithm with adjacent nodes as separating sets, score based tests, no refinement, and triangle resolution.

Alg4 Algorithm with minimal separating sets, Chi-square tests, no refinement and no resolution of triangles.

[2] The parameters are estimated with a Bayesian Dirichlet approach with a global sample size of 2.

	100	500	1000	5000	10000
Alg1	17.94	8.76	6.02	3.25	2.56
Alg2	16.29	8.16	5.59	3.7	3.28
Alg3	26.54	18.47	14.87	8.18	7.08
Alg4	29.07	11.49	8.42	3.53	2.14
Alg5	17.87	8.83	6.08	3.03	2.57

Table 1: Average number of missing links

	100	500	1000	5000	10000
Alg1	10.42	4.96	2.87	2.2	1.98
Alg2	26.13	17.52	16.32	16.01	15.38
Alg3	3.06	0.63	0.14	0.03	0.01
Alg4	14.73	5.96	5.68	4.78	4.79
Alg5	10.46	4.99	3.21	1.92	1.9

Table 2: Average number of added links

Alg5 This is the algorithm with minimal separating sets, score based tests, no refinement, and no resolution of triangles.

These combinations are designed in this way, as we consider Alg1 our basic algorithm to recover the graph structure, and then we want to study the effect of the application of each one of the variations to it.

Table 1 contains the average number of missing links, Table 2 the average number of added links, Table 3 the average Kulback-Leibler distance, and finally Table 4 contains the average running times of the different algorithms. In these results we highlight the following facts:

Refinement (Alg2) increases the number of errors in the recovering of the causal structure (mainly more added links), but decreases the Kullback-Leibler distance to the original distribution. So, its application will depend of our objective: approximate the joint distribution or recover the causal structure. Refinement is fast and it does not add a significant amount of extra time.

	100	500	1000	5000	10000
Alg1	4.15	2.46	1.81	0.98	0.99
Alg2	2.91	0.96	0.56	0.19	0.11
Alg3	4.98	3.27	2.58	1.11	0.77
Alg4	5.96	2.19	1.49	1.05	0.91
Alg5	4.15	2.36	1.86	1.11	0.98

Table 3: Average Kullback-Leibler distance

	100	500	1000	5000	10000
Alg1	2.16	4.1	5.88	21.98	42
Alg2	2.25	4.12	5.89	21.98	42.1
Alg3	0.33	1.56	3.34	20.73	44.14
Alg4	2.45	8.9	13.89	39.42	68.85
Alg5	2.21	4.15	6.02	22.95	44.4

Table 4: Average time

When comparing Alg1 with Alg3 (minimum size cut set vs set of adjacent nodes) we observe that with adjacent nodes fewer links are added and more ones are missing. The total amount of errors is in favour of Alg1 (minimum size cut set). This is due to the fact that Alg3 makes more conditional independence tests and with larger conditioning sets of variables, which makes more possible to delete links. Kullback-Leibler distance is better for Alg1 except for the largest sample size. A possible explanation, is that with this large sample size, we really do not miss any important link of the network, and added links can be more dangerous than deleted ones (when we delete links we are averaging distributions). With smaller sample sizes, Alg3 had a worse Kullback-Leibler as it can be missing some important links. We do not have any possible explanation to the fact that Alg1 does not improve Kullback-Leibler distance when increasing the sample size from 5000 to 10000. When comparing the time of both algorithms, we see that Alg1 needs more time (to compute minimum size cut sets) however, when the sample size is large this extra time is compensated by the lower number of statistical tests, being the total time for size 10000 lower in the case of Alg1 (with minimum size cut sets).

When comparing Alg1 and Alg4 (Score test and triangle resolution vs Chi-square tests and no triangle resolution) we observe than Alg4 always add more links and miss more links (except for the largest sample size). The total number of errors is lower for Alg1. It is meaningful the fact that the number of added links do not decrease when going from a sample of 5000 to a sample of 10000. This is due to the fact that the probability of considering dependence when there is independence is fixed (the significance level) for large the sample sizes. So all the extra information of the larger sample is devoted to decrease the number of missing links (2.14 in Alg4 against 2.56 in Alg1), but the difference in added links is 4.79 in Alg4 against 1.98 in Alg1. So the small decreasing in missing links is at the cost of a more important error in the number of added links. Bayesian scores tests are more balanced in the two types of errors. When considering the Kullback-Leibler distance, we observe again the same situation than when comparing Alg1 and Alg2: a greater number of errors in the structure does not always imply a greater Kullback-Leibler distance. The time is always greater for Alg4.

The differences between Alg1 and Alg4 are not due to the triangles resolution in Alg1. As we will see now, triangles resolution do not really implies important changes in Alg1 performance. In fact, the effect of Chi-square tests against Bayesian tests without any other additional factor, can be seen by comparing Alg5 and Alg4. In this case, we can observe the same differences as when comparing Alg1 and Alg5.

When comparing Alg1 and Alg5 (no resolution of triangles) we see that there is not important differences in performance (errors and time) when resolution of triangles is applied. It seems that the total number of errors is decreased for intermediate sample sizes (500-1000) and there are not important differences for the other sample sizes, but more experiments are necessary. Triangles resolution do not really add a meaningful extra time. Applying this step needs some time, but the graph is simplified and posterior steps can be faster.

5 Conclusions

In this paper we have proposed four variations of the PC algorithm and we have tested them when learning the Alarm network. Our final recommendation would be to use the PC algorithm with score based tests, minimum size cut sets, and triangle resolution. The application of refinement step would depend of the final aim: if we want to learn the causal structure, then refinement should not be applied, but if we want to approximate the joint probability distribution, then refinement should be applied. We recognize that more extensive experiments are necessary to evaluate the application of these modifications, specially the triangle resolution. But we feel that this modification is intuitively supported, and that it could have a more important role in other situations, specially if the faithfulness hypothesis is not verified.

Other combinations could be appropriated if the objective is different, for example if we want to minimize the number of added links, then Alg3 (with adjacent nodes as cut set) could be considered.

In the future we plan to make more extensive experiments testing different networks and different combinations of these modifications. At the same time, we will consider another possible variations, as for example an algorithm mixing the skeleton and orientation steps. It is possible that some of the independencies are tested conditional to some sets, that after the orientation do not separate the two links. We also plan to study alternative scores and to study the efect of using different sample sizes. Also partial orientations can help to make the separating sets even smaller as there can be some paths which are not active without observations. This can make algorithms faster and more accurate. Finally, we think that the use of PC and its variations as starting points for greedy searching algorithms needs further research effort.

Acknowledgments

This work has been supported by the Spanish Ministry of Science and Technology under the Algra project (TIN2004-06204-C03-02).

References

S. Acid and L.M. de Campos. 1996. Finding minimum d-separating sets in belief networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 3–10, Portland, Oregon.

S. Acid and L.M. de Campos. 2001. A hybrid methodology for learning belief networks: Benedit. *International Journal of Approximate Reasoning*, 27:235–262.

I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. 1989. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256. Springer-Verlag.

R. Blanco, I. Inza, and P. Larraaga. 2003. Learning bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems*, 18:205–220.

W. Buntine. 1991. Theory refinement in bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann, San Francisco, CA.

J. Cohen. 1988. *Statistical power analysis for the behavioral sciences (2nd edition).* Erlbaum, Hillsdale, NJ.

Elvira Consortium. 2002. Elvira: An environment for probabilistic graphical models. In J.A. Gmez and A. Salmern, editors, *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, pages 222–230.

G.F. Cooper and E.A. Herskovits. 1992. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

G.F. Cooper. 1997. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1:203–224.

D. Dash and M.J. Druzdzel. 1999. A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 142–149. Morgan Kaufmann.

L.M. de Campos, J.M. Fernndez-Luna, J.A. Gmez, and J. M. Puerta. 2002. Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning*, 31:511–549.

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

D. Heckerman, C. Meek, and G. Cooper. 1999. A bayesian approach to causal discovery. In C. Glymour and G.F. Cooper, editors, *Computation, Causation, and Discovery*, pages 141–165. AAAI Press.

D. Heckerman. 1995. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research.

S. Kullback. 1968. *Information Theory and Statistics.* Dover, New York.

A.L. Madsen, M. Land, U.F. Kjærulff, and F. Jensen. 2003. The hugin tool for learning networks. In T.D. Nielsen and N.L. Zhang, editors, *Proceedings of ECSQARU 2003*, pages 594–605. Springer-Verlag.

D. Margaritis. 2003. *Learning Bayesian Model Structure from Data.* Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

S. Moral. 2004. An empirical comparison of score measures for independence. In *Proceedings of the Tenth International Conference IPMU 2004, Vol. 2*, pages 1307–1314.

J. Pearl. 1988. *Probabilistic Reasoning with Intelligent Systems.* Morgan & Kaufman, San Mateo.

P. Spirtes, C. Glymour, and R. Scheines. 1993. *Causation, Prediction and Search.* Springer Verlag, Berlin.

H. Steck and V. Tresp. 1999. Bayesian belief networks for data mining. In *Proceedings of the 2nd Workshop on Data Mining und Data Warehousing als Grundlage moderner entscheidungsunterstuetzender Systeme*, pages 145–154.

S. van Dijk, L.C. can der Gaag, and D. Thierens. 2003. A skeleton-based approach to learning bayesian networks from data. In Nada Lavrač, Dragan Gamberger, Ljupčo Todorovski, and Hendrik Blockeel, editors, *Proceedings of the Seventh Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 132–143. Springer Verlag.