ROMA TRE
UNIVERSITÀ DEGLI STUDI

COLLANA DEL
DIPARTIMENTO DI ECONOMIA

LEARNING A BAYESIAN NETWORK FROM ORDINAL DATA

Flaminia Musella

Working Paper n° 139, 2011

I Working Papers del Dipartimento di Economia svolgono la funzione di divulgare tempestivamente, in forma definitiva o provvisoria, i risultati di ricerche scientifiche originali. La loro pubblicazione è soggetta all'approvazione del Comitato Scientifico.

Per ciascuna pubblicazione vengono soddisfatti gli obblighi previsti dall'art. l del D.L.L. 31.8.1945, n. 660 e successive modifiche.

Copie della presente pubblicazione possono essere richieste alla Redazione.

REDAZIONE:
Dipartimento di Economia
Università degli Studi Roma Tre
Via Silvio D'Amico, 77 - 00145 Roma
Tel. 0039-06-57335655 fax 0039-06-57335771
E-mail: dip_eco@uniroma3.it

ROMA TRE
UNIVERSITÀ DEGLI STUDI

DIPARTIMENTO DI ECONOMIA

LEARNING A BAYESIAN NETWORK FROM ORDINAL DATA

Flaminia Musella

Learning a Bayesian network from ordinal data

Flaminia Musella

Department of Economics, Roma Tre University

Abstract

Bayesian networks are graphical models that represent the joint distribution of a set of variables using directed acyclic graphs. When the dependence structure is unknown (or partially known) the network can be learnt from data. In this paper, we propose a constraint-based method to perform Bayesian networks structural learning in presence of ordinal variables. The new procedure, called OPC, represents a variation of the PC algorithm. A nonparametric test, appropriate for ordinal variables, has been used. It will be shown that, in some situation, the OPC algorithm is a solution more efficient than the PC algorithm.

Keyword: Structural Learning, Monotone Association, Nonparametric Methods.

EconLit codes: C140; C510

1 Introduction

Ordinal variables are becoming very common in observational studies of several research areas such as social science, biostatistics, education and marketing. Their increasing use in surveys has influenced the development of appropriate methods for ordinal variables (Agresti 2010). In the literature, Clogg and Shihadeh (1994) and Agresti (2002) argue that, due to the ordinal variables informative strength, results gained with ordinal methods may be quite different from those achieved using nominal ones. However, some nominal techniques are commonly used with ordinal data. This entails that the ordering among categories is not considered and a loss of (sometimes relevant) information is produced. For this reason an increasing interest in preserving the ordering of ordinal data in Bayesian networks structural learning has been developed. In the literature, there are a few ordinal-sensitive procedures for learning Bayesian networks structure from ordinal data. In this paper we consider the PC algorithm (Spirtes *et al.* 2000) and we propose a variation that takes into account information provided by ordinal variables. The paper is organized as follows. Section 2 provides a background on Bayesian networks; Section 3 deals with the PC algorithm; Section 4 introduces the OPC algorithm whose performance will be discussed according to empirical evaluations presented in Section 5. Finally, Section 6 addresses some conclusions and further developments.

2 Basics on Bayesian networks

Bayesian networks (BNs, Cowell *et al.* (1999)) are multivariate statistical models that represent the multivariate probability distribution $\mathcal{P}$ of a set of variables $\boldsymbol{X}$ by means of directed acyclic graphs (DAGs). A directed graph is a pair of sets usually denoted by $\mathcal{G}^D = (\mathcal{V};\mathcal{E}^D)$: $\mathcal{V}$ is a finite set of *vertices*, also called *nodes*, representing random variables in $\boldsymbol{X}$, and $\mathcal{E}^D$ is a set of directed *edges*. A directed graph is said to be acyclic if it does not contain directed cycles. In a BN, each node of the DAG is associated with a (conditional) probability table so that a BN is defined by the DAG that encodes the independence relations between variables and by the parameters i.e. the set of probabilities tables. Independence relations in the joint distribution can be read off the DAG by using the d-separation criteria (Pearl 1986). Different graphical configurations can encode the same set of independence relations. For instance, consider configurations in Figure 1.

$$i \rightarrow \gamma \rightarrow j \qquad\qquad i \leftarrow \gamma \leftarrow j \qquad\qquad i \leftarrow \gamma \rightarrow j \qquad\qquad i \rightarrow \gamma \leftarrow j$$

(a) *(b)* *(c)* *(d)*

Figure 1: Different structures for the triplet of nodes i, j and γ

Different structures involving a pair of nodes, i and j, directly connected with the node γ are displayed. Node γ plays the role of *transition* node in the serial configurations - *(a)* and *(b)* -, of *common source* node in the diverging structure - *(c)* - and of *common sink* in the converging structure - *(d)*. A common sink is also called *collider* or, following the notation of Cox and Wermuth (1996), *v-structure*. Serial and diverging configurations encode both the relations:

1. i and j are not independent;
2. i and j are independent given γ.

As a consequence of this, (a), (b) and (c) encode the same d-separations that, following th notation due to Dawid (1979) can be written as:

1. $i \not\perp\!\!\!\perp j$;
2. $i \perp\!\!\!\perp j|\gamma$.

On the contrary, (d) shows a converging connection that entails a different relation, that is i and j are not independent given γ, $i \not\perp\!\!\!\perp j|\gamma$. DAGs with the same d-separation properties are said *Markov equivalent* (Verma and Pearl 1990). The Markov equivalence permits the partition of DAGs space into classes of models, namely *equivalence classes*: Bayesian networks belonging to the same equivalence class are statistically indistinguishable since they represent equivalent parameterizations of the same distribution(Chickering 1995). The canonical pictorial representation of an equivalence class is given by an hybrid graph called *partial* DAG (PDAG). Given a DAG, a PDAG can be obtained

- by considering the skeleton of the DAG, that is the basic structure of the graph without directions;

- by keeping directed edges for those arrows involved in v-structure;
- by ignoring other directions.

For instance, the PDAGs of the DAGs in Figure 1 are displayed in Figure 2.

$i - \gamma - j$	$i - \gamma - j$	$i - \gamma - j$	$i \rightarrow \gamma \leftarrow j$
(a)	*(b)*	*(c)*	*(d)*

Figure 2: PDAGs of DAGs in Fugure 1

Using PDAG instead of DAG can be more profitable in structural learning (e.g.Chickering (2002)). The *structural learning* phase consits in estimating the DAG structure of a BN directly from data. Building a BN, in fact, requires to specify both the DAG and the parameters, but in many situations, the subject matter knowledge is absent or not sufficient to manually build the BN. In these cases, the DAG structure has to be inferred from data using appropriate learning methods. Structural learning has been extensively discussed in the literature (Buntine 1994; Buntine 1996; Neapolitan 2003) and it mainly can be supported through two approaches: *scoring and searching* (Cooper and Herskovits 1992; Heckerman 1995) and *constraint-based* (Pearl 1988; Spirtes *et al.* 2000). The first is based on two steps: given a chosen (Bayesian or not Bayesian) metric a *score* is assigned to all possible models in a given space; then algorithms *search* and select the model that maximises the score. Even though score-and-search methods are computationally expensive, their main advantage is that they compare several different models (Heckerman *et al.* 1997).

The second approach is based on dependence analysis. The general procedure consists in carrying out a sequence of independence statistical tests and in drawing the network according to the test results. Independence tests are iteratively performed to make, step by step, unchangeable decisions about edges presence in the graph. Constraint-based algorithms are intuitive and relatively fast but they can be unstable. Possible mistakes in test, in fact, determine erroneous decisions that can affect the future algorithm behaviour causing the selection of a suboptimal result (Dash and Druzdzel 1999).

This paper follows the last approach. We start by introducing the most used and well-known constraint-based algorithms, the PC algorithm, that is the starting point to develop the new procedure.

3 The PC algorithm

The *PC algorithm* (Spirtes *et al.* 2000) is a stepwise backward algorithm that takes as input a database $\mathcal{D}$ over a set of K variables and it provides, in output, a PDAG. The consistency of the PC algorithm has been proved under the assumptions that (1) the DAG and the joint probability distribution are faithful to each other and $\mathcal{G}^D$ is a perfect map (P-map) of $\mathcal{P}$; (2) data are infinite; (3) statistical tests have no errors.

The structure of the PC algorithm consists in three main steps:

1. find the skeleton of the graph;

2. find the head-to-head configurations;

3. orient the rest of the links without producing any cycle and any other head-to-head configuration.

1. Let $\boldsymbol{X}$ be a set of K random variables. Let $\mathcal{V}$ be a set of K nodes in a graph so that each node in $\mathcal{V}$ represents a random variable of $\boldsymbol{X}$. The PC algorithm starts from a complete undirected graph G', i.e. a graph where all nodes are connected to each other. Given a chosen significance level $0 < \alpha < 1$ and a specific ordering $Order(\mathcal{V})$ over $\mathcal{V}$, the PC algorithm performs statistical tests to decide if to remove or maintain edges between nodes in the graph. The procedure is shown in the following pseudo-code where $ne(i)$ denotes the set of nodes adjacent to i-th node, *i.e.* the set of vertices linked to i by an edge.

   ```
   Start with a complete undirected graph $G'$
   $\ell = 0$
   repeat
     for each $i \in \mathcal{V}$ do
       for each $j \in ne(i)$ do
         Test whether $\exists S \in ne(i) \setminus \{j\}$ with $|S| = \ell$ and $X_i \perp\!\!\!\perp X_j | S$
         if this set exists then
           Make $S_{ij} = S$
           Remove link between nodes $i$ and $j$ in $G'$
         end if
       end for
     end for
     $\ell = \ell + 1$
   until $|ne(i)| \leq \ell \; \forall i$
   ```

   In detail, the PC algorithm checks marginal and conditional relations between adjacent nodes, i and j, given a conditioning set S of increasing cardinality, $0 \leq \ell < |ne(i)|$. Computationally, the conditional cross entropy between corresponding variables, $CE(X_i, X_j|S)$, is calculated. The algorithm uses the test statistic G^2 which is equal to $2nCE(X_i, X_j|S)$ where n is the sample size. Under the null hypothesis of independence, G^2 follows a χ^2 distribution (Lindgren 1976) with degrees of freedom (df) equal to $(k_{x_i} - 1)(k_{x_j} - 1) \prod_{x_\gamma \in S} k_{x_\gamma}$ where k_{x_i}, k_{x_j}, k_{x_γ} respectively denote the number of values of variables X_i, X_j and $X_\gamma \in S$. Given a significance level α: if $G^2(X_i, X_j, \varnothing) < \chi^2_{(1-\alpha, df)}$ then X_i and X_j are marginally independent ($X_i \perp\!\!\!\perp X_j$); if $G^2(X_i, X_j, S) < \chi^2_{(1-\alpha, df)}$ then X_i and X_j are conditionally independent given S ($X_i \perp\!\!\!\perp Y_j|S$).
   The output of this first step is the underlying undirected graph, also called skeleton of the graph.

2. The second step of the algorithm consists in finding head-to-head configurations. When two generic variables X_i and X_j are not conditionally

independent given a subset $S_{ij} = X_\gamma$ then γ is a collider node and a v-structure $i \rightarrow \gamma \leftarrow j$ is drawn; if X_i and X_j are conditionally independent given a subset $S_{ij} = X_\gamma$ then γ is not a collider node and, at the second stage, edges remain undirected $i - \gamma - j$.

3. In the last step of the algorithm some constraints must be fulfilled: no new head-to-head configurations can be created; cycles in the graph are forbidden. At the end of this step some edges can remain undirected.

Many limitations of the PC algorithm have been discussed in the literature and many variations have been proposed to overcome them (Steck 2001; Fernandes *et al.* 2004; Abellán *et al.* 2006). We focus our attention on the PC algorithm behaviour in presence of ordinal variables and we introduce our proposal in the following Section.

4 The OPC algorithm

In order to check independences, the PC algorithm uses the G^2 test that treats categorical variables as nominal even if they are ordinal. This may produce an information loss. Our proposal consists in a new procedure that uses, in place of G^2, a more appropriate test for ordinal variables. The main advantage of the variation consists in considering additional information provided by ordinal variables by using a well known methodological scheme; the new procedure has been called Ordinal PC algorithm, *OPC algorithm*.

The OPC algorithm structure is the same of the PC algorithm, however it uses a different test for checking conditional independences. When variables are ordinal, independence is generally tested by rank-based nonparametric tests (Siegel and Castellan 1988). The OPC algorithm uses the Jonckheere-Terpstra test for checking conditional independences. The Jonckheere Terpstra test was proposed by Terpstra (1952) and Jonckheere (1954) as a nonparametric test for trend among ordered alternatives. The test, already implemented in `MIM` (Edwards 1995), is appropriate for contingency tables where both variables are ordinal. Here, the Jonckheere-Terpstra test is used for comparing a row ordered variable with a column ordered variable.

Let $\mathcal{D}$ be a set of data made of n observations on three variables X_1, X_2 and X_3. Suppose we are interested in checking $X_1 \perp\!\!\!\perp X_3|X_2$ where X_1 and X_3 are both ordinal with T and C levels respectively and X_2 is a discrete variables with L levels. For the k-th level of X_2, data are organized as in Table 1.

Let $F_{i,k}(x_3)$ be the distribution of X_3 given $X_1 = i$ and $X_2 = k$. The null hypothesis of Jonckheere-Terpstra test is that of homogeneity, that is:

$$H_0 : F_{1,k}(x_3) = F_{2,k}(x_3) = ... = F_{T,k}(x_3), \forall x_3, \forall k$$

This is tested against the alternative hypothesis of a stochastic ordering among distributions.

$$F_{i,k}(x_3) > F_{j,k}(x_3), \text{ with } i < j, \forall x_3, \forall k$$

	X_3				
X_1	1	2	....	C	Total
1	n_{11k}	n_{12k}	....	n_{1Ck}	n_{1+k}
2	n_{21k}	n_{22k}	....	n_{2Ck}	n_{2+k}
....	....	....	....	....	....
T	n_{T1k}	n_{T2k}	....	n_{TCk}	n_{T+k}
Total	n_{+1k}	n_{+2k}	....	n_{+Ck}	n_{++k}

Table 1: The k-th slice of the $T \times C \times L$ table

or

$$F_{i,k}(x_3) < F_{j,k}(x_3), \text{ with } i > j, \forall x_3, \forall k$$

The test statistic is:

$$JT = \sum_{k=1}^{L} \sum_{i=2}^{T} \sum_{j=1}^{i-1} \left\{ \sum_{s=1}^{C} w_{ijsk} n_{isk} - \frac{n_{i+k}(n_{i+k}+1)}{2} \right\}.$$

The test statistic is based on w_{ijsk} that are the Wilcoxon scores. They are denoted by:

$$w_{ijsk} = \sum_{t=1}^{s-1}(n_{itk} + n_{jtk}) + \frac{(n_{isk} + n_{jsk} + 1)}{2}$$

Under the null hypothesis the mean of JT is the following:

$$E(JT|H_0) = \frac{\sum_{k=1}^{L}(n_{++k}^2 - \sum_{i=1}^{T} n_{i+k}^2)}{4}$$

The asymptotic variance, discussed by Lehmann (1975) and Pirie (1983), is:

$$V\hat{a}r(JT|H_0) = \frac{V_1}{72} + \frac{V_2}{36(n_{++k}(n_{++k}-1)(n_{++k}-2))} + \frac{V_3}{8(n_{++k}(n_{++k}-1))}$$

where:

$$\begin{aligned} V_1 &= n_{++k}(n_{++k}-1)(2n_{++k}+5) - \sum_{i=1}^{T} n_{i+k}(n_{i+k}-1)(2n_{i+k}+5) - \\ &\quad - \sum_{j=1}^{C} n_{+jk}(n_{+jk}-1)(2n_{+jk}+5) \\ V_2 &= \sum_{i=1}^{T} n_{i+k}(n_{i+k}-1)(n_{i+k}-2) - \sum_{j=1}^{C} n_{+jk}(n_{+jk}-1)(n_{+jk}-2) \\ V_3 &= \sum_{i=1}^{T} n_{i+k}(n_{i+k}-1) - \sum_{j=1}^{C} n_{+jk}(n_{+jk}-1) \end{aligned}$$

Asymptotically, the test statistic is a standard normal and the two-sided p-value is given by:

$$p = Pr(|JT - E(JT|H_0)| \geq |JT_{obs} - E(JT|H_0)||H_0)$$

5 Empirical evaluations

Here the performance of PC and OPC algorithms are compared. The empirical evaluations have been conducted using two different sets of data: *Customer Satisfaction data* coming from a real survey of customer satisfaction and *Political Action data* borrowed from the literature (Barnes and Kaase 1979). Both datasets are made of six ordinal variables measured on three levels (Customer satisfaction data) and four levels (Political Action data) respectively. For each dataset, a Bayesian network has been learnt using the PC algorithm; the learnt network has been assumed as true and has been used as gold standard. According to every network, multiple datasets have been generated for different sample sizes: 50, 100 and 500. In detail, we generated 1000 training sets for each sample size. For every simulated dataset two Bayesian networks have been estimated: one using the PC algorithm and one using the OPC algorithm, given a level of significance equal to 0.05. We compared results of structural learning with the gold standard in term of (1) skeleton identification ability and (2) structural accuracy.

The skeleton identification ability has been measured by three indexes that compare the skeleton of the true DAG, here denoted by G^*, with the skeleton of the estimated DAG, H^*. The measures are: *true positive rate* (TPR) that is given by the number of edges correctly found in H^* over the number of true edges in G^*; *false positive rate* (FPR) that is the number of edges incorrectly found in H^* over the number of true gaps (absent edges) in G^*; *true discovery rate* (TDR) that is the proportion of edges correctly found on the total number of found edges (both in the estimated graph). These measures return a set of information interpretable in term of sensitivity (that is greater for TPR values closer to 1), specificity (that is greater for FPR values closer to 0) and precision (that is greater for TDR values closer to 1).

The structural accuracy has been evaluated using the *structural Hamming distance* (SHD, Tsamardinos *et al.* (2006)) that is a performance measure computing the structural distance between PDAGs. SHD is an overall metric that directly compares the learnt PDAG and the original PDAG by counting the number of operations required to convert the fitted graph into the true graph. Admitted operations are addition or deletion of edges and addition, deletion or reversal of directions.

In order to evaluate algorithms performance, a mean value of each performance indicator has been computed with respect to different sample sizes. The results are shown for each experimental evaluation separately.

The gold standard network for Customer Satisfaction data is the DAG in Figure 3.

	TPR			**FPR**			**TDR**		
Algorithm	*Sample size*			*Sample size*			*Sample size*		
	50	*100*	*500*	*50*	*100*	*500*	*50*	*100*	*500*
PC	0.42	0.64	0.97	0.11	0.04	0.02	0.73	0.91	0.98
OPC	0.69	0.87	0.97	0.04	0.03	0.02	0.93	0.96	0.99

Table 2: Mean values of performance for PC and OPC algorithm performed on 1000 datasets generated according to the DAG in Figure 3

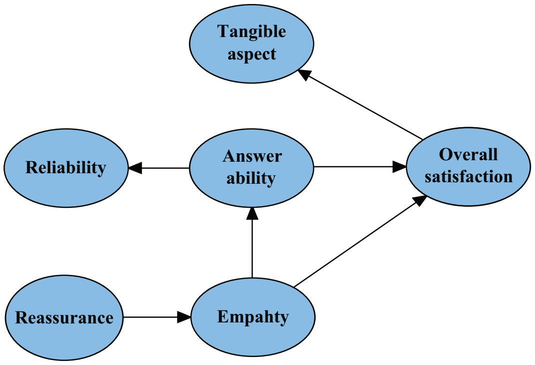

Figure 3: The true DAG for customer satisfaction data

Results shown in Table 2 suggest that the skeleton learning ability of both algorithms increases with larger sample sizes. In detail, since more observations makes statistical test more sensitive, when the sample size is equal to 500, the TPR is really close to one for both algorithms. Furthermore, for the largest sample size, algorithms have the same performance results considering every indexes. Despite that, when the sample size is 50 or 100 the TPR and the TDR of OPC algorithm are larger than those of PC algorithm and the FPR of OPC algorithm is smaller than that of PC algorithm. The gap of performance is relevant for the smallest sample size. On the basis of this results, the OPC algorithm seems to be more sensitive, more accurate and more reliable than PC algorithm above all for small samples.
Results of the SHD are shown in Table 3.

	Sample size		
Algorithm	**50**	**100**	**500**
PC	6.07	4.92	3.3
OPC	4.89	4.17	3.2

Table 3: The SHD average for PC and OPC algorithm performed on 1000 datasets generated according to the DAG in Figure 3

The SHD of both algorithms decreases when the sample size increases. However, given the same sample size, the SHD of PC algorithm is larger than the SHD of OPC algorithm. Since small values of SHD mean that less operations are required to make the estimated PDAG and the true PDAG match, the OPC algorithm outperforms the PC algorithm.
These results are confirmed also by the second experiment. The gold standard network for Political Action data is displayed in Figure 4.

Results in Table 4 highlight that algorithms performance increase with the increasing of sample size. In detail, OPC algorithm performance are more satisfying than PC algorithm performance when the sample size is 50 or 100. When

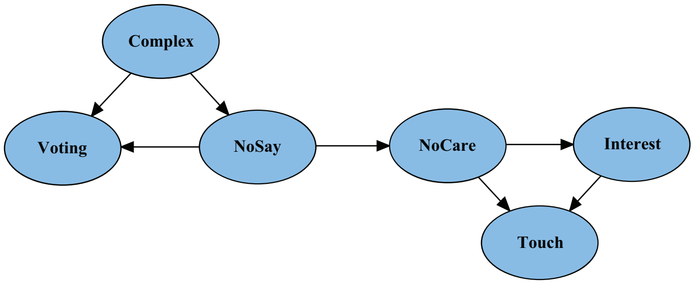

Figure 4: The true DAG for Political Action data

	TPR			**FPR**			**TDR**		
Algorithm	*Sample size*			*Sample size*			*Sample size*		
	50	*100*	*500*	*50*	*100*	*500*	*50*	*100*	*500*
PC	0.38	0.50	0.88	0.11	0.07	0.00	0.76	0.86	0.99
OPC	0.61	0.79	0.95	0.02	0.01	0.00	0.96	0.98	0.99

Table 4: Mean values of performance for PC and OPC algorithm performed on 1000 datasets generated according to the DAG in Figure 4

the sample size is 500 the algorithms behaviour is almost identical.
The SHD values computed in this experiment are presented in Table 5.

	Sample size		
Algorithm	**50**	**100**	**500**
PC	7.22	6.73	5.41
OPC	6.56	6.38	4.50

Table 5: The SHD average for PC and OPC algorithm performed on 1000 datasets generated according to the DAG in Figure 4

The SHD values decrease when the sample size increases and they are smaller if referred to OPC algorithm rather than PC algorithm.

6 Conclusion

This paper dealt with the problem of learning a Bayesian network when subject-matter knowledge is not available. In this situation it is necessary to infer the network from data using automatic learning procedures. The main purpose of this paper was to introduce a new procedure for structural learning in presence of ordinal variables. The new procedure is a variation of the PC algorithm namely OPC algorithm and represents an opportunity to learn the network without demoting ordinal variables in nominal. The OPC algorithm is based on a nonparametric rank-based test appropriate for ordinal variables, the Jonckheere-

Terpstra test. The test is used for checking monotonic trend so the alternative hypothesis is arranged in a specific order. This requires an ordering should be specified before the data are collected. According to the empirical evaluations, in presence of ordinal variables and restricted sample size, the OPC algorithm represents a more suitable solution for the structural learning matter. However, the Jonckheere-Terpstra test only checks for monotone association between ordinal variables. When ordinal variables are associated but not monotonically, the test fails and, in order to catch the association, it is necessary to restore to the G^2 test. A work in progress is the study of an alternative procedure that automatically selects the more suitable test according to the considerations coming from a contingency table analysis.

References

Abellán, J., Gómez-Olmedo, M., and S., S. M. (2006). Some variations on the PC Algorithm. In *Proceedings of Probabilistic Graphical Models*, pp. 1–8.

Agresti, A. (2002). *Categorical Data Analysis*. Wiley-Interscience, New Jersey.

Agresti, A. (2010). *Analysis of ordinal data*. Wiley, New Jersey.

Barnes, S. H. and Kaase, M. (1979). *Political Action: Mass Participation in Five Western Democracies*. Sage Pubblications, Beverly Hills, California.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, **2**, 159–225.

Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Transaction on Knowledge and Data Engineering*, **8**, 195–210.

Chickering, D. M. (1995). A transformational characterization of Bayesian network structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artifcial Intelligence (UAI-95)*, pp. 87–98. Morgan Kaufmann, San Francisco. CA.

Chickering, D. M. (2002). Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research*, **2**, 445–498.

Clogg, C. C. and Shihadeh, E. S. (1994). *Statistical Models for Ordinal Variables*. Sage Pubblications, Thousand Oaks, California.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for constructing Bayesian belief networks from databases. *Machine Learning*, **9**, 309–47.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer, New York.

Cox, D. and Wermuth, N. (1996). *Multivariate Dependencies. Models, analysis and interpretation*. Chapman and Hall/CRC, Boca Raton, Florida.

Dash, D. and Druzdzel, M. J. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 142–9. Morgan Kaufmann.

Dawid, A. P. (1979). Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society, Series B*, **41**, 1–31.

Edwards, D. (1995). *Introduction to Graphical Modelling*. Springer, New York.

Fernandes, C. M., Silva, W. T. D., and Ladeira, M. (2004). An Algorithm to Handle Structural Uncertainties in Learning Bayesian Network.

Heckerman, D. (1995). A tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research.

Heckerman, D., Meek, C., and Cooper, G. (1997). A bayesian approach to causal discovery. Technical Report MSR-TR-97-05.

Jonckheere, A. (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika*, **41**, 133–45.

Lehmann, E. L. (1975). *Nonparametrics: Statistical Methods Based On Ranks.* Holden-Day, San Francisco.

Lindgren, B. W. (1976). *Statistical Theory*, (third edn). Macmillan Publishing, New York.

Neapolitan, R. E. (2003). *Learning Bayesian Networks.* Pearson Prentice Hall, NewHaven, Connecticut.

Pearl, J. (1986). A constraint-propagation approach to probabilistic reasoning. In *Uncertainty in Artificial Intelligence.* (ed.L.N. Kanal and J.F. Lemmer), North Holland, Amsterdam, The Netherlands.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems:networks of plausible inference.* Morgan Kaufmann, San Francisco, CA, USA.

Pirie, W. (1983). Jonckheere tests for ordered alternatives. *Encyclopaedia of Statistical Sciences*, **4**, 315–8.

Siegel, S. and Castellan, N. J. J. (1988). *Nonparametrics: Statistical Methods Based On Ranks.* McGraw-Hill International Editions, New York.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*, (2nd edn). MIT Press, Cambridge, Massachusetts.

Steck, H. (2001). *Constraint-Based Structural Learning in Bayesian Networks using Finite Data*. PhD thesis, Institut für Informatik der Technischen Universität München.

Terpstra, T. J. (1952). The asymptotic normality and consistency of Kendall's test against trend when ties are present in one ranking. *Indagationes Mathematicae*, **14**, 327–333.

Tsamardinos, I., Brown, L. E., and Constantin, F. A. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. In *Proceedings of the 52nd International Statistical Institute*, pp. 31–78. Machine Learning.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 255–70.