

Causal Discovery of Dynamic Bayesian Networks

Cora Beatriz Pérez-Ariza¹, Ann E. Nicholson², Kevin B. Korb²,
Steven Mascaro², and Chao Heng Hu²

¹ Dept. of Computer Science and Artificial Intelligence
University of Granada
Granada, 18071, Spain

² Clayton School of IT
Monash University
Clayton, VIC 3800

Abstract. While a great variety of algorithms have been developed and applied to learning static Bayesian networks, the learning of dynamic networks has been relatively neglected. The causal discovery program CaMML has been enhanced with a highly flexible set of methods for taking advantage of prior expert knowledge in the learning process. Here we describe how these representations of prior knowledge can be used instead to turn CaMML into a promising tool for learning dynamic Bayesian networks.

Keywords: Learning dynamic Bayesian networks, causal discovery, dynamic Bayesian networks.

1 Introduction

Bayesian networks (BNs) have become a valuable tool for applied Artificial Intelligence, where they are effective in reasoning about and under uncertainty, managing uncertainty, and providing better information for planning. Bayesian networks have been widely and successfully applied across the sciences, including economy, biology, and medicine. In simpler cases, a Bayesian network can be built by expert elicitation. However, for many real world applications, BNs are too complex to be fully specified by experts, requiring learning from observational data. A great amount of effort has been spent on research on learning Bayesian networks from data, with many successful programs and algorithms the consequence (see Daly et al., 2011, for a recent review). While the generalization of static to dynamic Bayesian networks (DBNs), representing both “static” causal relations (within one time slice) and dynamic causal relations (across time slices), is relatively straightforward, the generalization of static BN learners to dynamic BN learners is a bit trickier. Here we present some preliminary results applying a static learner, the CaMML causal discovery program (causal discovery via Minimum Message Length), to learning dynamic BNs structure from data.

2 Bayesian Networks

A *Bayesian network* is a directed acyclic graph where each node represents a random variable, and the topology of the graph encodes the independence relations among the variables, according to the d -separation criterion (Pearl, 1988). Given the independencies implied by the graph, the joint distribution is determined by the conditional probability tables (CPTs) associated with each node; thus, for a Bayesian network with variables $X_1, \dots, X_n$:

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p_i(x_i | pa(x_i)), \quad (1)$$

where $pa(x_i)$ denotes a configuration of values of the parents of variable X_i in the network.

2.1 Learning Bayesian Networks

There are two fundamental difficulties in learning Bayesian network structures. One is the complexity of learning Bayesian network structures. The space of Bayesian networks is super-exponential in the number of variables, and the learning problem has been proven to be NP-hard (Chickering, 1996). The other difficulty is that given observational data there is usually no unique Bayesian network that represents observed joint distribution over a set of variables. Various approaches have been developed to make the problems tractable, which can be categorized into constraint-based methods and metric-based methods.

Constraint-based methods aim to discover causal models using conditional independences from statistical tests on data, incorporating heuristic search to find the best structure over the Markov equivalence classes.

Metric-based learners use metrics that score candidate structures embedded in a search procedure aiming to maximize or minimize that score. The Bayesian Information Criterion (BIC) score (?) and the Bayesian Dirichlet (BDe) score (Cooper and Herskovits, 1991) are two popular metrics that have been used in many BN learning techniques. Wallace and Korb (1999) used Minimum Message Length (MML) metric for learning BN in the casual discovery program CaMML.

2.2 The CaMML Learner

CaMML (Causal discovery via Minimum Message Length) attempts to learn the best causal structure to account for the data, using a Minimum Message Length (MML) metric (Wallace, 2005) with a two-phase search over the model space, simulated annealing followed by Markov Chain Monte Carlo (MCMC) search. CaMML has been developed at Monash University over the past 16 years (see Korb and Nicholson, 2011, Ch 9, and O'Donnell, 2010 for a full description). We used the most recent open-source version of CaMML,¹. CaMML supports multiple ways of describing prior information about relationships between variables

¹ <https://github.com/rodneynodonnell/CaMML>

(O'Donnell et al., 2006), each of which can be accompanied by a confidence level. The types of priors are:

- Full structure. An expert may supply a fully specified network for all or subnetwork.
- Direct causal connections between variables may be indicated (e.g., $A \rightarrow B$).
- Direct relation ($A - B$). I.e., there is an arc between two variables, but the direction is unknown.
- Causal dependency ($A \Rightarrow B$). One variable is an ancestor of the other.
- Correlation ($A \sim B$). There is some connection between the nodes, which may be a causal dependency in either direction or via a common ancestor.
- Temporal order ($\{A_1, A_2, \dots, A_m\} \prec \{B_1, B_2, \dots, B_n\}$). This sorts variables into tiers, a partial ordering based on the notion that tiers separate the variables on a timeline and that causality only occurs in the forward time direction. This is the most commonly allowed prior constraint in BN learning (e.g., Tetrad IV and K2). Thus $\prec$ means that the A_i occur before the B_j ; for CaMML this becomes the structural constraint that the A_i cannot be descendants of B_j .

The expert may provide CaMML with any combination of the priors above, together with the confidence in each (expressed as a probability).² CaMML then combines these into an MML code for a prior probability distribution over the model space, as explained, for example, in Korb and Nicholson (2011, Chap 9).

3 Dynamic Bayesian Networks

A *dynamic Bayesian network* (DBN) represents a system that changes over time. A static model over the variables $X_1, \dots, X_n$ is repeated for each time slice T_i .³ To distinguish between corresponding variables across time slices, we will denote them $X_n^{T_i}$, meaning variable X_n during time slice T_i . The time slices are connected through *temporal links*, that are the same for every consecutive pair of time slices. Usually the temporal links only connect one time slice with the next (so, the order 1 Markov assumption holds). Finally, the conditional probabilities don't change over time (the system is assumed to be stationary). A DBN can be specified very compactly:

- Set of node names, defined as $\Omega_X = \forall n \forall i X_n^{T_i}$
- Intra-slice links, that connect variables in the same time slice
- Temporal (inter-slice) links
- CPTs for the first time slice
- CPTs for the second time slice (when parents may be either from T_i or T_{i+1} time-slices)

² Some other BN learners also support the use of structural knowledge, but they have been limited to specifying one or two of these kinds of priors.

³ I.e., DBNs are stationary systems. The flexibility of CaMML would allow us also to look at learning *non-stationary* dynamic models, which we may do in the future.

3.1 Learning Dynamic Bayesian Networks

Friedman et al. (1998) were the first to look at automated learning of DBNs, extending both the BIC score and the BDe score to apply them to DBNs. They decomposed the learning problem into learning first the static network, representing time slices, and then the transition network relating one time slice to the next, applying Friedman’s structural EM algorithm to the learning tasks. Tucker and Liu (1999, 2003) applied evolutionary algorithms with an MDL score to DBN learning. Other techniques that have been applied include branch-and-bound (de Campos and Ji, 2011) and greedy search (Van Berlo et al., 2003).

Comparison with these methods is made difficult by the lack of a common set of learning problems and lack of access to the programs used (although the EM MDL approach is available in Murphy’s BNT). Here we will simply explore using CaMML for DBN learning in comparison with the PC algorithm in GeNIe. We chose CaMML because, not only are we familiar with it, but it has the widest and most flexible range of representations for prior learning constraints of the available BN learning programs (see §2.2), allowing DBN constraints to be readily incorporated.

4 Examples of Dynamic Bayesian Networks

To explore the possibilities of learning DBNs with CaMML, we have taken two Bayesian networks from the literature, modified to represent stationary models with two time slices. The CPTs were handcrafted.

4.1 eMilk Dynamic Bayesian Network

Milk from a cow may be infected. There is a test to detect whether the milk is infected, which may give either a positive or a negative test result. The test is not perfect. It may give a positive result on clean milk as well as a negative result on infected milk (Jensen and Nielsen, 2007).

We have added some weather conditions that may affect the milk, namely temperature and humidity. These two variables are summarized in a weather variable, that affects directly the quality of the food eaten by the cow and indirectly the probability of an infection in the milk. In our model high humidity increases the probability of an infection, and the temperature has an impact on the accuracy of the test, with extreme temperatures increasing the chances of a malfunction. The model can be extended over time to observe this situation over time. We duplicated the original structure to obtain two time slices, with some *temporal links* between them. The full model is shown in Fig.1.

4.2 eMetCancer Dynamic Bayesian Network

Metastatic cancer is a possible cause of brain tumors and is also an explanation for increased total serum calcium. In turn, either of these could

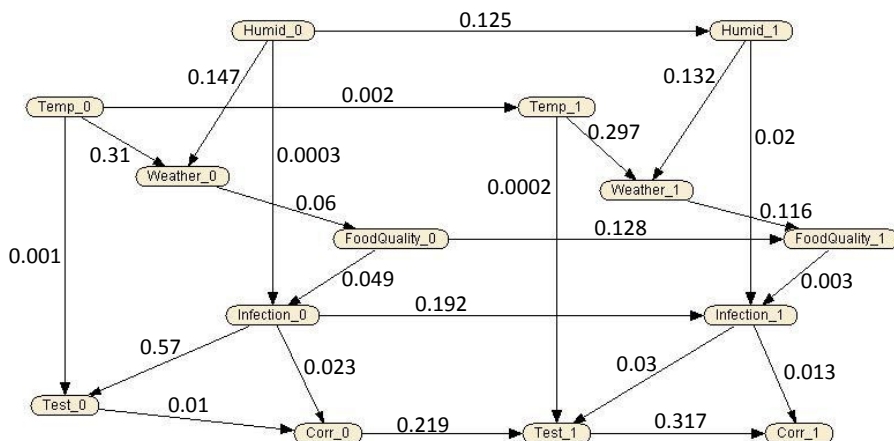


Fig. 1. Handcrafted extended milk model (eMilk)

explain a patient falling into a coma. Severe headache is also associated with brain tumors. (Cooper, 1984)

This situation can also be observed over time. For example the presence of a brain tumor at one time may be due to an earlier tumor metastasizing. We have expanded this model into two time slices by duplicating the original structure and adding five temporal links to it. Note that the model has not been checked with an expert and so it may not represent the medical reality. The resultant model is shown in Fig. 2.

4.3 Sensitivity Analysis

We performed sensitivity analysis on the models in order to determine the *connection strengths* of the links in the dynamic Bayesian networks (Jitnah, 1999; Boerlage, 1992). This allowed us to check the strengths of the links that CaMML was not able to detect. We obtained the mutual information between each pair of connected nodes using Netica's M.I. function.⁴ These strengths are specified in Fig.1 and Fig.2 as links weights. As shown, we designed both examples to have a mixture of weak and strong arcs.

⁴ See www.norsys.com. More specifically, the Netica M.I. measure is over all paths between two nodes; we limited it to the direct connection path by d-separating the nodes on any other paths, by averaging over the Markov blanket (excluding the path in question).

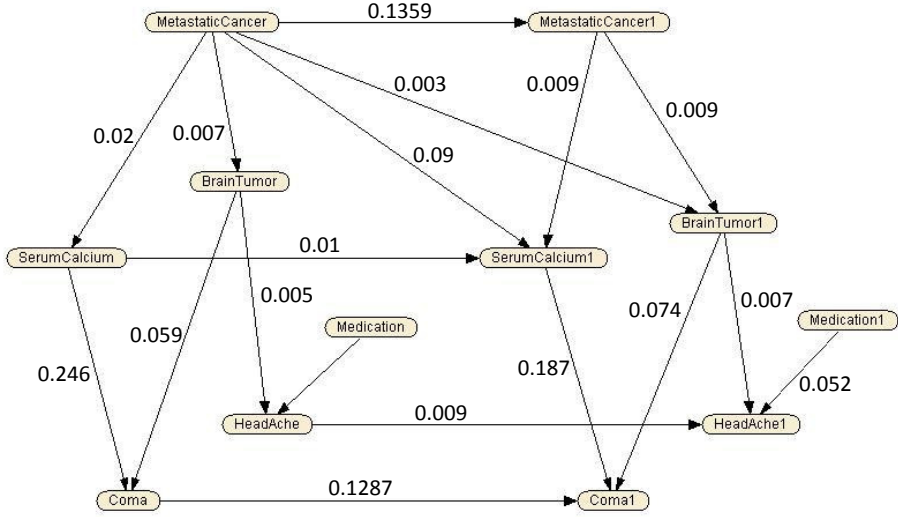


Fig. 2. Handcrafted extended Metastatic Cancer model (eMetCancer)

5 Experimental Evaluation

5.1 Methodology

In order to learn dynamic Bayesian networks with CaMML, we incorporated as prior knowledge constraints that define DBNs. The most obvious such constraint is to group variables within a time slice in a tier, so all the variables of one time slice must precede all those in the next. Thus we define the temporal order: $\{X_1^{T_0}, X_2^{T_0}, \dots, X_i^{T_0}\} \prec \{X_1^{T_1}, X_2^{T_1}, \dots, X_i^{T_1}\}$.

A more elaborate approach to learning DBNs is shown in Algorithm 1. The idea here is to learn a static Bayesian network for a single time slice; duplicate it, identifying it as the subnetwork for both time slices, by fixing its causal arcs as prior constraints with confidence 1 (as CaMML allows us to do); learn the temporal links connecting the two tiers.

We used three sample sizes for most of our experiments: 1000, 10000 and 100000 cases. CaMML was run with the default of $201n^3$ iterations in its search (where n is the number of variables).

For comparison, we also learned models using the version of the PC algorithm implemented in GeNIe,⁵ employing both tier priors and Algorithm 1. GeNIe permits the incorporation of certain types of prior information, in particular temporal tiers and the hard constraint of arcs that can or can't be in the model. Since PC does not direct all arcs learned, when measuring the quality of these

⁵ From the Decision Systems Laboratory of the University of Pittsburgh <http://dsl.sis.pitt.edu/>.

models, we penalized undirected arcs (when present in the original model) with half the penalty used for standard errors.

Algorithm 1: Learn a DBN with CaMML using learned prior information

Input: A database D
Output: $\mathcal{DBN}_D$, a DBnet for D

```

1 begin
2   // Divide  $D$  into  $D^{T_1}$  and  $D^{T_2}$ 
3   Let  $D^{T_1}$  = data from variables in  $T_1$ 
4   Let  $D^{T_2}$  = data from variables in  $T_2$ 
5   Let  $X_1...X_n$  be the generic variables in a time slice
6   Change variables in  $D^{T_1}$  and  $D^{T_2}$  for their equivalents in  $X_1...X_n$ 
7   // Note that  $D_{generic}$  will have twice as much data as  $D^{T_1}$  and  $D^{T_2}$ 
8   Make  $D_{generic} = D^{T_1} \cup D^{T_2}$ 
9   Learn  $G_{generic}$  from  $D_{generic}$  using CaMML
10  foreach time slice  $T_k, k = 1, 2$  do
11    foreach pair of variables  $X_i, X_j$  in  $T_i$  do
12      if  $X_i \rightarrow X_j \in G_{generic}$  then
13        | set the intra-slice link  $X_i^{T_k} \rightarrow X_j^{T_k}$  as prior with confidence 1
14      else
15        | set the intra-slice link  $X_i^{T_k} \rightarrow X_j^{T_k}$  as prior with confidence 0
16  Set tier for the temporal order ( $\{X_1^{T_1}, X_2^{T_1}, ...X_n^{T_1}\} \prec \{X_1^{T_2}, X_2^{T_2}, ...X_n^{T_2}\}$ )
17  Learn temporal links with CaMML using the priors defined in steps 10-15
    and the tier defined in step 16, using database  $D$ 
18  Set the result as  $\mathcal{DBN}_D$ 
19  return  $\mathcal{DBN}_D$ 

```

5.2 Evaluation

To evaluate the different approaches to learning we measured how close learned structures were to the original networks used to generate the data using edit distances. For learning static BNs, this can be as simple as counting the minimal number of arc deletions, additions and reversals to go from one model to the other.⁶ DBNs require some adjustment to the edit distance, however.

Edit Distance for DBNs (eED). In a stationary DBN, all time slices have the same structure, so if during the learning process this restriction is violated, the traditional computation of the edit distance does not detect it. We modify

⁶ Even for static BNs this is too simple, ignoring the difference between reversals that leave models Markov equivalent and those that don't. In any case, however, edit distance will remain crude compared to Kullback-Leibler divergence and other measures. For a discussion see Korb and Nicholson, 2011, Sec 9.8.

this metric by counting the errors in just the second slice and the errors in the temporal links separately. These links suffice to reconstruct the entire DBN, excepting the first time slice, whose importance disappears in a sufficiently large expansion of the DBN. Thus we used:

$$eED(N, N') = w_t E(N_t, N'_t) + w_s E(N_s, N'_s)$$

where N, N' are the original and learned networks respectively, $E(\cdot, \cdot)$ measures static BN edit distance, w_t is the weight assigned to temporal links and w_s that for static links, N_t is the original network restricted to temporal links, N_s is the same restricted to static links in the second tier, and N'_s and N'_t the corresponding learned subnetworks. For this work we set $w_t = w_s = 1$.

5.3 Results and Discussion

Some of our results are shown in Table 1. A more detailed view of results for eMilk without priors is given in Table 2. If we look only at the outcome of the experimentation with no priors and with tier priors, an inspection of the learned BNs for eMilk (e.g., Figure 3) shows that when arcs are missing, these arcs are the ones with the lower M.I. score, e.g., $Temp_0 \rightarrow Test_0$, $Temp_1 \rightarrow Test_1$ and $Humid_0 \rightarrow Infection_0$. In the case of the eMetCancer model, the errors are always found in the temporal link set.

While we have a general expectation that learning with more samples will reduce errors, interestingly this is not always the case. For example, GeNIe no priors, the eED for eMilk goes from 4.25 to 7.5 from 1000 to 100000 sample points. In such cases, the increase in the eED is commonly due to errors of commission, with spurious arcs being added. This happens for both networks and for learning with and without priors, suggesting that especially GeNIe's PC is overfitting its data. For a detailed comparison of PC and CaMML's learning performance, see Dai et al. (1997).

Tier priors for GeNIe reduced the eED in all cases. For CaMML, priors, whether with tiers or Algorithm 1, eliminated errors with eMetCancer, but did

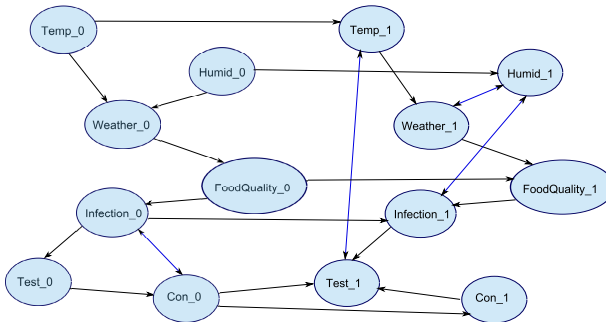


Fig. 3. Extended milk model learned with the PC algorithm, with a database of 10000 cases

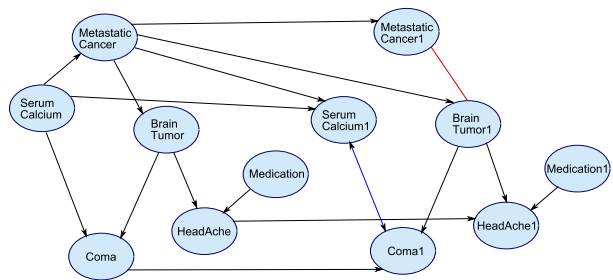


Fig. 4. Extended Metastatic Cancer model learned with the PC algorithm, with a database of 10000 cases and the temporal tier

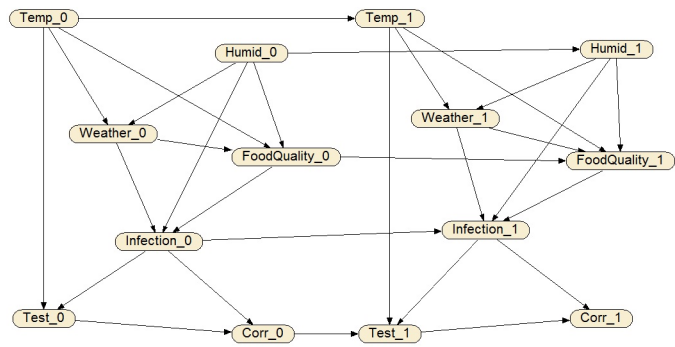


Fig. 5. Extended milk model learned by CaMML with Alg. 1 using a 100000 cases database

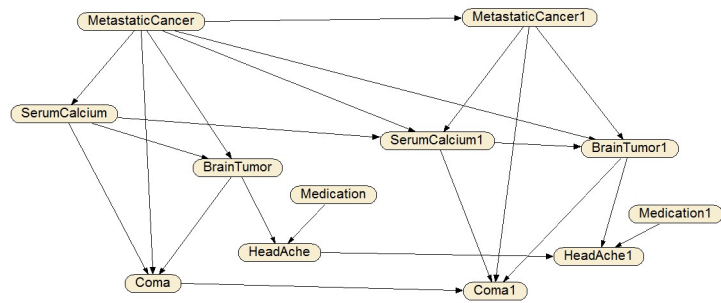


Fig. 6. Extended Metastatic Cancer model learned by CaMML with Alg. 1 using a 100000 cases database

Table 1. Edit distance for DBNs (eED) obtained for the different models learned

		CaMML		GeNIe	
		eMilk	eMetCancer	eMilk	eMetCancer
no priors	1000	0.95	2.6	4.5	4.5
	10000	0	0.5	2	4.5
	100000	0	0.5	7.5	1
tier	1000	0.975	2.4	3.5	2.5
	10000	0	0	3	1.25
	100000	0	0	4.5	0.25
Algorithm 1	1000	1.5	0	5.5	5
	10000	0	0	4.5	3
	100000	0	0	9	4.5

Table 2. Mean errors (omissions, commissions, reversals) and edit distances (eED) for eMilk without priors (reporting standard deviations for CaMML over twenty runs; PC is deterministic)

		CaMML				GeNIe			
		om	com	rev	eED (sd)	om	com	rev	eED
no priors	100	4.2	0	1.65	3.425 (0.43)	9	2	0	5.5
	1000	1.9	0	0	0.95 (0.15)	4	4	1	4.5
	10000	0	0	0	0 (0.0)	2	1	1	2

not make a difference for eMilk, where the learning is exact given 10k or larger samples. Both the use of tiers and Algorithm 1 appear to provide a satisfactory basis for learning dynamic networks.

Table 2 shows the results for eMilk without priors in more detail (other cases show similar results). In order to examine variation, we ran CaMML twenty times and estimated its mean and standard deviation in eED (PC is deterministic). The results show differences in performance between CaMML and PC at all sample sizes that are statistically significant.

6 Conclusions and Future Work

In this paper we have shown how a static BN learner with structural priors can be adapted to learn dynamic Bayesian networks, which explicitly model changes over time. We have presented experimental results applying two such learners, CaMML (a metric-based learner) and GeNIe (which uses conditional independence learning). Applying temporal tiers can result in networks with different structures in each time slice. An alternative two-stage algorithm ensures the connections within a slice remain the same, but may lead to other errors in the network structure. Our results also show that CaMML clearly outperforms GeNIe PC.

The obvious next step is to compare the performance of the static learners used with priors and Algorithm 1 with tailored DBN learners such as that provided with Murphy's BNT software (Murphy, 2001). There are further opportunities for using CaMML's flexible priors to boost DBN learning. Finally, CaMML's scoring metric may be enhanced to deal with DBN features intrinsically. While this work is very preliminary, it clearly points out a promising direction for future research.

Acknowledgments. This work has been partially funded by the FPI scholarship programme (BES-2008-002049).

References

- Boerlage, B.: Link Strengths in Bayesian networks. Master's thesis, University of British Columbia (1992)
- Chickering, D.M.: Learning Bayesian networks is NP-complete. In: Fisher, D., Lenz, H.-J. (eds.) *Learning from Data: AI and Statistics V*, ch. 12, pp. 121–130. Springer (1996)
- Cooper, G.: NESTOR: A Computer-Based Medical Diagnostic Aid That Integrates Causal and Probabilistic Knowledge. Ph. D. thesis, Stanford (1984)
- Cooper, G., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Technical Report Stanford KSL 9 (June 1991)
- Dai, H., Korb, K., Wallace, C., Wu, X.: A study of causal discovery, with weak links and small samples. In: *International Joint Conference on Artificial Intelligence*, vol. 15, pp. 1304–1309 (1997)
- Daly, R., Shen, Q., Aitken, S.: Learning Bayesian networks: Approaches and issues. *The Knowledge Engineering Review* 26, 99–157 (2011)
- de Campos, C., Ji, Q.: Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research* 12, 663–689 (2011)
- Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Cooper, G.F., Moral, S. (eds.) *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, July 24–26, pp. 139–147. Morgan Kaufmann, San Francisco (1998)
- Jensen, F.V., Nielsen, T.D.: *Bayesian networks and decision graphs*. Springer (2007)
- Jitnah, N.: Using Mutual Information for Approximate Evaluation of Bayesian Networks. Ph. D. thesis, Monash University (1999)
- Korb, K.B., Nicholson, A.E.: *Bayesian Artificial Intelligence*, 2nd edn. Chapman & Hall/CRC, Boca Raton (2011)
- Murphy, K.: The Bayes net toolbox for MATLAB. *Computing Science and Statistics* 33, 1024–1034 (2001)
- O'Donnell, R.: Flexible Causal Discovery with MML. In: *Faculty of Information Technology (Clayton)*, Monash University, Australia 3800 (2010)
- O'Donnell, R.T., Nicholson, A.E., Han, B., Korb, K.B., Alam, M. J., Hope, L.R.: Causal Discovery with Prior Information. In: Sattar, A., Kang, B.-H. (eds.) *AI 2006. LNCS (LNAI)*, vol. 4304, pp. 1162–1167. Springer, Heidelberg (2006)
- Pearl, J.: *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, San Mateo (1988)
- Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* 14, 461–464 (1978)

- Tucker, A., Liu, X.: Extending evolutionary programming methods to the learning of dynamic Bayesian networks. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H. (eds.) *Proceedings of the Genetic and Evolutionary Computation Conference* (1999)
- Tucker, A., Liu, X.: Learning Dynamic Bayesian Networks from Multivariate Time Series with Changing Dependencies. In: Berthold, M., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) *IDA 2003. LNCS*, vol. 2810, pp. 100–110. Springer, Heidelberg (2003)
- Van Berlo, R., Van Someren, E., Reinders, M.: Studying the conditions for learning dynamic Bayesian networks to discover genetic regulatory networks. *Simulation* 79, 689–702 (2003)
- Wallace, C.S.: *Statistical and Inductive Inference by Minimum Message Length*. Springer, Berlin (2005)
- Wallace, C.S., Korb, K.B.: Learning linear causal models by MML sampling. In: Gamerman, A. (ed.) *Causal Models and Intelligent Data Management*, pp. 89–111. Springer (1999)