Efficient Learning of Bayesian Network Classifiers

An Extension to the TAN Classifier

Alexandra M. Carvalho[1], Arlindo L. Oliveira[1], and Marie-France Sagot[2]

[1] IST, TULisbon/INESC-ID, Lisboa, Portugal
{asmc,aml}@inesc-id.pt
[2] Inria Rhône-Alpes, Université Claude Bernard, Lyon I, France
Marie-France.Sagot@inria.fr

Abstract. We introduce a Bayesian network classifier less restrictive than Naive Bayes (NB) and Tree Augmented Naive Bayes (TAN) classifiers. Considering that learning an unrestricted network is unfeasible the proposed classifier is confined to be consistent with the breadth-first search order of an optimal TAN. We propose an efficient algorithm to learn such classifiers for any score that decompose over the network structure, including the well known scores based on information theory and Bayesian scoring functions. We show that the induced classifier always scores better than or the same as the NB and TAN classifiers. Experiments on modeling transcription factor binding sites show that, in many cases, the improved scores translate into increased classification accuracy.

1 Introduction

Learning Bayesian networks has been a hot and fruitful research topic [23, 15, 3, 17]. The goal of learning a Bayesian network is to find both the structure and the parameters of the network that best fit the data, according to a given scoring function. The inference of a general Bayesian network has been shown to be an NP-hard problem [5], even for approximate solutions [7]. As a consequence, heuristic algorithms became the standard methodology for addressing this problem [23]. A common approach is to impose restrictions over the network structure. In this context, two results set the border between efficient and non-efficient structure learning. In one hand, Chow and Liu showed that trees can be learned in polynomial time [4], on the other hand, Dasgupta proved that learning 2-polytrees is NP-hard [8].

Bayesian networks have been widely used in the context of classification [21, 16, 14]. The simplicity and high accuracy of the *Naive Bayes* (NB) classifier [11] have led to its extensive use, and to several attempts to extends it. In this line of research Friedman et al [14] proposed the *Tree Augmented Naive Bayes* (TAN) classifier in order to overcome the strong independence assumptions imposed by the NB network. The TAN is an extension of NB which allows additional edges

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 16–25, 2007.

between the attributes of the network in order to capture correlations among them. Such correlations are however restricted to a tree structure. Friedman et al showed that TAN was indeed more accurate than NB on benchmark datasets.

Herein, we introduce a Bayesian network classifier less restrictive than NB and TAN classifiers. Considering that learning an unrestricted network is unfeasible, the underlying graph of the proposed classifier is confined to be consistent with the *breadth-first search* (BFS) order of an optimal TAN and to have a bounded in-degree, henceforward called *BFS-consistent k-graph* (BCkG). We show that learning BCkG's can be done efficiently and for any scoring functions that decomposes over the network structure. Well known scores with this property are those based on information theory, such as *log likelihood* (LL), *Akaike information criterion* (AIC), *Bayesian information criterion* (BIC), *minimum description length* (MDL); and Bayesian scoring function such as K2, *Bayesian Dirichlet* (BD) and its variants (BDe, BDeu), and *mutual information test* (MIT). We show that the classifiers induced from BCkG's score always better than or the same as the NB and TAN classifiers, and moreover that the search space of the learning algorithm is exponentially larger than the TAN learning algorithm. We check the quality of our approach with biological data. Experiments show that, in many cases, the improved scores translate into increased classification accuracy.

The paper is organized as follows. In Section 2, we briefly revise Bayesian networks, Bayesian network classifiers and their learning algorithms. In Section 3, we introduce the main contribution of this paper, the learning algorithm for BCkG classifiers. In Section 4, we apply our approach in the realm of computational biology, namely to model transcription factor binding sites, and present some experimental results. Finally, in Section 5 we draw some conclusions and discuss future work.

2 Basic Concepts and Results

In this section we introduce some notation, while recalling relevant concepts and results concerning Bayesian networks which are directly related with the contribution of this paper.

2.1 Bayesian Networks

A *Bayesian network* is a triple $B = (\mathbf{X}, G, \Theta)$. The first component $\mathbf{X} = (X_1, \ldots, X_n)$ is a finite random vector where each random variable X_i ranges over a finite domain D_i. We denote the joint domain $\mathbf{D} = \Pi_{i=1}^{n} D_i$. The second component $G = (N, E)$ is a directed acyclic graph with nodes $N = \{X_1, \ldots, X_n\}$ and edges E representing direct dependencies between the variables. The third component Θ encodes the parameters $\{\theta_{x_i|\Pi_{x_i}}\}_{\mathbf{x}\in\mathbf{D}}$ of the network, where $\theta_{x_i|\Pi_{x_i}} = P_B(x_i|\Pi_{x_i})$ for each possible value x_i of X_i, and Π_{x_i} of Π_{X_i}, where Π_{X_i} denotes the set of parents of X_i in G. A Bayesian network defines a unique joint probability distribution over $\mathbf{X}$ given by

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} \theta_{X_i|\Pi_{X_i}}. \quad (1)$$

We denote the set of all Bayesian networks with n variables by $\mathcal{B}_n$.

Informally, a Bayesian network encodes the independence assumptions over the component random variables of $\mathbf{X}$. An edge (i, j) in E represents a direct dependency of X_j to X_i. Moreover X_i is independent of its non descendants given its parents Π_{X_i} in G.

The problem of learning a Bayesian network given data T consists on finding the Bayesian network that best fits the data T. In order to quantify the fitting of a Bayesian network a *scoring function* $\phi : \mathcal{B}_n \times \mathbf{D}^m \rightarrow \mathbb{R}$ is considered. In this context, the problem of learning a Bayesian network can be recasted to the following optimization problem. Given a dataset $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and a scoring function ϕ, the *problem of learning a Bayesian network* is to find a Bayesian network $B \in \mathcal{B}_n$ that maximizes the value ϕ for T.

Several scoring functions have been proposed in the literature [6, 17, 18, 22]. The discussion of the advantages and disadvantages of each of these functions is outside the scope of this paper.

2.2 Bayesian Network Classifiers

A *Bayesian network classifier* is a Bayesian network where $\mathbf{X} = (X_1, \ldots, X_n, C)$. The variables $X_1, \ldots, X_n$ are called *attributes* and C is called the *class variable*. Moreover, the graph structure G is such that the class variable has no parents, that is, $\Pi_C = \emptyset$, and all attributes have at least the class variable as parent, that is, $C \in \Pi_{X_i}$. The corresponding classifier is defined as

$$\arg\max_C P_B(C|X_1, \ldots, X_n).$$

We therefore reformulate the model to make it more tractable. Using the definition of conditional probability and Equation (1) leads to the following classifier:

$$\arg\max_C P_B(C) \prod_{i=1}^{n} \theta_{X_i|\Pi_{X_i}}.$$

Informally, the problem of learning a Bayesian network classifier can be recasted as the problem of learning a Bayesian network where all attributes have the class variable as parent.

Naive Bayesian Network Classifier. A *naive Bayesian network* (NB) [11] is a Bayesian network classifier where each attribute has the class variable as its unique parent, that is, $\Pi_{X_i} = \{C\}$ for all $1 \leq i \leq n$. Since the NB has a fixed graph structure, learning the network reduces to computing the empirical distribution.

The NB classifier is one of the most effective classifiers, in the sense that, in many cases, its predictive performance is competitive with state-of-the-art classifiers [10, 9]. In fact, the NB classifier is computationally undemanding and

shows an unexpected accuracy in many applications. However, the independence assumption is too strict and relaxing this assumption may lead to more accurate classification.

Tree Augmented Naive Bayesian Network Classifier. A *tree augmented naive Bayesian network* (TAN) [14] is a Bayesian network classifier where there exists an $r \in \{1, \ldots, n\}$ such that $\Pi_{X_r} = \{C\}$ and $\Pi_{X_i} = \{C, X_j\}$ for all $1 \leq i \leq n$ with $i \neq r$. The TAN was first proposed by Friedman et al [14] to overcome the strong independence assumptions imposed by the NB network. In fact, the TAN is an extension of NB which allows additional edges between the attributes of the network in order to capture correlations among them. Such correlations are however restricted to a tree structure.

In [14] an algorithm to find an optimal TAN that maximizes the LL is given. The main idea is to consider a complete weighted undirected graph, where each edge between X_i and X_j is weighted with the conditional mutual information between X_i and X_j given the class variable C. Given this, the problem reduces to determining a maximal weighted spanning tree, using the algorithm by Chow and Liu [4]. After computing such spanning tree, a direction has to be assigned to each edge of the tree. This is done by choosing an arbitrary attribute as the tree root and then setting the direction of all edges to be outward from it.

Extending TAN Classifier to Deal with Decomposable Scores. The TAN was originally proposed for maximizing the LL score but it can be easily adapted to deal with any scoring function that is *decomposable* and *score equivalent*. We recall that a scoring function ϕ is *decomposable* if it can be written as

$$\phi(B, T) = \sum_{i=1}^{n} \phi_i(\Pi_{X_i}, T). \tag{2}$$

Moreover, a scoring function is said to be *score equivalent* if it assigns the same value to all directed acyclic graphs that are represented by the same essential graph. All interesting scoring functions in the literature are decomposable, since it is unfeasible to learn undecomposable scores. LL, AIC, BIC and MDL are decomposable and score equivalent, whereas K2, BD, BDe, BDeu and MIT are decomposable but not score equivalent.

According to Heckerman et al [17], finding an optimal TAN classifier for decomposable and score equivalent scoring functions reduces to weighting each undirected edge between X_i and X_j by $\phi_j(\{X_i, C\}, T) - \phi_j(\{C\}, T)$, which is equal to $\phi_i(\{X_j, C\}, T) - \phi_i(\{C\}, T)$ by score equivalence of ϕ, and to find a maximal weighted (undirected) spanning tree. Moreover, learning an optimal TAN classifier for scoring functions that are only decomposable, but not score equivalent, can also be done in polynomial time. In this case, however, an edge between X_i and X_j may score differently depending on its direction, and so a directed spanning tree must be found (instead of an undirected one). The idea is to weight each directed edge from X_i and X_j with $\phi_j(\{X_i, C\}, T) - \phi_j(\{C\}, T)$ and then, for each node X_r, find an optimal directed spanning tree rooted at X_r with Edmonds' algorithm [12]

3 BFS-Consistent Bayesian Network Classifiers

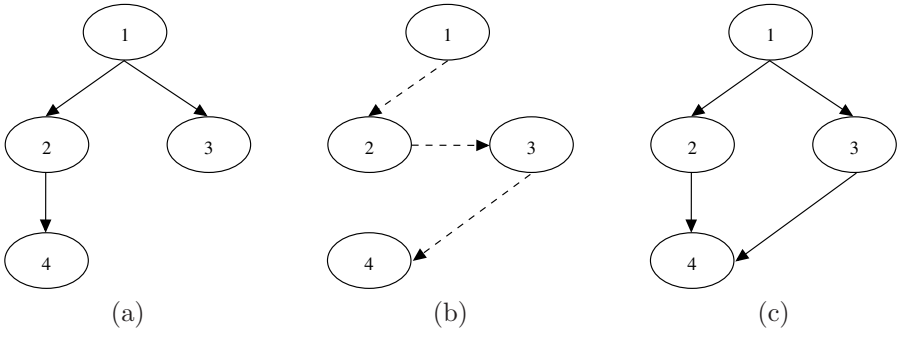
We now introduce the main contribution of this paper, a simple and effective heuristic for a causality order between the attributes based on a *breadth-first search* (BFS) over an optimal TAN. The main idea is to take the total order induced by the BFS over an optimal TAN and then search for an optimal network (of bounded in-degree) consistent with it. It is easy to show that the score of the resulting network is always greater than or equal to the score of TAN and NB. The foremost benefit of this approach is that learning such an optimal network can be done efficiently, that is, in polynomial time over the number of attributes. Moreover, the class of the networks consistent with the BFS order is exponentially larger, in the number of variables, when compared to TAN networks.

We start by introducing some auxiliary concepts. A *k-graph* is a graph where each node has in-degree at most k. Trees and forests are 1-graphs.

Definition 1 (BFS-consistent k-graph). Given a TAN R with a set of attributes N, a graph $G = (N, E)$ is said to be a *BFS-consistent k-graph* (BCkG) w.r.t R if it is a k-graph and for any edge in E from X_i to X_j the node X_i is visited in *breadth-first search* (BFS) of R before X_j. Henceforward, we denote by $\mathcal{B}_R^k$ the set of all BCkG's w.r.t. R.

The above definition of consistency imposes that there can only exist an edge from X_i to X_j in $G \in \mathcal{B}_R^k$ if X_i is less than or as deep as X_j in R. We assume that if $i < j$ and X_i and X_j are at the same level, then the BFS over R reaches X_i before X_j. Other approaches to order attributes at the same level are discussed in the conclusions.

Example 1. Given the underlying graph for the attributes of a TAN R in (a), its BFS is represented by a dashed line in (b). A BC2G *w.r.t R is presented in (c).*



The core idea of the BCkG learning algorithm is to compute an optimal TAN R and improve it by adding/removing dependencies which were omitted/present because of the TAN structure restrictions. For efficiency purposes, the modified model must be a BFS-consistent k-graph w.r.t. R. In this context, the total order induced by the BFS over R might add dependencies from higher nodes to deeper

nodes. In detail, the algorithm starts by computing an optimal TAN as described in Section 2.2. Then it performs a BFS over the optimal TAN to construct a total order. Finally, it ranges over each attribute X_i, generates the set α_i of all attributes less than X_i, and takes as parents of X_i the set $S \subseteq \alpha_i$ such that $\phi_i(S \cup \{C\}, T)$ is maximal over all subsets of α_i with at most k attributes. The pseudocode of the algorithm is presented in Algorithm 1.

Algorithm 1. Learning BCkG network classifiers

1. Run a (deterministic) algorithm that outputs an optimal TAN R according to ϕ.
2. Compute the total order $\sqsubseteq$ induced by the BFS over R (ignoring the class variable).
3. For each attribute X_i in R do:
   (a) Compute the set $\alpha_i = \{X_j \in R : X_i \sqsubseteq X_j \text{ and } X_i \neq X_j\}$.
   (b) For each subset S of α_i with at most k attributes:
      i. Compute $\phi_i(S \cup \{C\}, T)$.
      ii. If $\phi_i(S \cup \{C\}, T)$ is the maximal score for X_i, set $\Pi_i = S \cup \{C\}$.
4. Output the directed graph G such that the parents of an attribute X_i are Π_i.

Theorem 1 (Soundness). Algorithm 1 constructs a BCkG Bayesian network classifier that maximizes the ϕ-score given data T.

Proof. Since all potential parents for each node are checked, the algorithm returns the k-graph G BSF-consistent w.r.t R with the highest score. Moreover, this graph is acyclic since the parents of a node X_i must be in α_i, that is, must belong to the path in R from its root to X_i (excluding X_i). Moreover, it is easy to see that for any path $X_{i_1}, X_{i_2}, \ldots X_{i_k}$ in G we have that $X_{i_j} \in \alpha_{i_k}$ for $1 \leq j < k$. If there existed a cycle $X_{i_1}, X_{i_2}, \ldots X_{i_1}$ it would imply that $X_{i_1} \in \alpha_{i_1}$ which is absurd. □

Proposition 1. Algorithm 1 constructs a BCkG Bayesian network classifier whose ϕ-score is always greater than, or equal to, the ϕ-score of the optimal TAN.

Proof. Start by noticing that the soundness of Algorithm 1 assures that the resulting BCkG w.r.t R is the maximal among all BCkG's in $\mathcal{B}^k_R$. Moreover, observe that the underlying graph G_R of the TAN R (without the class variable) is BFS-consistent w.r.t R, that is, $G_R \in \mathcal{B}^k_R$ for all $k \geq 1$. Hence, the soundness of Algorithm 1 guarantees that the BN classifier B_G, constructed by adding an edge from the class variable C to all attributes in the output k-graph G, is such that $\phi(B_G, T) \geq \phi(R, T)$. □

Theorem 2 (Complexity). Algorithm 1 constructs a BCkG Bayesian network classifier in $O(n^{k+1}\gamma(k, T))$ time where $\gamma(k, T)$ is an upper bound for computing $\phi_i(S \cup \{C\}, T)$.

Proof. Step 2 takes $O(n)$ time. Step 3a) takes $O(n)$ time, while step 3b) takes $O(n^k\gamma(S, T))$ time because it ranges over all subsets S with at most k elements

(which takes $O(n^k)$ time) and for each of this sets it computes $\phi_i(S \cup \{C\}, T)$ (which takes $O(\gamma(S,T))$ time. Thus, the overall complexity of the algorithm is $O(n^{k+1}\gamma(S,T))$ time. □

The theorems above assert the soundness and polynomial-time bound of the BCkG learning algorithm. At this point it remains to show that, despite considering an optimal TAN to confine the search space, the number of graphs searched increases exponentially, in the number of attributes, when compared to TAN's.

Proposition 2. Let R be a TAN with n attributes, then the number of non-trees in $\mathcal{B}_R^k$ is at least $2^{nk-\frac{k^2}{2}-\frac{k}{2}-1}$ when $n \geq k$.

Proof. We denote by $(N, \sqsubseteq)$ the total order induced by BFS over R (ignoring the class variable). Since, this order is total (also called a linear), for any pair of nodes X_i and X_j in N, with $i \neq j$, we can say that a node X_i is *lower than* X_j if and only if $X_i \sqsubseteq X_j$. Given this, notice that the i-th node of R has precisely $(i-1)$ lower nodes. We conclude that, when $i > k(\leq n)$, there are at least 2^k subsets of N with at most k lower nodes. Moreover, when $(1 \leq)i \leq k$, only 2^{i-1} subsets of N with at most k lower nodes exist. Thus,

$$|\mathcal{B}_R^k| \geq \left(\prod_{i=k+1}^{n} 2^k\right) \times \left(\prod_{i=1}^{k} 2^{i-1}\right) = 2^{nk-\frac{k^2}{2}-\frac{k}{2}}$$

give us a lower bound for the total number of possible BCkG w.r.t R (recall that a BC1G is also a BC2G, both a BC1G and a BC2G are also a BC3G, and so on). Now, consider that X_i is the root, and X_j is the child of the root in R. The only two subsets of N with at most k lower elements than X_j are $\emptyset$ and $\{X_i\}$. This choice splits in two all BCkG's in $\mathcal{B}_R^k$. Those for which the set of parents of X_j is $\emptyset$ cannot be trees since X_i has no parents as well. Therefore, there are at lest $\frac{|\mathcal{B}_R^k|}{2} \geq 2^{nk-\frac{k^2}{2}-\frac{k}{2}-1}$ in $\mathcal{B}_R^k$ that are non trees. □

4 Experimental Methodology and Results

We compared the BC2G classifier with TAN and NB classifiers. We only considered discrete attributes and removed instances with missing values from the datasets. As suggested by Friedman et al [14] we improved the performance of both TAN and BCkG classifiers by introducing an additional smoothing operation. This is particularly important in small datasets where the estimation of the conditional probabilities, given the parent attributes plus the class variable, is unreliable. NB classifiers are almost not affected by this problem since the data is partitioned according to the class variable and, usually, the class variables are adequately represented in the training set. The parameters of TAN and BCkG networks were smoothed using *Dirichlet priors* [17]. This amounts to adding 5 pseudo instances with conditional probabilities, given the parent attributes plus the class variable, distributed according to the frequency of the corresponding attribute in the training set.

Modeling Transcription Factor Binding Sites

We wanted to evaluate our method in the context of computational biology. There is a straightforward application of the BCkG model in the representation of transcription factor binding sites.

An important part of gene regulation is mediated by specific proteins, called the *transcription factors*, which influence the transcription of a particular gene by binding to specific sites on DNA sequences, called *transcription factor binding sites* (TFBS). Such binding sites are relatively short stretches of DNA, normally 5 to 25 nucleotides long. A commonly used representation of TFBS is a *position specific scoring matrices* (PSSM). This representation assumes independence of nucleotides in the binding sites, and so can be modeled by a Naive Bayes network. Some works appeared that argued in the direction of non-additivity in protein-DNA interactions [19] making a way for more complex models to appear which account for nucleotide interactions. Barash et al had already obtained good results modeling TFBS with trees and mixtures of trees [1]. Recently, Sharon and Segal also contributed in this direction [20]. Herein, we do preliminary evaluation of the extent to which the richer BC2G models are beneficial in representing TFBS.

The TRANSFAC database [13] contains hundreds of biologically validated TFBS. We extracted 14 data sets of aligned binding sites from the TRANSFAC database for which there were 20 or more sites. For each binding site we evaluated the ability of NB, TAN and BC2G to describe the distribution underlying the TFBS. We performed a *10 fold cross-validation* test in each data set and the results of the evaluation are presented in Figure 1.

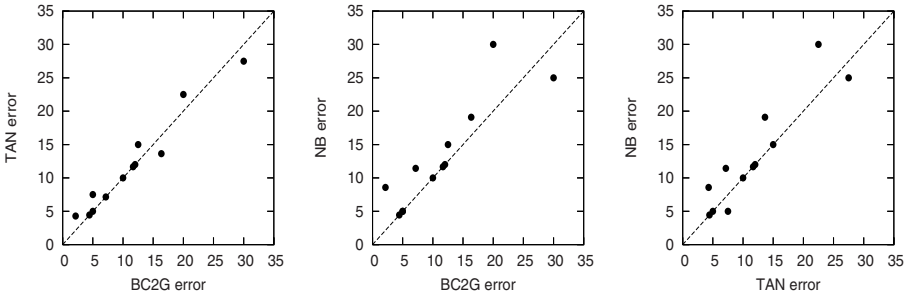


Fig. 1. Scatter plot comparing NB, TAN, and BC2G networks when modeling transcription factor binding sites. Points above the diagonal line corresponds to data sets on which the model in the x axis performs better than the model in the y axis.

5 Conclusions

This paper introduced a new heuristic to learn Bayesian network classifiers. The proposed heuristic consists of improving an optimal TAN classifier by adding important dependencies and removing irrelevant ones, guiding this process with

the total order induced by the BFS over the optimal TAN. The advantage of this restriction is twofold. First, the learning algorithm is polynomial. Second, as Friedmann et al observed [14], unrestricted learning does not necessarily outperforms TAN and NB due to overfitting,[1] and for this reason structure restriction helps avoiding this problem. The proposed heuristic is an improvement over the partial order based heuristic introduced in [2], adapted for classification. The proposed classifier scores always better than both TAN and NB classifiers. Moreover, experiments on modeling transcription factor binding sites show that, in many cases, the improved scores translate into increased classification accuracy.

Direction of future work include: instead of fixing an order for attributes at the same level in the BFS (c.f. Section 3 comment after Definition 1), (i) consider a random order over attributes at the same level or (ii) apply the TAN algorithm solely to attributes at the same level and order them with a BFS over the resulting TAN; combine and compare exhaustively our approach with other state-of-the-art Bayesian network learning methods; extending BCkG to deal with missing values and non discretized continuous variables; applying BCkG to a wider variety of datasets.

Acknowledgments

The first author thanks EU FEDER through FCT grant SFRH/BD/18660/2004. This work was partially supported by EU FEDER via FCT project POSC/EIA/57398/2004.

References

1. Barash, Y., Elidan, G., Friedman, N., Kaplan, T.: Modeling dependencies in protein-DNA binding sites. In: Proc. RECOMB 2003, pp. 28–37 (2003)
2. Carvalho, A.M., Oliveira, A.L., Sagot, M.-F.: Learning Bayesian networks consistent with the optimal branching. In: Proc. ICMLA 2007 (to appear, 2007)
3. Chickering, D.M.: Learning Bayesian networks is NP-Complete. In: Learning from data: AI and statistics V, pp. 121–130. Springer, Heidelberg (1996)
4. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Trans. Info. Theory 14(3), 462–467 (1968)
5. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artif. Intell. 42(2-3), 393–405 (1990)
6. Cooper, G.F., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9, 309–347 (1992)
7. Dagum, P., Luby, M.: Approximating probabilistic inference in Bayesian belief networks is NP-hard. Artif. Intell. 60(1), 141–153 (1993)
8. Dasgupta, S.: Learning polytrees. In: Proc. UAI 1999, pp. 134–141 (1999)

[1] Friedmann et al also noticed that the learning algorithm for Bayesian classifiers should maximize the conditional likelihood scoring function (CLL) instead of the likelihood. However, CLL is not decomposable, and therefore learning it seems to be intractable.

9. Domingos, P., Pazzani, M.J.: Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In: Proc. ICML1996, pp. 105–112 (1996)
10. Domingos, P., Pazzani, M.J.: Simple Bayesian classifiers do not assume independence. In: Proc. AAAI/IAAI 1996, vol. 2, p. 1386 (1996)
11. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons, New York (1973)
12. Edmonds, J.: Optimum branchings. J. Research of the National Bureau of Standards 71B, 233–240 (1967)
13. Wingender, E., et al.: The TRANSFAC system on gene expression regulation. Nuc. Ac. Res. 29(1), 281–283 (2001)
14. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29(2-3), 131–163 (1997)
15. Friedman, N., Koller, D.: Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. Machine Learning 50(1-2), 95–125 (2003)
16. Grossman, D., Domingos, P.: Learning Bayesian network classifiers by maximizing conditional likelihood. In: Proc. ICML 2004, pp. 46–53. ACM Press, New York (2004)
17. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20(3), 197–243 (1995)
18. Lam, W., Bacchus, F.: Learning Bayesian belief networks: An approach based on the MDL principle. Comp. Intell. 10, 269–294 (1994)
19. O'Flanagan, R.A., Paillard, G., Lavery, R., Sengupta, A.M.: Non-additivity in protein-DNA binding. Bioinformatics 21(10), 2254–2263 (2005)
20. Sharon, E., Segal, E.: A feature-based approach to modeling protein-DNA interactions. In: Proc. RECOMB 2007, LNCS(LNBI), vol. 4453, pp. 77–91. Springer, Heidelberg (2007)
21. Su, J., Zhang, H.: Full Bayesian network classifiers. In: Proc. ICML 2006, pp. 897–904. ACM Press, New York (2006)
22. Suzuki, J.: A construction of Bayesian networks from databases based on an MDL principle. In: Proc. UAI 1993, pp. 266–273 (1993)
23. Teyssier, M., Koller, D.: Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In: Proc. UAI 2005, pp. 584–591 (2005)