Local Score Computation in Learning Belief Networks

Y. Xiang and J. Lee

Dept. of Computing and Information Science
University of Guelph, Guelph, Ontario, Canada N1G 2W1
{yxiang, lee}@snowhite.cis.uoguelph.ca
http://snowhite.cis.uoguelph.ca/faculty_info/yxiang/

Abstract. We propose an improved scoring metrics for learning belief networks driven by issues arising from learning in pseudo-independent domains. We identify a small subset of variables called a *crux*, which is sufficient to compute the incremental improvement of alternative belief network structures. We prove formally that such local computation, while improving efficiency, does not introduce any error to the evaluation of alternative structures.

(Keywords: Knowledge discovery, data mining, machine learning, belief networks, uncertain reasoning.)

1 Introduction

Learning belief networks from data has been an active research area in recent years [2, 7, 4, 15, 3]. Successive graphical structures are evaluated with a scoring metrics until a stopping condition is met. As the task is NP-hard [1], a common method in selection the structure is the single-link lookahead, where successive structures adopted differ by a single link. It has been shown that a class of probabilistic models called *pseudo-independent* (PI) models cannot be learned by single-link search [14]. A more sophisticated method (multi-link lookahead) is proposed in [15] and is improved in [5] for learning decomposable Markov networks (DMNs) from data.

DMNs are less expressive than Bayesian networks (BNs). However, DMNs are the runtime representation of several algorithms for inference with BNs [8, 6, 10], and can be the intermediate results for learning BNs. For example, learning PI models needs multi-link lookahead and the search space for DAGs is much larger than that of chordal graphs. Learning DMNs first can then restrict the search for DAGs to a much smaller space, improving the efficiency.

In this work, we focus on learning DMNs using the entropy score which is closely related to other scoring metrics [15] such as Bayesian [2], minimum description length (MDL) [7], and conditional independence [9, 11]. The score of a DMN is defined as the entropy of the DMN computed from its joint probability distribution (jpd). Previous work [15, 5, 13] used entropy score as the sole control of both goodness-of-fit and complexity of the output structure. An increment threshold Δh of the entropy score is set by the user. Learning stops when no structure (allowed by the given lookahead search) can improve the score beyond Δh. The smaller the value of Δh, the better the goodness-of-fit of the output structure, and the more complex the structure is.

Such stopping control works fine with the single-link lookahead. However, an issue arises when multi-link lookahead is performed: It is possible at some point of learning that a best single link may produce a score improvement 0.0099 and get rejected since $\Delta h = 0.01$. On the other hand, a best double-link that produces a score improvement of 0.01 will be adopted. It can be argued that the double-link increases the complexity of the structure much more than it contributes to the goodness-of-fit. Hence if any link is to be added at all, a single link is a better choice than a double-link. However, using the entropy improvement as the sole stopping control, this issue cannot be resolved.

In this work, we address this issue by explicitly describing the model complexity in the score (a common approach in learning). We define a new score as

$$\Gamma(M) = \Gamma_1(M) + \alpha\ \Gamma_2(M)\ ,$$

where M is a DMN, $\Gamma_1(M)$ measures the goodness-of-fit of M, and $\Gamma_2(M)$ measures the complexity of M. The constant α is set by the user to trade goodness-of-fit with the complexity of the output DMN. Learning stops when no DMN M' can improve $\Gamma(M)$ for the current DMN M. Hence, threshold is no longer needed. The above issue will be resolved since the single link will improve $\Gamma(M)$ more than the double link.

In the rest of the paper, we propose how to compute the incremental change in $\Gamma(M)$ due to link addition by local computation using a small subset of variables called *crux*. We prove the correctness of the algorithms formally.

2 Background

Let $G = (V, E)$ be a graph, where V is a set of nodes and E a set of links. A graph is a *forest* if there are no more than one path between each pair of nodes. A forest is a *tree* if it is connected. A set X of nodes is *complete* if elements of X are pairwise adjacent. A maximal set of nodes that is complete is a *clique*. A path or cycle ρ has a *chord* if there is a link between two non-adjacent nodes in ρ. G is *chordal* if every cycle of length ≤ 4 has a chord.

A *cluster graph* is a triplet (V, Ω, S), where V is called a *generating* set, Ω is a set of nodes each of which is labeled by a nonempty subset of V and is called a *cluster*, S is a set of links each of which is labeled by the intersection of the two clusters connected and is called a *separator*. A cluster forest is a *junction forest* (JF) if the intersection of every pair of connected clusters is contained in every cluster on the path between them. Let $G = (V, E)$ be a chordal graph, Ω be the set of cliques of G, and F be a JF (V, Ω, S). We will call F a *corresponding* JF of G. Such a JF exists if and only if G is chordal.

A DMN is a triplet $M = (V, G, \mathcal{P})$, where V is a set of discrete variables in a problem domain, and $G = (V, E)$ is a chordal graph. $\mathcal{P}$ is a set of probability distributions one for each cluster defined as follows: Let F be a corresponding

JF of G. Direct links of F such that each cluster has no more than one parent cluster. For each cluster C with a parent Q, associate C with $P(C|Q)$. The jpd of M is defined as $P(V) = \prod_C P(C|Q)$. Probabilistic conditional independence among variables in V is conveyed by node separation in G, and by separator separation in F. It has been shown [12] that G and F encode exactly the same dependence relations within V. Hence, we will switch between the two graphical views from time to time.

3 Local computation for measure of goodness-of-fit

The goodness-of-fit of a DMN M to an underlying (unknown) domain model can be measured by the K-L cross entropy between them. It has been shown [15] that to minimize the K-L cross entropy, it suffices to minimize the entropy of M which can be computed as

$$H_M(V) = \sum_C H(C) - \sum_S H(S) , \tag{1}$$

where C is a cluster in the corresponding JF and S is a separator. Hence we shall use the entropy of a DMN M as the measure of goodness-of-fit, denoted as $\Gamma_1(M) = H_M(V)$.

During learning, a large number of alternative DMN structures need to be evaluated using the score. Since most of the clusters and separators do not change between successive structures, it is inefficient to compute the entropy of all of them for each structure. It is much more efficient to identify a small set of clusters and separators that contribute to the incremental change of the score after a set of links has been added to the current structure. In the following, we study how these clusters and separators can be identified effectively.

First, we define the context in which the learning takes place: At each step of learning, a set of links L is added to the current structure G to obtain a supergraph G' of G. The cardinality $|L|$ depends on whether it is single-link lookahead ($|L| = 1$) or multi-link lookahead ($|L| > 1$). The initial G at the start of learning is an empty (chordal) graph. We require that at each step, G' is also a chordal graph and the endpoints ED of L are contained in a clique of G'. We shall call G' the *chordal supergraph* of G induced by L. We denote the corresponding JF of G' by F'.

4 The notion of crux

In this section, we identify a small subset of V called *crux* that are defined by the structural change due to adding links L to a chordal graph. We establish some properties of crux. In the next section, we show that the crux is a sufficient subset of variables necessary to compute the incremental change of of entropy.

Lemma 1 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Then the clique that contains ED, the set of endpoints of L, is unique.*

Proof:

Suppose that two distinct cliques C and Q exist in G' that contain ED. Then there exist $c \in C$ and $q \in Q$ such that $c \notin Q$ and $q \notin C$. That is, $\{c, q\}$ is not a link in G' and hence not in G as well.

Let $\{x, y\}$ be any link in L. Since x, y and c are all in C, they must be complete in G'. Since $\{x, c\}$ and $\{y, c\}$ are not in L, they must be links in G. Similarly, $\{x, q\}$ and $\{y, q\}$ must be links in G. We have therefore found a cycle (x, c, y, q, x) in G and neither $\{x, y\}$ nor $\{c, q\}$ is a link in G: a chordless cycle. This contradicts that G is chordal. $\square$

Definition 2 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let C be the unique clique in G' that contains the endpoints of L. Let Q be any clique of G' such that $Q \cap C \neq \emptyset$ and Q is not a clique in G. Denote the set of all such cliques by Φ. Then the union of elements in Φ, namely, $\bigcup_{Q \in \Phi} Q$ is called the* `crux` *induced by G and L, and the set Φ is called the* `generating set` *of the crux.*

Note that the crux contains C. Note also that since each pair of cliques in a chordal graph is incomparable, given the crux R, its generating set Φ can be uniquely identified.

Figure 1 illustrates the concept of crux in different cases. In each box, the upper graphs are chordal graphs G and G' where dashed link(s) indicate the set L of links added. The lower graphs in each box depict the corresponding JFs where the dashed cluster(s) form the generating set Φ. For example, in (a) and (b), the generating set Φ contains only a single cluster which is the crux itself. In (c), however, Φ consists of $\{b, c, f\}$ and $\{c, e, f\}$ while the crux is $\{b, c, e, f\}$.

The following proposition says that each clique in Φ contains the endpoints of at least one link in L, and Φ is made of all such cliques.

Proposition 3 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let R be the crux induced by G and L and Φ be its generating set.*

1. *For each $Q \in \Phi$, there exists a link $\{x, y\} \in L$ such that $\{x, y\} \subset Q$.*
2. *For each clique Q in G', if there exists a link $\{x, y\} \in L$ such that $\{x, y\} \subset Q$, then $Q \in \Phi$.*

Proof:

(1) Suppose for $Q \in \Phi$, no such $\{x, y\}$ is contained in Q. Then Q is not a clique newly created or enlarged by the addition of L to G. That is, G is a clique in G: contradiction to $Q \in \Phi$.

(2) Let Q be a clique in G' such that the stated condition holds. The $Q \cap C \neq \emptyset$ and Q is not a clique in G. $\square$

The following proposition shows that the crux is in fact the union of all cliques newly formulated due to the addition of L. In the proposition, "\" is the set difference operator.

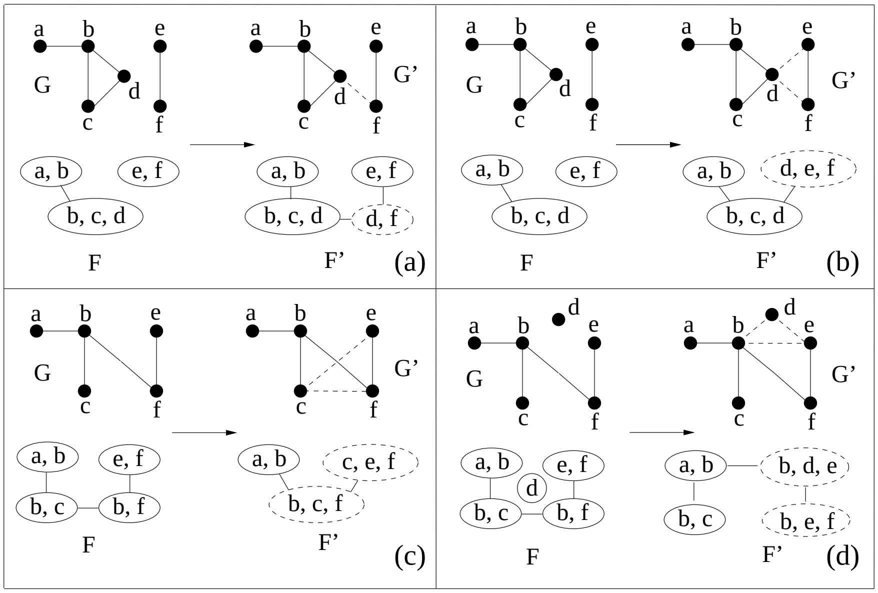

Fig. 1. Illustration of crux

Proposition 4 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let Ω be the set of cliques in G, Ω' be the set of cliques in G', R be the crux induced by G and L, and Φ be the generating set of R. Then $\Phi = \Omega' \setminus \Omega$.*

Proof:

Each clique contained in R is in $\Omega' \setminus \Omega$ by the definition of crux. We only need to show that each $Q \in \Omega' \setminus \Omega$ is also contained in R, that is $Q \cap C \neq \emptyset$, where C is the unique clique in G' that contains endpoints ED of L. Each clique Q in G' that is created or is modified from cliques of G due to adding L must contain elements of ED, and hence $Q \cap C \neq \emptyset$. □

Although Proposition 4 gives a much simplier definition of crux, Definition 2 allows more efficient computation of the crux. Based on Proposition 4, the crux can be obtained by computing $\Omega' \setminus \Omega$. The complexity is $O(|\Omega|^2)$ since $|\Omega'| \approx |\Omega|$. On the other hand, based on Definition 2, one pass through Ω' is needed to find C, another pass is needed to find cliques intersecting with C (assuming k such cliques are found), and additional k passes through Ω are needed to identify the newly created or enlarged cliques. The complexity is $O((k+2)\ |\Omega|)$. The value of k is usually a very small integer. Hence for large problem domains, $k+2$ is much smaller than $|\Omega|$. Since the crux needs be obtained for every structure to be evaluated, significant computational savings can be obtained if Definition 2 is followed.

The following proposition says that the generating set of the crux forms a subtree in F'.

Proposition 5 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let R be the crux induced by G and L and F' be the corresponding JF of G'. The generating set Φ of R forms a connected subtree in F'.*

Proof:

We prove by contradiction. Let C be the unique clique of G that contains endpoints ED of L. Suppose that members of Φ do not form a connected subtree in F'. Then there exists a cluster $Q \in \Phi$ and a cluster $Z \notin \Phi$ such that Z is on the path between C and Q in F'. This implies $Z \supset C \cap Q$. By Proposition 3, there exists $\{x, y\} \in L$ such that $\{x, y\} \subset Q$. By Lemma 1, we also have $\{x, y\} \subset C$. Therefore, we have $\{x, y\} \in Z$. By Proposition 3, this implies that $Z \in \Phi$: a contradiction. □

5 Sufficient subdomain for entropy score computation

The following proposition shows that if the two corresponding junction forests F and F' share some clusters, then there exists one such cluster that is terminal in F'.

Proposition 6 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let F and F' be the corresponding JF of G and G', respectively. If F' shares clusters with F, then at least one of them is terminal in F'.*

Proof:

Suppose the conclusion does not hold. Let R be the crux induced by G and L. Then the generating set of R will not form a connected subtree in F': a contradiction with Proposition 5. □

The following proposition says that if a cluster shared by F and F' is terminal in F', then its boundary is complete and identical in both chordal graphs, G and G'.

Proposition 7 *Let G be a chordal graph and G' be a chordal supergraph of G induced by a set L of links. Let F and F' be the corresponding JF of G and G', respectively. Let Q be a cluster shared by F and F', and is terminal in F'. Then the boundary between Q and $V \setminus Q$ in both G and G' is complete and identical.*

Proof:

Since Q is terminal in F', its boundary with $V \setminus Q$ in G' is complete. Since Q is shared by F and F', it does not contain any $\{x, y\} \in L$ by Propositions 3 and 4. Hence, its boundary with $V \setminus Q$ in G was not altered by adding L to G. This implies that its boundary with $V \setminus Q$ in G is identical to that in G'. □

The following proposition shows that the increment of entropy score can be correctly computed without variables in a shared terminal cluster. Let G be the structure of a DMN M over V. Let Q be a clique in G with a complete boundary S. If we remove variables $Q \setminus S$ from V and remove the corresponding nodes from G, the resultant graph is still chordal and is a valid structure of a DMN. We call the resultant DMN a *reduced* DMN.

Proposition 8 *Let G be the structure of a DMN M over V and G' be the structure of another DMN M' that is a chordal supergraph of G induced by a set L of links. Let $H(V)$ and $H'(V)$ be the entropy of M and M', respectively. Let Q be a cluster shared by F and F', and is terminal in F'. Let S be the separator of Q in F'. Then $\delta h = H(V) - H'(V)$ can be computed using reduced DMNs where variables in $V \setminus (Q \setminus S)$ are removed.*

Proof:

Denote $V^* = V \setminus (Q \setminus S)$. Since S is the boundary of Q in G', we have

$$H'(V) = H'(V^*) + H(Q) - H(S) \ ,$$

where $H'(V^*)$ is the entropy of the DMN obtained by removing variables $Q \setminus S$ from M'. By Proposition 7, S is also the boundary of Q in G. We have

$$H(V) = H(V^*) + H(Q) - H(S) \ ,$$

where $H(V^*)$ is the entropy of the DMN obtained by removing variables $Q \setminus S$ from M. Hence $\delta h = H(V) - H'(V) = H(V^*) - H'(V^*)$. □

By recursively applying Proposition 8, the following theorem establish the correctness of local computation for the incremental entropy score.

Theorem 9 *Let G be the structure of a DMN M over V and G' be the structure of another DMN M' that is a chordal supergraph of G induced by a set L of links. Let $H(V)$ and $H'(V)$ be the entropy of M and M', respectively. Then the crux R induced by G and L is a sufficient subset of V needed to compute $\delta h = H(V) - H'(V)$.*

Proof:

Let F be the corresponding JF of G, and F' be that of G'. If F and F' have shared clusters, by Proposition 6 a terminal cluster Q shared by F and F' can be found. Denote the separator of Q in F' by S and $V^* = V \setminus (Q \setminus S)$. By Proposition 8, δh can be computed as $H(V^*) - H'(V^*)$. By recursively applying Propositions 6 and 8, eventually we can remove all clusters shared by F and

F'. The remaining clusters is the generating set Φ of R, and hence δh can be computed as $\delta h = H(R) - H'(R)$. $\square$

Theorem 9 suggests the following method to compute δh by local computation: First compute the crux R based on Definition 2. Then compute the subgraphs of G and G' spanned by R. Convert the subgraphs into junction forest representations and compute δh using equation 1.

6 Complexity of a decomposable Markov network

We now shift to the computation of the complexity of a DMN, which we define as the total number of unconstrained parameters needed to specify $\mathcal{P}$. We denote the space of a set X of variables by D_X. The following Lemma derives the complexity of two adjacent cluster representations in a DMN. Due to space limit, the proofs for all formal results on the complexity will be included in a longer version of this paper.

Lemma 10 *Let C be a cluster in the junction forest representation of a DMN, Q be its terminal parent, and S be their separator. Then the total number of unconstrained parameters required to specify $P(C \cup Q)$ is $|D_C| + |D_Q| - |D_S| - 1$.*

The following theorem derives the complexity of a DMN whose structure is a JT.

Theorem 11 *Let Ω be the set of clusters in the junction tree representation of a DMN over variables V and Ψ be the set of separators. Then the total number of unconstrained parameters needed to specify $P(V)$ is*

$$N = \sum_{C_i \in \Omega} |D_{C_i}| - \sum_{S_j \in \Psi} |D_{S_j}| - 1 \ .$$

The following corollary extends Theorem 11 on the complexity of a JT representation to a junction forest representation.

Corollary 12 *Let Ω be the set of clusters in a junction forest representation of a DMN over V and Ψ be the set of separators. Let the junction forest consist of k junction trees. Then the total number of unconstrained parameters needed to specify $P(V)$ is*

$$N = \sum_{C_i \in \Omega} |D_{C_i}| - \sum_{S_j \in \Psi} |D_{S_j}| - k \ .$$

Based on Corollary 12, we have the measure of complexity of a DMN M as

$$\Gamma_2(M) = \sum_{C_i \in \Omega} |D_{C_i}| - \sum_{S_j \in \Psi} |D_{S_j}| - k \ .$$

7 Local computation of DMN complexity

Following the same idea of local computation of δh, we want to find a small subset of variables sufficient to compute the incremental change of complexity due to the addition of links L to the current DMN. We show below that the crux is just such a subset.

The following proposition says that a terminal cluster unchanged by the addition of L is irrelevant to the computation of the incremental complexity.

Proposition 13 *Let G be the structure of a DMN M over V and G' be the structure of another DMN M' that is a chordal supergraph of G induced by a set L of links. Let N and N' be the total number of unconstrained parameters needed to specify $P(V)$ for M and $P'(V)$ for M', respectively. Let Q be a cluster shared by F and F', and is terminal in F'. Let S be the separator of Q in F'. Then $\delta n = N' - N$ can be computed using reduced DMNs where variables in $V \setminus (Q \setminus S)$ are removed.*

The following theorem shows that the crux is sufficient for computing the incremental complexity.

Theorem 14 *Let G be the structure of a DMN M over V and G' be the structure of another DMN M' that is a chordal supergraph of G induced by a set L of links. Let N and N' be the total number of unconstrained parameters needed to specify $P(V)$ for M and $P'(V)$ for M', respectively. Then the crux R induced by G and L is a sufficient subset of V needed to compute $\delta n = N' - N$.*

Theorem 14 suggests the following method to obtain the incremental change to the DMN complexity by local computation: First compute the crux R based on Definition 2. Then compute the subgraphs of G and G' spanned by R. Convert the subgraphs into junction forest representations and compute δn using Corollary 12.

8 Conclusion

We have shown that crux forms a subset of variables sufficient to compute the incremental change of both goodness-of-fit and complexity of a DMN during search of alternative dependence structures. The overall incremental improvement due to adding links L is $\delta \Gamma = \delta h - \alpha\ \delta n$, computed using the crux. Search can terminate when no alternative structures provide positive $\delta \Gamma$. The computation is much more efficient than direct evaluation as the crux is small and computation is local. There is no loss of accuracy due to the local computation. The method is currently being implemented in WEBWEAVR-III toolkit.

Acknowledgements

This work is supported by Research Grant OGP0155425 from NSERC of Canada and a grant from IRIS in the Networks of Centres of Excellence Program of Canada.

References

1. D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: serach methods and experimental results. In *Proc. of 5th Conf. on Artificial Intelligence and Statistics*, pages 112–128, Ft. Lauderdale, 1995. Society for AI and Statistics.
2. G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, (9):309–347, 1992.
3. N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In G.F. Cooper and S. Moral, editors, *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, pages 139–147, Madison, Wisconsin, 1998. Morgan Kaufmann.
4. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
5. J. Hu and Y. Xiang. Learning belief networks in domains with recursively embedded pseudo independent submodels. In *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, pages 258–265, Providence, 1997.
6. F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, (4):269–282, 1990.
7. W. Lam and F. Bacchus. Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.
8. S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *J. Royal Statistical Society, Series B*, (50):157–244, 1988.
9. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
10. G. Shafer. *Probabilistic Expert Systems*. Society for Industrial and Applied Mathematics, Philadelphia, 1996.
11. P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–73, 1991.
12. Y. Xiang. A characterization of single-link search in learning belief networks. In P. Compton H. Motoda, R. Mizoguchi and H. Liu, editors, *Proc. Pacific Rim Knowledge Acquisition Workshop*, pages 218–233, Singapore, 1998.
13. Y. Xiang, J. Hu, N. Cercone, and H. Hamilton. Learning pseudo-independent models: analytical and experimental results. In H. Hamilton, editor, *Advances in Artificial Intelligence*, pages 227–239. Springer, 2000.
14. Y. Xiang, S.K.M. Wong, and N. Cercone. Critical remarks on single link search in learning belief networks. In *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*, pages 564–571, Portland, 1996.
15. Y. Xiang, S.K.M. Wong, and N. Cercone. A 'microscopic' study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26(1):65–92, 1997.