A Microscopic Study of Minimum Entropy Search in Learning Decomposable Markov Networks

Y. Xiang, S.K.M Wong and N. Cercone
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: yxiang, wong, nick@cs.uregina.ca
Tel: (306) 585-4088, Fax: (306) 585-4745

Abstract

Several scoring metrics have been used in conjunction with search procedures in algorithms for learning probabilistic networks from data. In this paper, we study the properties of entropy as a scoring metric in learning a decomposable Markov network. Though entropy and related scoring metrics have commonly been used, its 'microscopic' property and its asymptotic behavior when used in a search is unknown. We provide such a microscopic study of a minimum entropy search algorithm, and show that the algorithm learns an I-map of the domain model as the size of sample data approaches infinity.

Search procedures that modify a network structure one link at a time have been used by several learning algorithms because of their efficiency. In these procedures, a single link, representing a dependence relation between a pair of variables, may be added or deleted after each evaluation of a set of alternative links according to the scoring metric. However, the single-link lookahead should be used with caution. The microscopic study indicates that a class of probabilistic domain models cannot be learned by such procedures. Our results strongly suggest that prior knowledge about the problem domain together with a multi-link search strategy would provide an effective way to uncover many domain models.

Learning by heuristic search may generate superfluous links that are unnecessary to encode domain dependencies. False dependencies inferred from a finite database may cause the generation of additional superfluous links. We show that the entropy metric has built-in resistance to some superfluous links which can be further reduced by performing a conditional independence test in a multi-link lookahead search.

Keywords: Inductive learning, reasoning under uncertainty, knowledge acquisition, Markov networks, probabilistic networks.

1 Introduction

A probabilistic network [28, 26, 14, 4] combines a *qualitative* graphic structure, which encodes domain dependencies, with a *quantitative* probability distribution, which encodes the strength of the dependencies. The network structure can be a directed or undirected graph. A Bayesian network (BN) structure is a directed acyclic graph and a decomposable Markov network (DMN) structure is an undirected chordal graph. As many effective probabilistic inference techniques have been developed [27, 17, 23, 20, 36] and the applicability of probabilistic networks have been amply demonstrated in many artificial intelligence domains [4], many researchers turn their attention to automatic learning of such networks from data.

Chow and Liu [7] pioneered learning of probabilistic networks. They developed an algorithm to approximate a joint probability distribution (jpd) by a tree-structured BN. Rebane and Pearl [29] extended their method to learn a polytree-structured BN. However, many real world domain models cannot be represented adequately with a tree-structured network. The following algorithms are all applicable to learning a multiply connected network. Herskovits and Cooper [18] developed the Kutato algorithm to learn a BN from a database of cases by minimizing the entropy of the distribution defined by the BN. Their method starts with an empty graph (no links) and adds one link at each pass during search. Later, they proposed the K2 algorithm [8] that learns a BN based on a Bayesian method which selects a BN with the highest posterior probability given a database. A similar algorithm was independently developed by Buntine [2]. Recently, Heckerman et al [16] applied the Bayesian method to learning a BN by combining prior knowledge and statistical data. Spirtes and Glymour [31] developed the PC algorithm that learns a BN by deleting links from a complete graph. Lam and Bacchus [22] applied the minimal description length (MDL) principle to learning a BN, which evaluates a BN as the best if it has the minimal sum of its own encoding length and the encoding length of the data given the BN. Instead of learning a BN, Fung and Crawford [11] developed the Constructor algorithm that learns a DMN. A more extensive review of literature for learning probabilistic networks can be found in [18, 8, 3, 15].

In this paper we consider learning a DMN from a database. Pearl [28] showed that directionality makes BNs a richer language in expressing dependencies. For instance, an induced dependency can be expressed by a BN but not by a DMN. Most of the work reviewed above, except that by Fung and Crawford, learn BNs from data. However, the usefulness of learning a DMN can be seen as follows.

One important application of BNs is to compute posterior marginal probabilities. An elegant algorithm for doing that with a multiply connected network is proposed by Jensen et al [20]. The method uses a DMN, in terms of its junction tree (JT), as the run time representation of a BN. In converting the original BN into a DMN and then into a JT, directionality is discarded. Therefore, as long as computing posterior marginal probabilities is concerned, a DMN is as equally expressive as a BN. Jensen et al's method can be extended to probabilistic inference with multiply sectioned Bayesian networks in a single agent oriented system [36, 35] as well as in a multiagent distributed interpretation system [34]. The run time representation is a set of DMNs (in terms of a set of JTs). It has been shown [33] that computation of posterior marginal probabilities of a BN can be performed using an extended relational database once the BN is converted into its equivalent DMN. This implies that once a probabilistic model is expressed in terms of a DMN, inference can be performed using standard relational DBMSs. Finally, as BNs and DMNs are so closely related, knowledge gained in learning one of them will benefit the learning of the other.

It has been shown that learning probabilistic networks is NP-hard [1, 6]. Therefore, it is justified to use heuristic search in learning. Many algorithms developed use a scoring metric and a search procedure. The scoring metric evaluates the goodness-of-fit of a structure to the data, and the search procedure generates alternative structures and selects the best based on the evaluation.

Out of many possible scoring metrics, the Bayesian metric, the description length metric and the entropy metric have been used and studied by several researchers [18, 2, 8, 22, 16, 1, 32]. Cheeseman [5] showed that the Bayesian metric and the description length metric are equivalent subject to a constant difference. Lam and Bacchus [22] showed that in applying MDL principle to learning a BN, the encoding length of the data is a monotonically increasing function of the Kullback-Leibler cross entropy between the distribution defined by the BN and the true distribution. It has also been shown [32] that the cross entropy of a DMN can be expressed as the difference between the entropy of the distribution defined by the DMN and the entropy of the true distribution which is a constant given a static domain. Entropy has also been used as a means to test conditional independence in learning BNs [29]. Therefore, the maximization of the posterior of a network given a database [8, 16], the minimization of description length [22], the minimization of cross entropy between a network and the true model [22], the minimization of entropy of a network [18, 32], and conditional independence tests are all closely related. A better understanding of any of them will lead to a better understanding of all of them.

In all the methods mentioned above, a heuristic method with a single-link lookahead search is adopted in order to avoid the exponential complexity of exhaustive comparison of all possible networks. However, as far as we know, the interplay of the scoring metric and the search process has not been analyzed. Many questions have not been answered. For example, how does the current score determine the next link (dependency) that will be selected? How does the inclusion of a new link change the score and why? Is it possible that once a superfluous link is added, the search may continue until a complete graph structure is generated? We have already had a good 'macroscopic' perspective about which network(s) should be chosen if an exhaustive comparison is possible according to a particular scoring metric. However, in viewing the search process as a chain that connects the initial network to some learned network, we do not seem to have a satisfactory ‘microscopic’ understanding about what is occurring during the transition from one link to the next on the chain. We do not seem to know how good or how bad the learned network is relative to the global optimal. As pointed out by Spirtes and Glymour [31] and acknowledged by Cooper and Herskovits [8], the “asymptotic reliability of the procedure is unknown.”

In this paper we provide such a microscopic study under the context of learning a DMN from a database by using an entropy scoring metric and a minimum entropy search procedure. The microscopic understanding leads to the identification of drawbacks of a single-link lookahead search, which is commonly used in learning probabilistic networks.

We show that a class of probabilistic domain models cannot be learned by a single-link lookahead search procedure. Although our observation is based on the entropy scoring metric, because of the close relationship between the entroy metric and other metrics described above, the results we obtain are valid for other algorithms as well. We will show that some domain models cannot be learned by the standard methods [18, 31, 22]. We therefore propose a multi-link lookahead learning algorithm. We will analyze the computational complexity of this algorithm and suggest solutions to alleviate the problem.

This microscopic study also establishes the ‘asymptotic’ behavior of the minimum entropy search algorithm. We will show that, when the number of cases in a database becomes very large, the algorithm will halt and return an I-map of the domain model.

In practice, learning is performed on a database of a finite size. A finite database may contain *false* dependencies that do not exist among the domain variables. They cause the learning algorithms to generate superfluous links. These links and their associated numerical probability values tend to encode ’noise’ and bias the jpd of the learned networks. Even though the database is very large and contains no false dependencies, learning using heuristic search may generate superfluous links that do not reflect the true domain dependencies. These superfluous links tend to make the inference using the resultant network unnecessarily more complex. The Bayesian metric and the description length metric have *automatic* (versus user controlled) mechanism to balance the complexity of the network and the fitness of data. The entropy metric is equivalent only to the encoding length of the data given the learned network as mentioned above. Classifying superfluous links generated from different cases reveals the built-in resistance of the entropy metric to some superfluous links. To further balance complexity and fitness, we use the standard χ^2 test for independence. We discuss the problem involved in performing such a test in a multi-link lookahead search.

Section 2 provides the background and terminology. We present in Section 3 the rational of the minimum entropy approach. In Section 4, we study the microscopic mechanism of the minimum entropy search in learning a decomposable Markov network as an I-map of a domain model. We will also discuss the built-in resistance of the entropy metric to superfluous links. In Section 5, we demonstrate the limitation of a single-link lookahead search. We present in Section 6 a multi-link lookahead algorithm based on the minimum entropy search. In Section 7, we discuss how to use the χ^2 test in a multi-link search. Experimental results are presented in Section 8, followed by a concluding discussion.

2 Background and Terminology

2.1 Graph related terminology

A *chord* in an undirected graph is a link that connects two nonadjacent nodes. A graph is *chordal* if every cycle of length > 3 has a chord. The undirected graph G_1 in Figure 1 is *not* chordal since the cycle a, (a, b), b, (b, d), d, (d, c), c, (c, a), a of length 4 has a pair of nonadjacent nodes b and c that are unconnected. If we add the chord (b, c) to G_1, it becomes G_2 which is chordal. A *clique* of a graph is a maximal set of nodes pairwise linked. G_2 has four cliques $\{a, b, c\}$, $\{b, c, d\}$,$\{c, e\}$ and $\{c, f\}$. A *component* of a graph is a maximal subgraph that is connected. In Figure 1, G_2 has a single component and G_3 has two components.

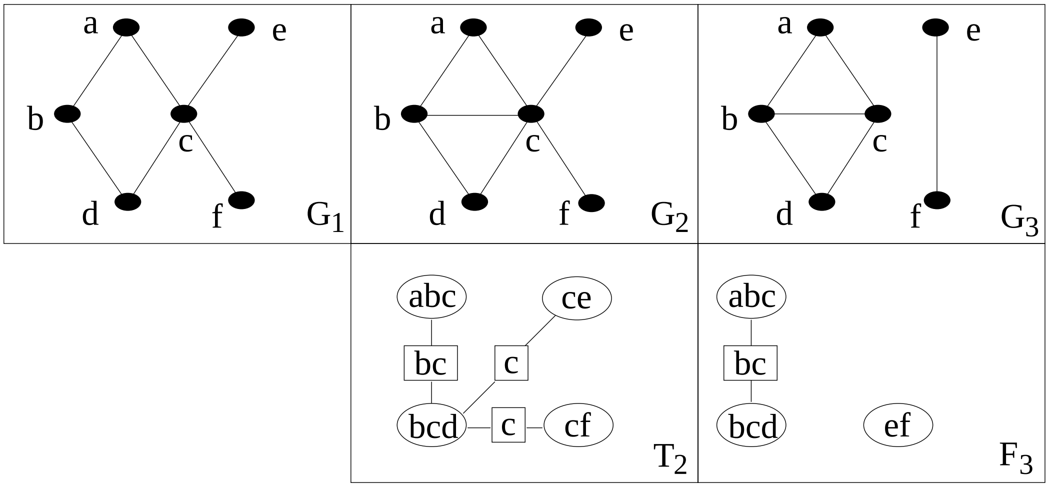

Figure 1: G_1: a non-chordal graph. G_2: a chordal graph with a single component. G_3: a chordal graph with two components. T_2: a junction tree of G_2, where nodes are drawn as ovals and sepsets are drawn as boxes. F_3: a junction forest of G_3.

Let G be a connected chordal graph. A *junction tree* (JT) T of G is a tree whose nodes are labeled by cliques of G such that for each pair of nodes of T, their intersection is contained in every node on the unique path between them. In Figure 1, T_2 is a JT of G_2. Without confusion, we sometimes refer to a *node* C in T as a *clique* when the nodes of G contained in C are of interest. For instance, we may say that T_2 has a clique $\{a, b, c\}$. The intersection of two adjacent cliques in T is called the *sepset* of the two cliques. In general, G may not be connected. A *junction forest* (JF) F of G is a set of JTs each of which is a JT of one component of G. In Figure 1, F_3 is a JF of G_3 and F_3 consists of two JTs. T_2 is a (trivial) JF of G_2.

Let X, Y and Z be three subsets of nodes in a graph. We use $< X|Z|Y >$ to mean that nodes in Z intercept all paths between nodes of X and nodes of Y. In G_2 of Figure 1, we have $< \{a\}|\{b,c\}|\{d\} >$. In a JT, we use $< X|Z|Y >$ to mean that Z is a sepset on the unique path between the clique that contains X and the clique that contains Y. For example, the statements $< \{a\}|\{b,c\}|\{d\} >$, $< \{b,d\}|\{c\}|\{f\} >$ and $< \{a,b\}|\{c\}|\{e\} >$ are all true in T_2.

2.2 Dependency graphs

Let N be a set of discrete variables in a problem domain and $X \subseteq N$. A *configuration* $\overline{x}$ of X is an assignment of values to every variable in X. A *probabilistic model* (PM) over N is an encoding of probabilistic information that determines the probability of every configuration of X for every $X \subseteq N$. A PM over N can be specified by a jpd over N. The entropy of X defined by a probability distribution P over X is $H(X) = -\sum_{\overline{x}} P(\overline{x}) \log(P(\overline{x}))$.

We will denote a PM by $\mathcal{M}$. Our task is to learn a probabilistic network from the data generated by $\mathcal{M}$. In practice, we usually have less data than what is necessary to reliably estimate the jpd over N. However, we may be able to estimate reliably the marginal distribution over $X \subset N$ if $|X|$ is small. Therefore, the jpd over N is mainly used in this paper as a conceptual entity.

Let X, Y and Z be three subsets of N. X and Y are *conditionally independent* given Z, denoted $Ind(X,Z,Y)$, iff $P(\overline{x}|\overline{yz}) = P(\overline{x}|\overline{z})$ whenever $P(\overline{yz}) > 0$.

Since we use graphs to represent independency relations among variables, we will use *nodes* and *variables* interchangeably. An undirected graph G is an *independency map (I-map)* of $\mathcal{M}$ over N if there is a one-to-one correspondence between nodes of G and variables in N such that for all disjoint subsets X, Y and Z of N, we have $< X|Z|Y > \Rightarrow Ind(X,Z,Y)$. That is, in an I-map, variables that are graphically separated are independent. However, there is no guarantee that variables graphically connected are necessarily dependent. For a detailed treatment of graphical representation of dependency models, see Pearl [28].

Let $G = (N,E)$ be a chordal graph, F be a JF of G, and $\mathcal{M}$ be a PM over N. Let C_i be a clique of F and S_j be a sepset of F. Let $P_{\mathcal{M}}(C_i)$ and $P_{\mathcal{M}}(S_j)$ be the marginal distributions over C_i and S_j, respectively, defined by $\mathcal{M}$. The jpd $P = (\prod_i P_{\mathcal{M}}(C_i))/(\prod_j P_{\mathcal{M}}(S_j))$ is called the *projected distribution of $\mathcal{M}$ on G (or on F)*. Note that we have written $P(N)$ as P for simplicity. The pair (G,P) is a *decomposable Markov network* (DMN) of $\mathcal{M}$, where G is the *structure* of the DMN and P is the distribution of the DMN[1]. In practice,

[1] What we call a *decomposable Markov network* has been termed differently in the literature. It is called simply *Markov network* in [11, 32] and *Markov graph* in [8]. The term *decomposable Markov network* is implicitly used in [28] to mean the similar thing as defined

$P_{\mathcal{M}}(C_i)$ is estimated from the data generated by $\mathcal{M}$. Note that (G, P) defines a PM which may or may not be equivalent to $\mathcal{M}$. The entropy of N defined by P is equal to $H(N) = \sum_i H(C_i) - \sum_j H(S_j)$ [32]. Whenever $< X|Z|Y >$ holds in G (or F), $Ind(X, Z, Y)$ must hold in P. Therefore, we say that $Ind(X, Z, Y)$ is *implied* by G (or F).

3 The Rational of the Minimum Entropy Approach

This section briefly reviews the rational behind the minimum entropy approach originally presented in [32].

Given $\mathcal{M}$ over N, we would like to learn a DMN (G, P) that is an approximation of $\mathcal{M}$. To measure the *closeness* of (G, P) to $\mathcal{M}$, we adopt the Kullback-Leibler cross entropy [21]: $K(P_{\mathcal{M}}, P) = \sum_{\overline{x}} P_{\mathcal{M}}(\overline{x}) \log(P_{\mathcal{M}}(\overline{x})/P(\overline{x}))$, where $P_{\mathcal{M}}$ is the *true* jpd defined by $\mathcal{M}$ and $\overline{x}$ is a configuration of N. A DMN that minimizes $K(P_{\mathcal{M}}, P)$ will be considered as the *best* approximation of $\mathcal{M}$. Since $K(P_{\mathcal{M}}, P) = H(N) - H_{\mathcal{M}}(N)$ [32], where $H(N)$ is the entropy of N defined by P and $H_{\mathcal{M}}(N)$ is defined by $\mathcal{M}$, minimizing $K(P_{\mathcal{M}}, P)$ can be achieved by simply minimizing $H(N)$. We call this the minimum entropy approach.

In Section 4.1, we will show that a DMN that minimizes $K(P_{\mathcal{M}}, P)$ is actually an I-map of $\mathcal{M}$. Thus, the best DMN is a *minimal* I-map, i.e., an I-map that contains no superfluous links. The problem of learning a minimal I-map is NP-hard [1]. Therefore, it is justified to use heuristic learning methods. We can design a learning algorithm by combining the entropy metric with a single-link lookahead search strategy. We will refer to such an algorithm as *learning by minimum entropy search*. One such algorithm [32] starts with an empty graph. At each pass, it searches all possible links and adds to the current graph the link that minimizes the entropy. It terminates when no additional link can decrease the entropy *significantly*. In Section 5, we identify a class of PMs that cannot be learned by such a single-link lookahead search. A multi-link lookahead search is required to discover the dependencies in these PMs. In the following discussion, we will assume a more general search procedure with the single-link lookahead as a special case.

4 The Minimum Entropy Search

In this section, we analyze how the dependency relations are derived in a DMN in minimum entropy search.

Recall that the pair (G, P) is a DMN of $\mathcal{M}$. That is, P is defined by the marginals of $P_{\mathcal{M}}$ on cliques of G. In practice, we can only estimate these marginals from a database of cases, e.g., using the maximum-likelihood estimator (the relative frequencies). According to the *law of large numbers*, the

above. However, there the term *Markov network* is restricted to a minimal I-map of a given dependency model. We do not require the structure of a DMN to be an I-map.

relative frequency of each configuration approaches its true probability as the size of the database approaches infinity. Since our objective here is to analyze the microscopic mechanism of the minimum entropy search and its asymptotic behavior, one may assume that P is obtained directly from the projection of $P_{\mathcal{M}}$. As we move from the theoretical analysis to practical implementation in Section 6, we will discuss the related issues.

Let us outline the theorems to be presented in this section. Theorem 2 establishes the relationship between the entropy of a DMN and its I-mapness. Theorem 3 identifies a false independence relation in a DMN if its entropy is not minimum. Theorem 6 says that if the inclusion of one or more links can remove a false independence relation, the entropy of the DMN will decrease. Together, Theorems 3 and 6 state that the process of decreases in entropy closely parallels the process of removal of false independence relations satisfied by the intermediate DMNs. Theorem 7 summerizes Theorems 2, 3 and 6. It asserts that the minimum entropy search algorithm will produce an I-map. Theorems 9 and 10 reveal the built-in resistance of the entropy metric to some superfluous links. Theorem 9 shows that if a link does not help remove a false independence from the current DMN, the entropy scoring metric will not select such a link. Theorem 10 shows how a greedy search (add only those links that maximizes the decrease of entropy at each pass) further helps reduce superfluous links.

4.1 Characterization of the minimum entropy search space

Let us first show that the entropy of a DMN cannot be smaller than that of the underlying $\mathcal{M}$. This means that the search space of DMNs is lower-bounded in terms of the entropy scoring metric as indicated by the following corollary.

Corollary 1 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let (G, P) be a decomposable Markov network of $\mathcal{M}$. Let $H_{\mathcal{M}}(N)$ be the entropy of N defined by $\mathcal{M}$ and $H(N)$ be the entropy of N defined by P. Then $H(N) \geq H_{\mathcal{M}}(N)$.*

Proof:

Let $P_{\mathcal{M}}$ be the jpd defined by $\mathcal{M}$. The cross entropy $K(P_{\mathcal{M}}, P) \geq 0$ [21]. Since $K(P_{\mathcal{M}}, P) = H(N) - H_{\mathcal{M}}(N)$ [32], we have $H(N) \geq H_{\mathcal{M}}(N)$.
□

The following theorem says that the lower bound of the search space can only be reached by a DMN that is an I-map of $\mathcal{M}$. Therefore, it shows clearly that the minimum entropy search targets an I-map.

Theorem 2 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let (G, P) be a decomposable Markov network of $\mathcal{M}$. Let $H_{\mathcal{M}}(N)$ be the entropy of N defined by $\mathcal{M}$ and $H(N)$ be the entropy of N defined by P. Then $H(N) = H_{\mathcal{M}}(N)$ iff (if and only if) G is an I-map of $\mathcal{M}$.*

Proof:

The cross entropy $K(P_{\mathcal{M}}, P) = 0$ iff $P = P_{\mathcal{M}}$ [21]. Because $K(P_{\mathcal{M}}, P) = H(N) - H_{\mathcal{M}}(N)$ [32], $H(N) = H_{\mathcal{M}}(N)$ is equivalent to $P = P_{\mathcal{M}}$. Since (G, P) is a DMN, we form a JF of G and have $P = \prod_i P_{\mathcal{M}}(C_i) / \prod_j P_{\mathcal{M}}(S_j) = P_{\mathcal{M}}$ where C_i is a clique of F and S_j is a sepset. This means that every independence relation implied by G is true in $\mathcal{M}$, namely, G is an I-map of $\mathcal{M}$. □

4.2 Construction of an I-map

Theorem 2 implies that if the entropy of a DMN is not the minimum, it must contain a false independence relation. The next theorem describes such a false independence relation more specifically in terms of its topological features.

Theorem 3 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let (G, P) be a decomposable Markov network of $\mathcal{M}$ and F be the junction forest of G. Let $H_{\mathcal{M}}(N)$ be the entropy of N defined by $\mathcal{M}$ and $H(N)$ be the entropy of N defined by P.*

If $H(N) > H_{\mathcal{M}}(N)$, there exist three disjoint subsets $X \neq \phi$, Z and $Y \neq \phi$, where Z is either empty or is a sepset of F, $X \cup Z$ is a clique of F and $Y \cup Z$ covers a connected subgraph of F such that $Ind(X, Z, Y)$ holds in P but does not hold in $\mathcal{M}$.

Proof:

Suppose that $H(N) > H_{\mathcal{M}}(N)$. The graph G is not completely connected, since otherwise we would have $P = P_{\mathcal{M}}$ and $H(N) = H_{\mathcal{M}}(N)$. This implies that G has more than one clique, or equivalently, F has at least two nodes. Let C_0 be a leaf node of F. If C_0 is the only node of a JT in F, let $S_0 = \phi$. This is the case of Figure 2 (left). Otherwise, let S_0 be the sepset of C_0 and its unique adjacent clique. This is the case of Figure 2 (right). Then P satisfies $Ind(C_0 \setminus S_0, S_0, N \setminus C_0)$. If $Ind(C_0 \setminus S_0, S_0, N \setminus C_0)$ does not hold in $\mathcal{M}$, the proof is complete with $X = C_0 \setminus S_0 \neq \phi$, $Y = N \setminus C_0 \neq \phi$ and $Z = S_0$ (Z may be empty).

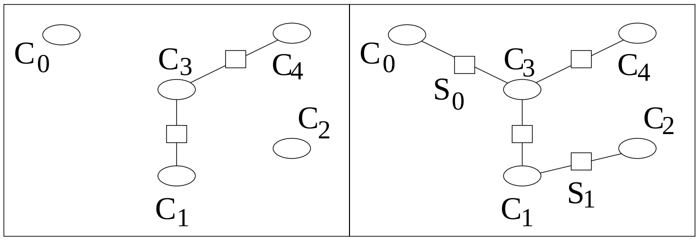

Figure 2: Two junction forests. Nodes of are drawn as ovals and sepsets are drawn as boxes.

Suppose $Ind(C_0 \setminus S_0, S_0, N \setminus C_0)$ holds in $\mathcal{M}$. Let N_0 denote the set $S_0 \cup (N \setminus C_0)$. For example, in Figure 2 (both left and right), $N_0 = \cup_{i=1}^{4} C_i$. In the case where $S_0 \neq \phi$, we have $P_{\mathcal{M}} = P_{\mathcal{M}}(N_0)P_{\mathcal{M}}(C_0)/P_{\mathcal{M}}(S_0)$, which implies $H_{\mathcal{M}}(N) = H_{\mathcal{M}}(N_0) + H_{\mathcal{M}}(C_0) - H_{\mathcal{M}}(S_0)$. On the other hand,

$$H(N) = H(N_0) + H(C_0) - H(S_0),$$

where $H(N_0)$ is computed from the projected distribution P of $\mathcal{M}$ on the subgraph of F without the node C_0. Denote this subgraph by F_0 (a JF). Since $H(C_0)$ is computed from $P(C_0)$ and $P(C_0) = P_{\mathcal{M}}(C_0)$ by definition, we obtain

$$H(N) = H(N_0) + H_{\mathcal{M}}(C_0) - H_{\mathcal{M}}(S_0).$$

Therefore, the assumption $H(N) > H_{\mathcal{M}}(N)$ implies $H(N_0) > H_{\mathcal{M}}(N_0)$.

In the case where $S_0 = \phi$, we have $P_{\mathcal{M}} = P_{\mathcal{M}}(N_0)P_{\mathcal{M}}(C_0)$, $H_{\mathcal{M}}(N) = H_{\mathcal{M}}(N_0) + H_{\mathcal{M}}(C_0)$ and $H(N) = H(N_0) + H_{\mathcal{M}}(C_0)$. The same result is obtained.

Since $H(N_0) > H_{\mathcal{M}}(N_0)$, we can repeat the above argument on F_0. Because F has only a finite number of nodes, we will eventually find an independence relation $Ind(X, Z, Y)$ that holds in P but not in $P_{\mathcal{M}}$. We now show that the contrary leads to a contradiction.

Suppose such an $Ind(X, Z, Y)$ is not found when F is reduced to *only* two cliques C_1 and C_2, i.e., $N_0 = C_1 \cup C_2$. Denote the only sepset as $S_1 = C_1 \cap C_2$. By the above argument, we have $H(C_1 \cup C_2) > H_{\mathcal{M}}(C_1 \cup C_2)$, where $H(C_1 \cup C_2)$ is computed from the projected distribution P of $\mathcal{M}$ on the subgraph of F with only cliques C_1 and C_2.

In the case where $S_1 \neq \phi$, since P satisfies $Ind(C_1 \setminus S_1, S_1, C_2 \setminus S_1)$, we obtain $H(C_1 \cup C_2) = H_{\mathcal{M}}(C_1) + H_{\mathcal{M}}(C_2) - H_{\mathcal{M}}(S_1)$. If $Ind(C_1 \setminus S_1, S_1, C_2 \setminus S_1)$ holds in $\mathcal{M}$ as well, then marginalization produces

$$P_{\mathcal{M}}(C_1 \cup C_2) = \sum_{N \setminus (C_1 \cup C_2)} P_{\mathcal{M}} = P_{\mathcal{M}}(C_1)P_{\mathcal{M}}(C_2)/P_{\mathcal{M}}(S_1).$$

This is equivalent to $H_{\mathcal{M}}(C_1 \cup C_2) = H_{\mathcal{M}}(C_1) + H_{\mathcal{M}}(C_2) - H_{\mathcal{M}}(S_1)$, which implies $H(C_1 \cup C_2) = H_{\mathcal{M}}(C_1 \cup C_2)$. Thus, we obtain a contradiction.

In the case where $S_1 = \phi$, P satisfies $Ind(C_1, \phi, C_2)$. Therefore, we have $H(C_1 \cup C_2) = H_{\mathcal{M}}(C_1) + H_{\mathcal{M}}(C_2)$. If $Ind(C_1, \phi, C_2)$ holds in $\mathcal{M}$ as well, then $P_{\mathcal{M}}(C_1 \cup C_2) = \sum_{N \setminus (C_1 \cup C_2)} P_{\mathcal{M}} = P_{\mathcal{M}}(C_1)P_{\mathcal{M}}(C_2)$ and $H_{\mathcal{M}}(C_1 \cup C_2) = H_{\mathcal{M}}(C_1) + H_{\mathcal{M}}(C_2)$. This again implies $H(C_1 \cup C_2) = H_{\mathcal{M}}(C_1 \cup C_2)$. □

Theorem 6 will show that if a sepset in a JF can be augmented to remove a false independence relation, then the augmentation will decease the entropy. Lemma 4 and 5 prepare for Theorem 6 by showing the result in a two-clique JF.

Augmentation of a two-clique JF can be exhaustively classified into the seven cases shown in Figure 3, where X, Y, Z, A and B represent disjoint and *nonempty* sets.

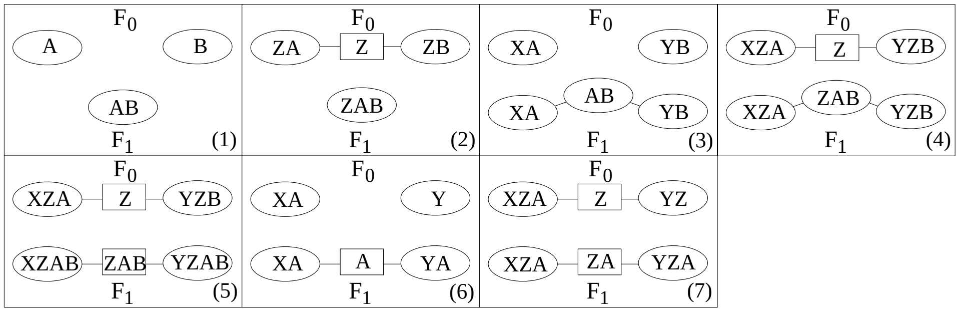

Figure 3: Seven cases of augmentation of a two-clique JF F_0 into another JF F_1. Cliques are drawn as ovals and sepsets as boxes. Sepsets of F_1 for cases (3) and (4) are not shown.

Cases (1) and (2) are the only possible ways a two-clique JF can be augmented into a single-clique JT. The augmentation corresponds to including links between each pair (a, b) where $a \in A$ and $b \in B$ in the chordal graph of F_0.

Cases (3) and (4) are the only possible ways a two-clique JF can be augmented into a three-clique JT. The augmentation corresponds to including links between each pair (a, b) where $a \in A$ and $b \in B$.

If the augmentation of case (4) is performed with additional links between each pair of (a, y) and (b, x) where $x \in X$ and $y \in Y$, we obtain case (5).

For cases (3) and (4), if we let $B = \phi$ in F_0, we get cases (6) and (7), respectively. The augmentation corresponds to including links between each pair (a, y).

No other case is possible except for a renaming of sets. For example, if we let $A = \phi$ in cases (6) and (7) and augment the chordal graph of F_0 with links (x, y), the result is cases (1) and (2), respectively.

Lemma 4 relates cases (1) and (2) to the decrease of entropy.

Lemma 4 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let A, B and Z be disjoint sets such that $A \neq \phi$, $B \neq \phi$, and $Z \cup A \cup B = N$. Let F_0 be a junction forest of two cliques $Z \cup A$ and $Z \cup B$. Let F_1 be a junction tree of a single clique $Z \cup A \cup B$. Let $H_0(N)$ and $H_1(N)$ be the entropies defined by the projected distributions of $\mathcal{M}$ on F_0 and F_1, respectively.*

If there exists a pair $a \in A$, $b \in B$ such that $Ind(\{a\}, N \setminus \{a, b\}, \{b\})$ does not hold in $\mathcal{M}$, then $H_1(N) < H_0(N)$.

Proof:

The topology of F_0 satisfies $< A|Z|B >$. That $Ind(\{a\}, N \setminus \{a, b\}, \{b\})$ does not hold in $\mathcal{M}$ implies that $Ind(A, Z, B)$ is false in $\mathcal{M}$. Therefore, F_0 is not an I-map of $\mathcal{M}$. According to Corollary 1 and Theorem2, we have $H_0(N) > H_{\mathcal{M}}(N)$. Since F_1 is a trivial I-map of $\mathcal{M}$, by Theorem 2, we have $H_1(N) = H_{\mathcal{M}}(N)$. □

Lemma 5 relates cases (3) through (7) to the decrease of entropy.

Lemma 5 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let X, Y, Z, A and B be disjoint sets such that $X \cup A \neq \phi$, $Y \cup B \neq \phi$, $A \cup B \neq \phi$, and $X \cup Y \cup Z \cup A \cup B = N$. Let F_0 be a junction forest of two cliques $X \cup Z \cup A$ and $Y \cup Z \cup B$. Let F_1 be a junction forest constructed by augmentation of F_0 as in cases (3) through (7) of Figure 3. Let $H_0(N)$ and $H_1(N)$ be the entropies defined by the projected distributions of $\mathcal{M}$ on F_0 and F_1, respectively.*

If $Ind(X \cup A, Z, Y \cup B)$ does not hold in $\mathcal{M}$,[2] but $Ind(X, Z \cup A \cup B, Y)$ holds in $\mathcal{M}$, then $H_1(N) < H_0(N)$.

Proof:

Again, we denote $X \cup A$ by XA. Since $Ind(XA, Z, YB)$ is implied by F_0 but it does not hold in $\mathcal{M}$, F_0 is not an I-map of $\mathcal{M}$. On the other hand, since $Ind(X, ZAB, Y)$ is the only independence relation implied by F_1 and it holds in $\mathcal{M}$, F_1 is an I-map of $\mathcal{M}$. According to Corollary 1 and Theorem 2, we can immediately conclude that $H_0(N) > H_1(N)$. □

Now we want to show in general that if links are added to a DMN to remove a false independence relation, then the addition will decease the entropy. This implies that the minimum entropy search is precisely a process of removal of false independence relations.

Theorem 6 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let $G_0 = (N, E_0)$ be a chordal graph and F_0 be the junction forest of G_0.*

Let X, Y, Z, A and B be disjoint subsets of N ($X \cup A \neq \phi$, $Y \cup B \neq \phi$, $A \cup B \neq \phi$) and F_0^* be a subgraph of F_0 such that (1) the union of all cliques in F_0^* equals to $X \cup Y \cup Z \cup A \cup B$, (2) either $Z = \phi$ or Z is a sepset in F_0^*, and (3) $Ind(X \cup A, Z, Y \cup B)$ is implied by F_0^*.

Let G_0 be augmented into one of the following three chordal graphs by including links only among variables in $X \cup Y \cup Z \cup A \cup B$.

G_1: $X \cup Y \cup Z \cup A \cup B$ becomes a single clique.

G_2: $X \cup Z \cup A \cup B$ and $Y \cup Z \cup A \cup B$ become two cliques.

G_3: $X \cup Z \cup A$, $Y \cup Z \cup B$ and $Z \cup A \cup B$ become three cliques.

Let $H_i(N)$ ($i = 1, 2, 3$) be the entropy defined by the projected distributions of $\mathcal{M}$ on G_i.

Suppose $Ind(X \cup A, Z, Y \cup B)$ does not hold in $\mathcal{M}$.

1. *If there exist $u \in X \cup A$ and $v \in Y \cup B$ such that $Ind(\{u\}, N \setminus \{u, v\}, \{v\})$ does not hold in $\mathcal{M}$, then $H_1(N) < H_0(N)$.*
2. *If $Ind(X, Z \cup A \cup B, Y)$ holds in $\mathcal{M}$, then $H_i(N) < H_0(N)$ ($i = 2, 3$).*

Proof:

The theorem is true if $XYZAB = N$ due to Lemma 4 and 5. Therefore, we only have to consider the case $W = XYZAB \subset N$ here.

[2] The augmentation of case (5) is necessary only if in addition neither $Ind(\{b\}, N \setminus \{b, x\}, \{x\})$ nor $Ind(\{a\}, N \setminus \{a, y\}, \{y\})$ holds in $\mathcal{M}$. We include case (5) here anyway as a unique augmentation.

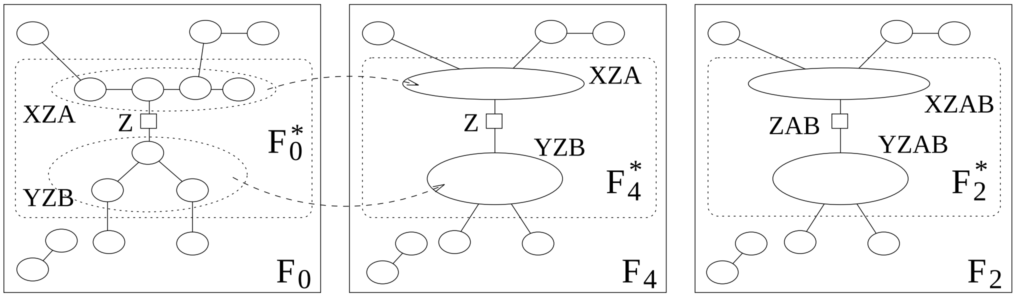

Figure 4: Illustration of the proof for Theorem 6.

First, we consider statement 2 with $i = 2$. Since $Ind(XA, Z, YB)$ is implied by F_0^*, cliques of F_0^* can be separated into two groups with the union of one group being XZA and the other being YZB. Figure 4 (left) illustrates this for the case where $Z \neq \phi$. Cliques outside F_0^* are disconnected from F_0^*, only connected to XZA group, or only connected to YZB group.

Let F_4 be a JF, otherwise identical to F_0, except that the XZA group is augmented into one clique and the YZB group into another (Figure 4 middle). Let F_4^* denote the subgraph consisting of these two cliques.

Let $H_4(N)$ be the entropy defined by the projected distributions of $\mathcal{M}$ on F_4, and $H_0^*(W)$ and $H_4^*(W)$ be the entropies on F_0^* and F_4^*, respectively. We have $H_0(N) = H_0^*(W) + h$ where h is the entropy contribution from outside F_0^*. Similarly we have $H_4(N) = H_4^*(W) + h$.

Since $Ind(XA, Z, YB)$ is implied by F_0^*, we have $H_0^*(W) = H_0^*(XZA) + H_0^*(YZB) - H_{\mathcal{M}}(Z)$, where $H_0^*(XZA)$ and $H_0^*(YZB)$ are the entropy contributions of the two subgraphs of F_0^*, and $H_{\mathcal{M}}(Z)$ is the entropy of Z defined by $\mathcal{M}$. Since $Ind(XA, Z, YB)$ is also implied by F_4^*, we have $H_4^*(W) = H_{\mathcal{M}}(XZA) + H_{\mathcal{M}}(YZB) - H_{\mathcal{M}}(Z)$. If we restrict $\mathcal{M}$ to XZA and YZB and apply Corollary 1, we have $H_{\mathcal{M}}(XZA) \leq H_0^*(XZA)$ and $H_{\mathcal{M}}(YZB) \leq H_0^*(YZB)$, which implies $H_4(N) \leq H_0(N)$.

Let F_2 be the JF of G_2 as shown in Figure 4 (right). Let the subgraph of F_2 over W be F_2^* and $H_2^*(W)$ be the entropy defined by the projected distributions of $\mathcal{M}$ on F_2^*. We then have $H_2(N) = H_2^*(W) + h$.

By assumption, $Ind(XA, Z, YB)$ does not hold in $\mathcal{M}$, but $Ind(X, ZAB, Y)$ holds in $\mathcal{M}$. Therefore, if we restrict $\mathcal{M}$ to W and apply Lemma 5, we get $H_2^*(W) < H_4^*(W)$, which implies $H_2(N) < H_4(N)$. Hence, we have $H_2(N) < H_4(N) \leq H_0(N)$.

Using similar arguments and replacing F_2 by a JF of G_3, we can prove that statement 2 with $i = 3$ holds.

Likewise, replacing F_2 by a JF of G_1 and applying Lemma 4, we can show $H_1(N) < H_0(N)$. □

Theorem 7 says that, started with an arbitrary DMN, if the entropy of the current DMN is not the minimum, a sequence of DMNs can be found which monotonically decreases the entropy to its minimum. It therefore establishes the asymptotic behavior of the minimum entropy search.

Theorem 7 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let (G, P) be a decomposable Markov network of $\mathcal{M}$. Let $H(N)$ be the entropy of N defined by P, and $H_{\mathcal{M}}(N)$ defined by $\mathcal{M}$. If $H(N) > H_{\mathcal{M}}(N)$, there exists a sequence $(G, P) = (G_0, P_0), \ldots, (G_k, P_k)$ of decomposable Markov networks*[3] *of $\mathcal{M}$ with the corresponding sequence of entropies $H(N) = H_0(N) > \ldots > H_k(N) = H_{\mathcal{M}}(N)$ such that G_i $(i = 1, \ldots, k)$ is constructed by adding links to G_{i-1}, and the last graph G_k is an I-map of $\mathcal{M}$.*

Proof:

Suppose $H(N) > H_{\mathcal{M}}(N)$. By Theorem 3, an independence relation $Ind(X', Z, Y')$ that holds in P but not in $\mathcal{M}$ can be found, where Z is either empty or is a sepset in the JF F of G, $X' \neq \phi$, $Y' \neq \phi$ and Y' covers a connected subgraph of F. Denote the corresponding subgraph of G covered by $X' \cup Y' \cup Z$ by G'.

We claim that there exist disjoint subsets X, Y, A and B of N ($XA = X'$, $YB = Y'$ and $AB \neq \phi$) such that (1) $Ind(X, ZAB, Y)$ holds in $\mathcal{M}$ and (2) it is implied by a chordal graph G_1 formed by augmenting the subgraph G' of G. Denote the augmented G' by G'_1 which is a subgraph of G_1.

Statement (1) is true since, in the extreme case where $X = Y = \phi$, $A = X'$ and $B = Y'$, i.e., G'_1 is a single clique, we have the trivial independence relation $Ind(\phi, X'ZY', \phi)$ that always holds. This augmentation is the case G_1 in Theorem 6.

To see that G_1 is chordal (statement (2)) in this case, recall the proof of Theorem 3. There the subsets X', Y' and Z are found by recursively removing leaf cliques from F (or equivalently from G) until the subgraph G' of G is found, where $Ind(X', Z, Y')$ holds in P but not in $\mathcal{M}$. Now since $X = Y = \phi$, cliques of G_1 are identical to those of G except cliques covered by G' are unioned into a single clique $X' \cup Y' \cup Z$ in G'_1. If we apply to G_1 Graham reduction [24], which recursively removes leaf cliques of a graph, we will end up with an empty graph (Graham reduction *succeeds*). This is because we can follow the same sequence of leaf clique removal that leads us from G to G', and then remove the clique $X' \cup Y' \cup Z$ in G_1 at the last step. The success of Graham reduction implies that G_1 is chordal [24] (p460).

In general, subsets $X \neq \phi$ and $Y \neq \phi$ can be found such that $Ind(X, ZAB, Y)$ holds in $\mathcal{M}$ and the subgraph G'_1, augmented as cases G_2 or G_3 in Theorem 6, is chordal. We now show that statement (2) is still true, i.e., the augmented graph G_1 is chordal. Again, we can apply Graham reduction to G_1 to first produce G'_1. Since G'_1 is chordal by assumption, continuation of Graham reduction will eventually succeed, which implies that G_1 is chordal.

Projecting $P_{\mathcal{M}}$ to G_1, we obtain a new DMN (G_1, P_1). From the discussion above, G is augmented into G_1 through one of the three cases of Theorem 6.

[3] Frydenberg and Lauritzen [10] (p553) proved that, given two chordal graphs with one being the subgraph of the other, there is an increasing sequence of chordal graphs between them that differ by exactly one link. Our result here involves a sequence of chordal graphs that differ by more than one links and fix some false independence relations.

In any case, Theorem 6 dictates that (G_1, P_1) satisfies $H_1(N) < H_0(N)$. If $H_1(N) > H_{\mathcal{M}}(N)$, the above arguments lead to (G_2, P_2) that satisfies $H_2(N) < H_1(N)$.

Since only a finite number of links can be added to G and the entropy of a DMN with a complete graph is equal to $H_{\mathcal{M}}(N)$, the sequence $(G_0, P_0), \ldots, (G_k, P_k)$ of DMNs does exist. By Theorem 2, G_k is an I-map of $\mathcal{M}$. □

Theorem 7 illustrates the microscopic working mechanism of the minimum entropy search. The entropy acts as a motor that drives the search for identifying a false independence relation. The removal of the false independence moves the current state forward in a chain leading the starting DMN to the goal I-map.

4.3 Superfluous links

Theorem 7 ensures that the minimum entropy search halt and produce an I-map. It does not, however, eliminate the possibility of producing a trivial I-map. Now we want to show that in practice halting at a trivial I-map rarely occurs. We identify two types of superfluous links that may be added. In fact, the entropy scoring metric has some built-in resistance to these two types of superfluous links. However, there exists a third type of superfluous link in to which the entropy scoring metric has no resistance. We discuss the third type in Section 7.

We start the search with an empty DMN. At each pass, links are added to correct a false independence relation and thus the entropy is reduced. Eventually we will obtain an I-map of $\mathcal{M}$. To examine the possibility of halting at a trivial I-map, we ask the following two questions:

1. Will those links that do not correct a false independence reduce the entropy?

2. Can the entropy scoring metric distinguish a *direct* dependence from an *indirect* dependence?

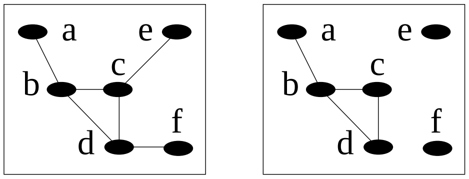

Figure 5: Left: An example of a minimal I-map of a PM. Right: The structure of a DMN generated during learning.

The first question concerns the possibility of adding what we refer to as *uncalled-for* links. For example, suppose the graph in the left of Figure 5 is the minimal I-map of a PM. Assume that the current learned structure is the graph in the right with the link (c, e) missing. If the link (a, c) is added next, it is an uncalled-for link since it does not correct any false independence in the current structure.

The second question concerns the inclusion of what we refer to as *redundant* links. Redundant links repair a false independence but not in the most direct way. In Figure 5 (right), since e is disconnected from the rest of the graph, it implies that e is independent of every other variable. This is a false independence since e is connected to every other variable in the minimal I-map (left). In Figure 5 (right), if the link (a, e) is added next, it is a redundant link. It repairs the false independence between a and e. Since it does not repair the false independence between c and e, the link (c, e) must eventually be included, making (a, e) redundant.

Note that the classification of superfluous links into uncalled-for versus redundant is for conceptual convenience and is not absolute. Whether a superfluous link is classified as one or the other depends on the current structure. If the current structure already contains (c, e), the link (a, e) would be classified as uncalled-for rather than redundant.

We provide a definite answer to the first question and a partial one to the second.

Definition 8 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables, and $G_{\mathcal{M}} = (N, E_{\mathcal{M}})$ be a minimal I-map of $\mathcal{M}$. Let $G_1 = (N, E_1)$ and $G_2 = (N, E_2)$ be two chordal graphs such that $E1 \subset E2$.*

G_1 and G_2 are `equivalent partial I-maps` *of $\mathcal{M}$, if $E_1 \cap E_{\mathcal{M}} = E_2 \cap E_{\mathcal{M}}$.*

If G_1 is the current structure and G_2 is a candidate new structure, then the set $E_2 \setminus E_1$ of links formalizes what we call uncalled-for links. Theorem 9 shows that a minimum entropy search will never add any uncalled-for links. Thus it answers the first question.

Theorem 9 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let $G_1 = (N, E_1)$ and $G_2 = (N, E_2)$ be two equivalent partial I-maps of $\mathcal{M}$. Let $H_1(N)$ and $H_2(N)$ be the entropies defined by the projected distributions of $\mathcal{M}$ on G_1 and G_2, respectively. Then $H_2(N) = H_1(N)$.*

Proof:

Let $P_1(N)$ and $P_2(N)$ be the projected distributions of $\mathcal{M}$ on G_1 and G_2, respectively. Making use of the independence relations implied by G_1, we have $P_1(N) = (\prod_i P_{\mathcal{M}}(C_i))/(\prod_j P_{\mathcal{M}}(S_j))$, where C_i is a clique of G_1 and S_j is a sepset. By Definition 8, G_2 does not remove any independence relations that are implied by G_1 but do not hold in $\mathcal{M}$. Therefore, $P_2(N) = P_1(N)$. □

An intermediate structure may imply many false independence relations. In order not to include many redundant links, we must not correct just any false relation. Theorem 10 shows that the number of redundant links can be reduced if we choose to correct the false relation that maximizes the decrement of entropy. It says that, given three subsets A, B and C of variables, if A and B are dependent, and A and C are either marginally independent or conditionally independent given B, then including links between A and B reduces entropy more than including links between A and C. This result formally justifies the use of a *greedy* search.

Theorem 10 *Let $\mathcal{M}$ be a probabilistic model over a set N of variables. Let $G = (N, E)$ be a chordal graph, A, B and C be three distinct cliques of G and A is disconnected from B and C. Let G_1 be a chordal graph formed by only adding links to G such that $A \cup B$ becomes a clique. Let G_2 be a chordal graph formed by only adding links to G such that $A \cup C$ becomes a clique. Let $H_1(N)$ and $H_2(N)$ be entropies defined by Markov networks of $\mathcal{M}$ with structures G_1 and G_2, respectively.*

Then we have $H_1(N) < H_2(N)$ if (1) $Ind(A, \phi, B)$ does not hold in $\mathcal{M}$ and (2) either $Ind(A, \phi, C)$ holds in $\mathcal{M}$, or $Ind(A, B, C)$ holds in $\mathcal{M}$ but $Ind(A, C, B)$ does not.

Proof:

In G_1, a new clique AB replaces cliques A and B. Hence, we have $H_1(N) = H(N) + H_{\mathcal{M}}(AB) - H_{\mathcal{M}}(A) - H_{\mathcal{M}}(B)$, where $H(N)$ is the entropy of the DMN with the structure G, and $H_{\mathcal{M}}(AB)$ is the entropy of the new clique defined by $\mathcal{M}$. Similarly, $H_2(N) = H(N) + H_{\mathcal{M}}(AC) - H_{\mathcal{M}}(A) - H_{\mathcal{M}}(C)$. Therefore, we have

$$H_2(N) - H_1(N) = H_{\mathcal{M}}(AC) - H_{\mathcal{M}}(C) - H_{\mathcal{M}}(AB) + H_{\mathcal{M}}(B).$$

Using the well known *average mutual information* between two sets U and V of variables,

$$I(U;V) = \sum_{UV} P(UV) \log \frac{P(UV)}{P(U)P(V)}$$

we obtain

$$\begin{aligned} & H_2(N) - H_1(N) \\ &= [H_{\mathcal{M}}(A) + H_{\mathcal{M}}(C) - I_{\mathcal{M}}(A;C)] - H_{\mathcal{M}}(C) - [H_{\mathcal{M}}(A) + H_{\mathcal{M}}(B) - I_{\mathcal{M}}(A;B)] + H_{\mathcal{M}}(B) \\ &= I_{\mathcal{M}}(A;B) - I_{\mathcal{M}}(A;C). \end{aligned}$$

If $Ind(A, \phi, C)$ holds in $\mathcal{M}$, then $I_{\mathcal{M}}(A;C) = 0$. Since $Ind(A, \phi, B)$ does not hold in $\mathcal{M}$, we have $H_2(N) - H_1(N) = I_{\mathcal{M}}(A;B) > 0$.

On the other hand, if $Ind(A, B, C)$ holds in $\mathcal{M}$, then $I_{\mathcal{M}}(A;B) = I_{\mathcal{M}}(A;C) + I_{\mathcal{M}}(A;B|C)$ [12] (equation 2.3.18), where $I(A;B|C)$ is the *average conditional mutual information* between A and B given C,

$$I(A;B|C) = \sum_{ACB} P(ACB) \log \frac{P(A|CB)}{P(A|C)}.$$

Hence, $I_{\mathcal{M}}(A;B) - I_{\mathcal{M}}(A;C) = I_{\mathcal{M}}(A;B|C) \geq 0$, with equality iff $Ind(A, C, B)$ holds in $\mathcal{M}$. Since $Ind(A, C, B)$ does not hold by assumption, $I_{\mathcal{M}}(A;B) - I_{\mathcal{M}}(A;C) > 0$. □

Theorem 10 answers the second question partially. It only asserts that redundant links will never be added under certain situations, but it does not guarantee the total avoidance of such links. Since finding the minimal I-map is NP-hard, it is unlikely that any heuristic search using any scoring metric will be able to eliminate all redundant links.

5 When Will a Single-Link Lookahead Search Fail?

Theorem 7 states that as long as the current DMN is not yet an I-map, a set of links can always be added such that the new DMN is closer to an I-map. No upper bound is given for the number of links that must be added each time. If we use a greedy algorithm as suggested by Theorem 10, a single-link lookahead search needs only to explore $O(N^2)$ links before one link is added. The number of links to be explored increases to $O(N^4)$ for a double-link lookahead, and to $O(N^6)$ for a triple-link lookahead. The single-link lookahead search has been adopted by several learning algorithms [18, 8, 2, 31, 22, 32] for computational efficiency. However, the following question is unanswered: What might be compromised by using a single-link lookahead search?

With the understanding of the minimum entropy search, we answer the above question in this section. Theorem 11 shows the existence of a class of PMs that displays a special pattern of dependence relations. Theorem 12 shows that a single-link lookahead search is unable to learn the I-map for this class of PMs.

Theorem 11 *Given an integer $\eta \geq 3$, there exists a probabilistic model $\mathcal{M}$ of a set X of η binary variables such that*

(S1) *for each $v \in X$, $P_{\mathcal{M}}(X \setminus \{v\}) = \prod_{x \in X, x \neq v} P_{\mathcal{M}}(x)$, and*

(S2) *for each pair $v, w \in X$ and $v \neq w$, $Ind(\{v\}, X \setminus \{v, w\}, \{w\})$ does not hold in $\mathcal{M}$.*

We shall refer to such a model as a `pseudo-independent` *model.*

Before proving the theorem, we describe intuitively the dependency pattern displayed by the pseudo-independent models. S2 means that no pair of variables of X are independent given all other variables. Therefore, in the *minimal* I-map $G_{\mathcal{M}}$ of $_{\mathcal{M}}$, there must be a *direct* line between each pair of them, i.e., $G_{\mathcal{M}}$ is a complete graph. We will refer to variables in such PMs as *collectively dependent*. On the other hand, S1 means that variables in any subset of X of size $\eta - 1$ are *pairwise marginally independent*.

Proof:

It is sufficient to construct a jpd given η such that S1 and S2 hold.

Let $x_1, \ldots, x_\eta$ denote η binary variables and $P(x_{i,0}) = P(x_{i,1}) = 0.5$ ($i = 1, \ldots, \eta$) where $x_{i,0}$ and $x_{i,1}$ are the two outcomes of x_i. There are exactly η distint subsets of X of size $\eta - 1$. For each subset $\{x_{i_1}, \ldots, x_{i_{\eta-1}}\}$ where $1 \leq i_j \leq \eta$, S1 is equivalent to

$$P(x_{i_1}, \ldots, x_{i_{\eta-1}}) = 0.5^{\eta-1}.$$

We have omitted the second index because the particular configuration does not affect the probability value. Models that satisfy S1 do exist. A jpd of mutual independence $P^* = P(x_1, \ldots, x_\eta) = 0.5^\eta$ is one example. However, P^* does not satisfy S2. We will construct a jpd of $\mathcal{M}$ which satisfies both S1 and S2.

We can view the above equation, which is equivalent to S1, as a constraint

$$P(x_{i_1}, \ldots, x_{i_{\eta-1}}, x_{i_\eta,0}) + P(x_{i_1}, \ldots, x_{i_{\eta-1}}, x_{i_\eta,1}) = 0.5^{\eta-1}$$

for the subset $\{x_{i_1}, \ldots, x_{i_{\eta-1}}\}$. We therefore have η constraints, one for each subset.

To construct a desired jpd, we assign a probability value to each of the 2^η configurations, each of which is denoted by a binary η-tuple. For example, the configuration $(x_{1,0}, \ldots, x_{\eta,0})$ is denoted $(0, \ldots, 0)$. We group the tuples according to the number of 1s contained in each tuple and index the groups as $GP_0, \ldots, GP_\eta$. For example, GP_0 has a single tuple $(0, \ldots, 0)$, GP_1 has η tuples $(0, \ldots, 0, 1)$, $(0, \ldots, 0, 1, 0)$, ..., and $(1, 0, \ldots, 0)$.

We assign probability values to configurations group by group in the ascending order of the group index. To make a new assignment, we check the configurations whose probability values have been assigned, determine how many of the η constraints are involved in the assignment, and ensure that the new assignment conform to the constraints.

We start by assigning the single configuration in GP_0: $P(0, \ldots, 0) = 0.5^{\eta-1}q$ where $q \in [0, 1]$ and $q \neq 0.5$. This assignment does not violate any constraints. We then assign a configuration in GP_1:

$$P(0, \ldots, 0, 1) = P(x_{1,0}, \ldots, x_{\eta-1,0}) - P(x_{1,0}, \ldots, x_{\eta-1,0}, x_{\eta,0}) = 0.5^{\eta-1}(1-q).$$

Note that the assignment involves only one constraint and involves the only configuration whose value has been assigned. We will say that the assignment of probability value to configuration $(0, \ldots, 0, 1)$ involves the above constraint *relative to* the configuration $(0, \ldots, 0, 0)$.

We make the following observation: If c_1 is a configuration whose probability has been assigned and c_2 is a configuration whose probability is to be assigned, then the assignment involves a constraint relative to c_1 if and only if c_1 and c_2 differ by exactly one attribute.

The observation implies two things. First, the assignment of c_2 cannot involve a constraint relative to another configuration in the same group, since configurations in the same group differ by at least two attributes. For example, $(0, \ldots, 0, 1)$ and $(0, \ldots, 1, 0)$ in GP_1, and $(0, \ldots, 0, 1, 1)$ and $(0, \ldots, 1, 0, 1)$ in GP_2.

Second, if $c_2 \in GP_i$, the assignment of c_2 can only involve a constraint relative to configurations in GP_{i-1}. This is because configurations in GP_j ($j \leq i-2$) differ from c_2 by at least two 1s. Therefore, when we assign a configuration, we only have to check configurations in the very last group assigned. Note that the assignment may still involve multiple constraints each relative to a distinct configuration. For example, the assignment of $(0, \ldots, 0, 1, 1, 1)$ in GP_3 involves three constraints relative to $(0, \ldots, 0, 1, 1)$, $(0, \ldots, 0, 1, 1, 0)$ and $(0, \ldots, 0, 1, 0, 1)$ in GP_2, respectively. We show that all of the constraints involved can be satisfied simultaneously.

Each configuration in GP_1 involves a single constraint relative to the single configuration $(0, \ldots, 0)$ in GP_0. To satisfy each constraint, we assign the configuration $0.5^{\eta-1}(1-q)$ as we did in the second assignment above. Hence all configurations in GP_1 have the *same* probability value, since all distributions of $\eta - 1$ order have the same value $0.5^{\eta-1}$. Therefore, for each configuration $c \in GP_2$, even though it involves two constraints, each relative to a different configuration in GP_1, the assignment $P(c) = 0.5^{\eta-1}q$ satisfies both simultaneously.

Thus, by following this procedure, we can construct a jpd for $\mathcal{M}$ by alternating the assignment of $0.5^{\eta-1}q$ and $0.5^{\eta-1}(1-q)$ to configurations in successive groups. The resultant jpd clearly satisfies S1.

To show that the jpd also satisfies S2, we need to show, for an arbitrary pair x_i, x_j $(i \neq j)$ and $Y = X \setminus \{x_i, x_j\}$, that $P(x_i|x_j, Y) \neq P(x_i|Y)$, or equivalently, $P(x_i, x_j, Y) \neq P(x_i|Y)P(x_j, Y)$. Since $P(x_i|Y) = 0.5$ and $P(x_j, Y) = 0.5^{\eta-1}$ by S1, we have $P(x_i|Y)P(x_j, Y) = 0.5^{\eta}$. However, $P(x_i, x_j, Y)$ has the value $0.5^{\eta-1}q$ or $0.5^{\eta-1}(1-q)$ where $q \neq 0.5$. □

Consider the following example of a pseudo-independent model. Suppose we have a digital gate with three inputs x_i $(i = 1, 2, 3)$ and an output x_4. The output $x_4 = 1$ whenever any two inputs are 0 and a third input is 1, or all inputs are 1. Suppose the three inputs are independent to each other and each of them has equal chance to be 0 or 1. Table 1 shows the jpd of these four variables. Readers are encouraged to verify that (1) the marginal distribution of each variable is 0.5, (2) any subset of two or three variables are mutually independent, and (3) the jpd is not $0.5^4 = 0.0625$.

(x_1, x_2, x_3, x_4)	$P(x_1, x_2, x_3, x_4)$	(x_1, x_2, x_3, x_4)	$P(x_1, x_2, x_3, x_4)$
$(0,0,0,0)$	0.125	$(1,0,0,0)$	0
$(0,0,0,1)$	0	$(1,0,0,1)$	0.125
$(0,0,1,0)$	0	$(1,0,1,0)$	0.125
$(0,0,1,1)$	0.125	$(1,0,1,1)$	0
$(0,1,0,0)$	0	$(1,1,0,0)$	0.125
$(0,1,0,1)$	0.125	$(1,1,0,1)$	0
$(0,1,1,0)$	0.125	$(1,1,1,0)$	0
$(0,1,1,1)$	0	$(1,1,1,1)$	0.125

Table 1: An example of a pseudo-independent model.

Since the parameter q in the proof of Theorem 11 can take any value in the intervals $[0, 0.5)$ and $(0.5, 1]$, there is an *infinite* number of models given η. In all these models, the marginal of each variable is equal to 0.5. However, it should be noted that pseudo-independent models are not restricted to 0.5 marginals. Table 2 provides a jpd of three variables that has different marginals, in which (1) the marginals are $P(x_{1,0}) = 0.6$, $P(x_{2,0}) = 0.4$ and $P(x_{3,0}) = 0.2$, (2) any two variables are marginally independent, and (3) the jpd is not equal to the product $P(x_1)P(x_2)P(x_3)$.

(x_1, x_2, x_3)	$P(x_1, x_2, x_3)$	(x_1, x_2, x_3)	$P(x_1, x_2, x_3)$
$(0, 0, 0)$	0.024	$(1, 0, 0)$	0.056
$(0, 0, 1)$	0.216	$(1, 0, 1)$	0.104
$(0, 1, 0)$	0.096	$(1, 1, 0)$	0.024
$(0, 1, 1)$	0.264	$(1, 1, 1)$	0.216

Table 2: A pseudo-independent model defined by different marginals for individual variables.

Among all PMs, pseudo-independent models represent one extreme. The other extreme is represented by models which display a totally different pattern of dependence relations. In the I-map of those models, no pair of variables connected by a link displays marginal independence. Between the two extremes, a whole spectrum of pseudo-independent models exist, in which variables are collectively dependent, marginally independent in some pairs and not marginally independent in other pairs. To classify these models, we shall refer to the models in Theorem 11 as *full* pseudo-independent models and the models between the two extremes as *partial* pseudo-independent models. Table 3 shows such a partial model of three variables. The marginal for each variable is 0.5. Any pair of variables are dependent given the third. However, x_1 and x_2 are marginally independent ($P(x_1, x_2) = P(x_1)P(x_2)$), so are x_1 and x_3, but x_2 and x_3 are *not* marginally independent ($P(x_2, x_3) \neq P(x_2)P(x_3)$).

(x_1, x_2, x_3)	$P(x_1, x_2, x_3)$	(x_1, x_2, x_3)	$P(x_1, x_2, x_3)$
$(0, 0, 0)$	0.225	$(1, 0, 0)$	0.20
$(0, 0, 1)$	0.025	$(1, 0, 1)$	0.05
$(0, 1, 0)$	0.025	$(1, 1, 0)$	0.05
$(0, 1, 1)$	0.225	$(1, 1, 1)$	0.20

Table 3: An example of a partial pseudo-independent model.

Before discussing more general pseudo-independent models, we show that the single-link lookahead search cannot learn full pseudo-independent models.

Theorem 12 *Let $G_{\mathcal{M}}$ be the minimal I-map of a full pseudo-independent model $\mathcal{M}$ over a set X of η variables. Let G_0 be an initial chordal graph from which the learning starts and let the number of links of G_0 be $L \leq (\eta(\eta - 1)/2) - 2$. Then $G_{\mathcal{M}}$ cannot be recovered by the single-link lookahead minimum entropy search.*

Proof:

Since $\mathcal{M}$ is a full pseudo-independent model, there is a link between each pair of variables in the minimal I-map of $\mathcal{M}$. Hence, $G_{\mathcal{M}}$ is a complete graph and has $M = \eta(\eta - 1)/2$ links.

Let (G_0, P_0) be the initial DMN. Suppose G_0 has $L \leq M - 2$ links. Then G_0 cannot have two cliques of size $\eta - 1$. Otherwise, G_0 will differ from a complete graph by a single link, i.e., $L = M - 1$. We will show that no link will be added to G_0 by the search.

Since only a single link can be added each time and the resultant graph must be chordal, at each pass of the search, either a clique of size 2 is formed by joining two nodes in *disconnected* components (e.g., the dotted link (b, c) in Figure 6) or a clique of size $k > 2$ is formed by joining two cliques of size $k - 1$ with their intersection of size $k - 2$. For example, a clique of size 3 is formed by joining two cliques of size 2 with their intersection of size 1 (e.g., the clique $\{a, b, d\}$ formed by adding the dotted link (a, d) in Figure 6). A clique of size 4 is formed by joining two cliques of size 3 with their intersection of size 2 (e.g., the clique $\{d, e, f, g\}$ formed by adding the dotted link (d, g) in Figure 6).

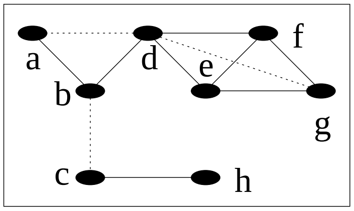

Figure 6: An example graph to show clique formation by single link addition. Dotted links are to be added (one at a time) to form new cliques.

Let (G_1, P_1) be a candidate DMN such that G_1 is augmented from G_0 by adding a single link (a, b). According to the discussion above, the link must join two cliques. Denote the two cliques by $W \cup \{a\}$ and $W \cup \{b\}$.

G_1 and G_0 differs in that cliques $W \cup \{a\}$ and $W \cup \{b\}$ are replaced by $W \cup \{a, b\}$. Therefore, the difference between the entropies of the two DMNs is

$$H_1(N) - H_0(N) = H_{\mathcal{M}}(Wab) - (H_{\mathcal{M}}(Wa) + H_{\mathcal{M}}(Wb) - H_{\mathcal{M}}(W)).$$

According to S1 in Theorem 11, variables in any subset of size $\eta - 1$ are pairwise marginally independent. Since the largest two cliques of an equal size in G_0 has a size $\eta - 2$, we have $|Wab| \leq \eta - 1$. Hence we have $Ind(Wa, \phi, b)$ and $Ind(W, \phi, b)$, which implies that $H_1(N) - H_0(N) = 0$. Therefore no (G_1, P_1) will be selected and no link will be added to G_0. □

Although Theorem 12 involves learning only full pseudo-independent models, the conclusion is applicable to learning partial pseudo-independent models as well. For example, if the single-link lookahead search is applied to the model in Table 3, it will only find the dependence between x_2 and x_3 and will output a structure with a single link. A two-link lookahead search after the single-link lookahead search will identify the collective dependency among the three variables.

So far, the pseudo-independent models are defined based on the entire domain of variables. This again is only a special case. In general, pseudo-independent models can be *embedded*. Examples of PMs with embedded pseudo-independent models are shown in Section 8.

The existence of vast number of pseudo-independent models poses a challenge to learning probabilistic networks as approximate I-maps. Suppose we have no prior knowledge about the possible size of an embedded pseudo-independent model. Then Theorem 11 dictates that search of potential cliques up to the size of the entire domain by multi-link lookahead is necessary in general. Since such a search is infeasible, prior knowledge should be used for restricting the number of links required for the search. We will discuss this problem in more detail in Section 6.

It is perhaps worth mentioning that Pearl noticed the existence of pseudo-independent models. In his book [28] (p395), he considered a bell that rings whenever the outcomes of two fair coins are equal to demonstrate collective dependence and pairwise marginal independence. However, there is no further discussion on such models.

We have shown that the single-link lookahead search combined with the entropy scoring metric is unable to learn pseudo-independent models. In fact, the same conclusion can be drawn in learning probabilistic networks (including DMNs and BNs) with other scoring metrics. We now show this is indeed the case in a few well-known algorithms. To simplify the discussion, we will only consider full embedded pseudo-independent models, i.e., models in which a subset of domain variables display collective dependence and pairwise marginal independence.

Pseudo-independent models cannot be learned by Kutato [18]. The algorithm starts with an empty graph and uses a single-link lookahead search to learn a BN with an entropy scoring metric. Since variables in a pseudo-independent model are pairwise marginally independent, no link between any pair will decrease the entropy and hence no dependence can be discovered.

Likewise, pseudo-independent models cannot be learned by the algorithm suggested by Lam and Bacchus [22], which uses the MDL principle to learn a BN. Let us first briefly describe their algorithm. It first computes the mutual information between each pair of nodes (corresponding to a link). It then places all links in a list in descending order of mutual information between the end nodes. The candidate BNs are generated by starting with an empty graph and including links from the beginning of the link list and down the list. It allocates equal amount of computational resources to explore candidate BNs of identical number of links (the same complexity). After each complexity class has exhausted its resources, the best candidate BN according to the cross entropy scoring metric is chosen. The BN that has the minimal description length across classes will be the final output. If the true PM contains an embedded pseudo-independent model, links between each pair of nodes in the model has zero mutual information. These links will be placed at the end of the link list and will be the last to be included in any candidate BNs. If these BNs are ever considered, the algorithm must have exhausted almost all possible BNs, which has an exponential complexity. Therefore, in practice, these BNs would have no chance to be tested and selected as the final output.

The previous two algorithms start with an empty graph. The algorithm PC [31] learns a BN by starting from a complete graph. In the first pass, it removes each link if the end nodes of the link are *marginally* independent. In the second pass, it removes each link if the end nodes of the link are independent conditioned on a third node. In each of the following passes, it remove each link if the end points of the link are independent conditioned on a subset of nodes of higher and higher order until a stopping condition is met. If the problem domain contains an embedded psedu-independent model, each pair of nodes in the model are marginally independent and the link between them will be deleted in the first pass of the search. Therefore the collective dependency of the model will not be reflected in the final BN.

It is an open question whether or not this limitation also applies to K2 [8], which uses the Bayesian scoring metric to learn a BN.

6 A Multi-link Lookahead Learning Algorithm

The existence of infinite number of pseudo-independent PMs and the inability of single-link lookahead search to learn such models suggest the adoption of more general learning algorithms when prior knowledge about the problem domain cannot rule out the possibility of such models. In this section and the section to follow, we present one such algorithm and discuss related issues. As we are now moving from the theoretical analysis of the minimum entropy search to its practical implementation, we make some explicit assumptions on the context where the proposed algorithm is to be applied.

Assumption 1 *The database variables are discrete.*

We have assumed a discrete problem domain throughout the paper as indicated in the beginning of Section 2.2. This assumption simply restates it in terms of the feature of the database.

Assumption 2 *The database is complete, i.e., no cases have missing variables.*

The above two assumptions are seen in most algorithms for learning probabilistic networks [8, 15].

To reduce the complexity of a multi-link lookahead search, we make the following sparseness assumption.

Assumption 3 *Let η be the size of an embedded pseudo-independent model in the problem domain. The higher the value of η, the less likely that a pseudo-independent model of size η exists in the problem domain.*

This assumption allows us to lookahead a small number of links such that we will not miss many embedded pseudo-independent models. In the case where the number of variables involved in an embedded pseudo-independent model is actually large, we probably will not be able to estimate its distribution reliably from the available data anyway. Even if the database is very large and such estimation is possible, the inference computation using such models will be very expensive, making them much less useful. Based on the sparseness assumption, the algorithm to be presented bounds the multi-link lookahead search with a parameter specified by the user.

A finite database may contain *false* dependencies that cause the generation of superfluous links in addition to those we discussed in Section 4.3. In the algorithm to be presented, the χ^2 test of conditional independence (CI test) is used to reduce such superfluous links. This will be discussed in detail in Section 7.

Algorithm 1

```
Input: A database $D$ over a set $N$ of variables, a maximum size $\eta$ of
        pseudo-independent models, and an $\alpha$ level for $\chi^2$ test.
begin
        initialize an empty graph $G = (N, E)$;
        $G' := G$;
        for $i = 1$ to $\eta(\eta - 1)/2$, do % search by levels
            repeat % search by passes
                initialize the entropy decrement $dh' := 0$;
                for each set $L$ of $i$ links ($L \cap E = \phi$), do % search by steps
                    if $G^* = (N, E \cup L)$ is chordal and $L$ is implied by a single
                        clique of size $\leq \eta$, then
                        compute the entropy decrement $dh^*$ locally;
                        if $dh^* > dh'$, then $dh' := dh^*$, $G' := G^*$;
                if $G'$ passes $\chi^2$ test at $\alpha$ level, then $G := G'$, done := false;
                else done := true;
            until done = true;
        return $G$ and the projected distribution $P$ of the database on $G$;
end
```

The search is structured into *levels* and each level is a search with the identical number of lookahead links. Each level consists of multiple *passes* and each pass is composed of multiple *steps*. Each pass at the same level tries to add the same number (i) of links. For instance, level one search adds a single link in each pass, level two search adds two links, and so on. Search at each pass selects i links after testing all distinct and legal combinations, one at each search step, of i links. The i links that decrease the entropy maximally are selected. The entropy decrement dh^* is computed *locally* using F_0^* and F_2^* in Figure 4. The χ^2 test for conditional independence is then performed to determine if the replacement of F_0^* is justified by the database at the α

significance level. If F_0^* is rejected, F_2^* will be adopted and search continues at the same level, otherwise the next higher level of search starts. The following analyzes the worst case time complexity of the algorithm.

Testing the chordality of G^* can be performed in $O(|N|)$ time [13].

A JT can be computed by a maximal spanning tree algorithm [19]. A maximal spanning tree of a graph with v nodes and e links can be computed in $O((v+e)\log v)$ time [25]. Since a complete graph has $O(v^2)$ links, a maximal spanning tree can be computed in $O(v^2 \log v)$ time. Equivalently, computation of a JT of a chordal graph with k nodes and v cliques takes $O(v^2 \log v)$ time. Since $v \leq k$, computation of a JT of a chordal graph with k nodes takes $O(k^2 \log k)$ time. In computing dh^*, we need to compute F_0^* and F_2^* from the corresponding chordal subgraphs. Each of them contains no more than 2η variables, where η is the maximum allowable size of a clique. Therefore, we can compute F_0^* and F_2^* in $O(\eta^2 \log \eta)$ time.

Let n be the number of cases in the database. We can extract the distribution P' on the 2η variables from the database directly in $O(n)$ time. The projected distribution on F_0^* and F_2^* can be computed by marginalizing P' to cliques and multipling clique distributions, which takes $O(\eta\ 2^\eta)$ time. The computation of dh^* from the projected distributions can be performed in $O(2^\eta)$ time. The complexity of each step is then $O(|N|+n+\eta\ (\eta \log \eta + 2^\eta))$. Since n is much larger than $|N|$, the complexity of each search step is $O(n+\eta\ (\eta \log \eta + 2^\eta))$.

The algorithm repeats for $O(\eta^2)$ levels. Each level contains $O(|N|^2)$ passes. Each pass has $C(C(|N|,2),\eta) = O(|N|^{2\eta})$ steps. Hence, the algorithm has $O(\eta^2\ |N|^{2\eta})$ search steps. The overall complexity of the algorithm is then $O(\eta^2\ |N|^{2\eta}\ (n+\eta\ (\eta \log \eta + 2^\eta)))$.

The computation is feasible only if η is close to one. Otherwise, it is not practical with problem domains of large number of variables. This suggests the use of prior knowledge about the problem domain to further constrain the search. By exploring the prior knowledge of the problem domain, if we can partition the problem domain N into β equal subdomains and assert that there is no embedded pseudo-independent models that crosses subdomain boundaries, then we can perform the single-link lookahead search in the entire problem domain but perform the multi-link lookahead search only in each subdomain. We will have $O(|N|^2 + \eta^2\ \frac{|N|^{2\eta}}{\beta^{2\eta}})$ search steps, which amounts to a complexity reduction of $\beta^{2\eta}$ times.

For example, suppose $|N| = 48$ and $\eta = 5$. The number of search steps is in the order of 6.5×10^{16}. If we can restrict the multi-link lookahead search to three subdomains of no more than 16 variables each, the number of search steps will be reduced to the order of 1.1×10^{12}.

Another useful heuristic is to apply single-link lookahead search first. If a disconnected network is generated and we have prior knowledge that it should be connected, then we can focus the multi-link lookahead search based on the resultant network. We leave such an investigation to future work.

7 Conditional Independence Test for Multi-link Inclusion

In Section 4, we showed that an I-map of a PM can be learned by the minimum entropy search when marginal distributions of cliques can be obtained *accurately*. This is equivalent to a database of infinite size, which contains only *true* dependencies. In Section 4.3, we classified two types (uncalled-for and redundant) of superfluous links that may be generated even when learning is performed using such databases. Hence, their generation is due to the use of heuristic search. These links are undesirable because they unnecessarily increase complexity of inference computation.

In practice, we must learn from a finite database. Such a database may contain *false* dependencies that do not exist in the underlying problem domain. They cause the generation of a third type of superfluous link which we will refer to as *false* links. False links have a different undesirable effect *apart from* the complexity increase shared by the other two types of superfluous links. The probability values associated with false links tend to encode noise contained in the database. The encoded noise biases the jpd of the learned network and causes inference errors. We will illustrate this in Section 8.

The description length metric [22], equivalent to the Bayesian scoring metric, uses the encoding length of the learned model to penalize *automatically* the generation of all three types of superfluous links. As discussed in Section 1, the entropy scoring metric is equivalent to the encoding length of the data given the learned model. We have shown in Section 4.3 that the entropy metric has total resistance to the generation of uncalled-for links and partial resistance to the generation of redundant links. However, it has *no* resistance to the false links at all. Without additional controls, the minimum entropy search tends to encode all false dependencies contained in the database. The CI test used in Algorithm 1 is aimed at reducing false links as well as redundant links.

In learning a DMN, Fung and Crawford [11] used the χ^2 test to determine if two variables are independent conditioned on a third set. They used a three-way contingency table for the test. The χ^2 test is also used in Kutato [18], but the details of the test are not given. Since the single-link lookahead search is used in Kutato and only the removal of one conditional independence between a pair of variables needs to be justified, the test can be performed in the same way as Fung and Crawford. In the multi-link lookahead search, however, the test is to justify the removal of possibly several conditional independencies (see Figure 4). Therefore, the method of Fung and Crawford cannot be used under these circumstances. We have instead applied the χ^2 test of *goodness-of-fit for composite hypotheses* [9]. We describe the method below.

Recall from Section 6 that the replacement of F_0^* by F_2^* causes the maximum amount of decrease of entropy. The CI test is performed for deciding if F_0^* should be rejected at the α level of significance. Suppose F_0^* is a JT, we can direct links of F_0^* to form a rooted tree. We then start at the root and index cliques as we move away from the root in a breadth-first fashion. Denote the variables involved in F_0^* by X. If F_0^* is a correct depiction of independencies among X, we have

$$P_{\mathcal{M}}(X) = \prod_i P_{\mathcal{M}}(C_i|S_i), \tag{1}$$

where C_i is a clique of F_0^* and S_i is the sepset between C_i and its parent clique. The total number of independent parameters on the right side of equation 1 is

$$\sum_i dc_i - \sum_j ds_j - 1,$$

where dc_i is the dimension of $P(C_i)$ and ds_j is the dimension of $P(S_i)$. The first summation is over all cliques and the second is over all sepsets. Note that this result is independent of how the links are directed in forming the rooted tree. In general, F_0^* is a JF and we have

$$P_{\mathcal{M}}(X) = \prod_k (\prod_i P_{\mathcal{M}}(C_i|S_i)), \tag{2}$$

where the first product is over all JTs of F_0^*. The total number of independent parameters needed to specify the right side of equation 2 is

$$s = \sum_k (\sum_i dc_i - \sum_j ds_j - 1), \tag{3}$$

where the first summation is over all JTs of F_0^*.

The *composite null hypothesis* is stated as follows:

Hy_0 : variables in X have the independence relations depicted by F_0^*

or equivalently "equation 2 is correct." The *alternative hypothesis* is

Hy_1 : variables in X does not have the independence relations depicted by F_0^*

or equivalently "'equation 2 is incorrect." If Hy_0 is true, the entropy reduction obtained by including a new set L of links is due to either false dependencies caused by finite sampling or L being redundant links. We therefore should accept F_0^*. Otherwise, updating F_0^* is justified.

To perform the test, we form the χ^2 test statistic

$$\chi^2 = n \sum_{\overline{x}} \frac{[P_D(\overline{x}) - \prod_k(\prod_i P_D(\overline{c_i}) / \prod_j P_D(\overline{s_j}))]^2}{\prod_k(\prod_i P_D(\overline{c_i}) / \prod_j P_D(\overline{s_j}))},$$

where n is the number of cases in D, $P_D()$ is the distribution estimated from database D, $\overline{x}$ is a configuration of X, $\overline{c_i}$ is a projection of $\overline{x}$ on C_i, $\overline{s_i}$ is a projection of $\overline{x}$ on S_i, $\prod_k$ is over all JTs, $\prod_i$ and $\prod_j$ are over all cliques and sepsets of the same JT, respectively. The degree of freedom of the χ^2 distribution is

$$df = d - 1 - s,$$

where d is the number of independent parameters in $P_D(X)$ and s is defined by equation 3.

The worst case complexity of a χ^2 test is about the same as that of computing dh^*.

The level of significance of the test α is supplied by the user. In essence, α specifies to what extent the user is willing to trade complexity of the generated network with fitness of data. As the value of α increases, both the complexity of the network and the degree of fitness to data increase. Therefore, in contrast to the automatic balance of model complexity and fitness of data in MDL and Bayesian approaches, the α level here acts as a user-controlled lever to balance the two. The necessity of such balance is discussed in [30]. In Section 8, we further illustrate experimentally the leverage that can be provided by the CI test.

8 Experimental Results

A set of ten DMNs were randomly generated to serve as the control PMs. We set up the simulator to embed a pseudo-independent model for some DMNs. Then databases were generated from each DMN using logic sampling. The learning algorithm was then applied to each database. Each learned DMN was then compared with the corresponding control DMN. To evaluate the quality of the learned DMN, we compared both the complexity of the DMN and its cross entropy with the control DMN. We will show the results on two DMNs in detail and then present the overall results for all ten DMNs.

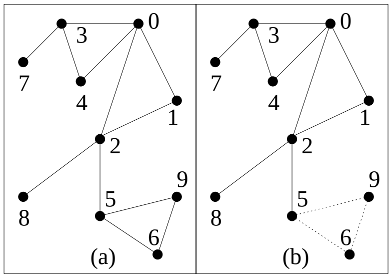

Figure 7: (a) The structure of a control DMN. (b) The structure of a DMN learned from a database generated by the control DMN in (a). Dotted links are the missing links in the learned DMN.

Figure 7 (a) is a generated DMN where variables 5, 6 and 9 form an embedded pseudo-independent model. A database of 10000 cases was generated and the learning algorithm was applied to the database with the α level 0.1%. Applying the single-link lookahead and double-link lookahead search, the DMN in Figure 7 (b) was learned, where dotted links represent the missing links. The search failed to learn the embedded pseudo-independent model. The cross entropy is 0.02173. Applying the triple-link lookahead search, the identical DMN as in (a) was learned. The cross entropy drops to 0.00088.

Figure 8 (a) is another control DMN. It does not contain any embedded pseudo-independent model. We generated a database of 250 (about 25% of

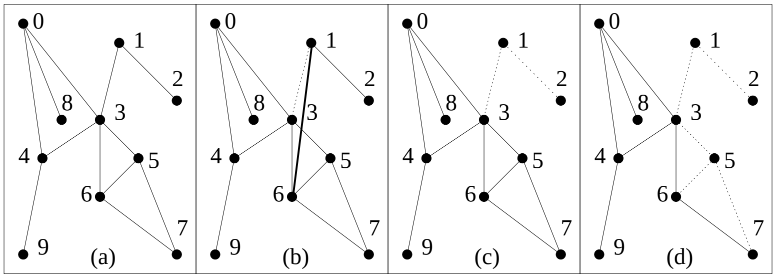

Figure 8: (a) The structure of a control DMN. (b) through (d) are the structures of DMNs learned from a database generated by the control DMN in (a). The α levels are 30%, 0.1% and 0.01%, respectively. The heavy link in (b) does not exist in (a). Dotted links are missing links in the learned DMN.

the total number of distinct configurations) cases. To compare the effect of different α levels, we used 30%, 0.1% and 0.01% with the corresponding learned DMNs shown in Figure 8 (b), (c) and (d), respectively. The heavy link does not exist in the control DMN. The dotted links are the missing links. The general trend is that as the value of α decreases, the complexity of the learned DMN decreases as expected.

The cross entropies of the three learned DMNs are 0.1444, 0.0649 and 0.2027. We interpret them as follows: When the database is *small*, it may contain many false dependencies. The true dependencies of the PM are also weakly represented. When the α value is high, false dependencies are easily included in the learned DMN which represent noise. This noise, once encoded, tends to increase the cross entropy. When the α value is small, both false dependencies and weak true dependencies are excluded. The missing true dependencies tend to increase the cross entropy. Between these two extremes of α values, there are values of α that minimize the false dependencies included and include sufficient true dependencies, which optimizes the cross entropy. We see that this is not the case when the database is relatively large.

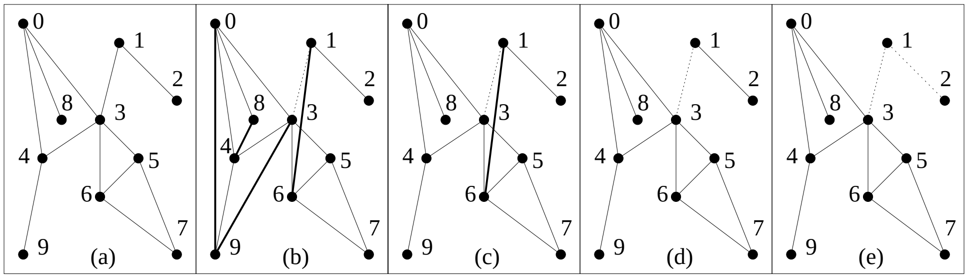

Figure 9: (a) The structure of a control DMN. (b) through (e) are the structures of DMNs learned from a database generated by the control DMN in (a). The α levels are 30%, 20%, 0.1% and 0.001%, respectively.

Using the same control DMN as in Figure 9 (a), we generated a larger database of 1000 (close to the total number of distinct configurations) cases. Figure 9 (b) through (e) show the learned DMNs using α values 30%, 20%, 0.1% and 0.001%. The corresponding cross entropies are 0.4885, 0.1158, 0.0805 and 0.0381. This time as the complexity of the learned DMNs decreases, the cross entropy decreases as well. We have found the same general trend in learning other DMNs. When the database is large, the strong true dependencies are so dominantly represented by the data, they will not be missed even with small α values. On the other hand, small α values tend to exclude false dependencies. This helps decrease the cross entropy. They also tend to exclude the weak true dependencies which are easily biased by noise. Since those weak dependencies do not matter much anyway, the overall effect amounts to just decreasing the cross entropy. By a careful examination of the control DMN, we notice that variable 1 is almost certain with its marginals 0.007 and 0.993 (very close to 0 and 1). Its dependencies with variables 2 and 3 are not properly represented in the data since configurations corresponding to the marginal 0.007 of variable 1 rarely occur in the database of 1000 cases. Therefore, the marginal independence among these variables as depicted in Figure 9 (d) and (e) appears closer to the control DMN than the biased dependence in (b) and (c) (the heavy link between variables 1 and 6).

Our results suggest the use of medium α values when the database is small and the use of small α values when the database is large.

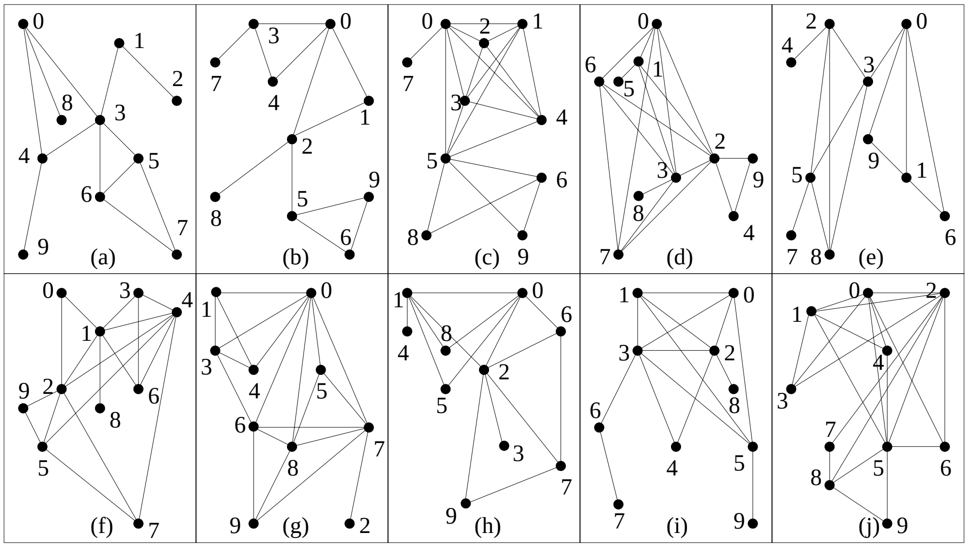

Figure 10: The structures of ten simulated control DMNs.

Figure 10 shows all ten control DMNs. The DMNs in (b) and (d) have embedded pseudo-independent models $\{5, 6, 9\}$ and $\{2, 4, 9\}$, respectively. Table 8 shows the summary of experimental results using databases of 10000 cases for each DMN. For each database, two different α values were used. We terminated the learning process after the triple-link lookahead search. The second column shows the total number of graphs searched when the triple-

	# nets	links	max clq	cross entr
model a		12	3	
$\alpha = 10\%$	5530	14	4	0.0145
$\alpha = 0.1\%$	6491	12	3	0.0137
model b		12	3	
$\alpha = 10\%$	7423	20	5	0.0215
$\alpha = 0.1\%$	13631	12	3	0.0009
model c		20	5	
$\alpha = 10\%$	11374	24	6	0.1967
$\alpha = 0.1\%$	10044	21	6	0.1942
model d		18	5	
$\alpha = 10\%$	5730	22	5	0.0283
$\alpha = 0.1\%$	4311	18	5	0.0070
model e		14	4	
$\alpha = 10\%$	4310	17	4	0.0048
$\alpha = 0.1\%$	5560	16	4	0.0042
model f		18	4	
$\alpha = 10\%$	3709	22	5	0.0037
$\alpha = 0.1\%$	4338	19	4	0.0021
model g		20	4	
$\alpha = 10\%$	3128	24	5	0.0044
$\alpha = 0.1\%$	3479	24	5	0.0044
model h		15	3	
$\alpha = 10\%$	3059	21	5	0.0038
$\alpha = 0.1\%$	5123	17	4	0.0020
model i		15	4	
$\alpha = 10\%$	4310	17	4	0.0037
$\alpha = 0.1\%$	4689	16	4	0.0033
model j		21	4	
$\alpha = 10\%$	3128	24	5	0.0053
$\alpha = 0.1\%$	4038	21	4	0.0024

Table 4: Summary of experimental results with the ten simulated DMNs in Figure 10.

link lookahead terminates. The third and fourth columns list the number of links and the size of the maximum clique in the control DMN and the learned DMNs as an indication of the complexity of the learned DMNs. The last column lists the cross entropy, indicating the closeness of the distribution of the learned DMNs relative to the control DMN. For all ten models, the learned DMNs have only a slight increase in complexity. The DMNs learned using the smaller α value are simpler and also have lower cross entropy as the size of databases is large. The total processing time (20 DMNs) was 6 minutes on a SGI INDY workstation.

9 Discussion

In this paper, we studied learning a decomposable Markov network from a database of cases using the entropy scoring metric and a heuristic search. Our analysis reveals the 'microscopic' mechanism of a minimum entropy search and its asymptotic behavior. We showed that the process to decrease the entropy parallels the process to remove false independence relations in the intermediate networks. The decreasing entropy drives the search forward until an I-map of the domain model is learned when the size of the database is very large.

The understanding of this mechanism uncovers that the I-map of a probabilistic model cannot be fully recovered unless some false independence relations (equivalently, a true dependence not yet encoded) can be identified at each search pass. We showed that there exists a large number of probabilistic models whose dependences can only be detected with a lookahead of multiple links. As a single-link lookahead search has been adopted by several learning algorithms for efficiency reasons, our analysis indicates that results obtained by these methods may be compromised.

To uncover the pseudo-independent models, we have proposed an algorithm that uses the multi-link lookahead search. We have suggested some ways in which prior knowledge can be applied to reduce the complexity of the multi-link lookahead search although more research is needed along this direction.

To avoid learning networks of unnecessarily high complexity has been a major concern in developing learning algorithms. We classified superfluous links into three types. We showed that the entropy metric has total resistance to one type, partial resistance to another, and no resistance to the third. We devised a method for applying the χ^2 test to reduce the generation of the latter two types of superfluous links. Experimental results using ten randomly generated DMNs of 10 variables demonstrated satisfactory performance of the multi-link lookahead algorithm.

Acknowledgement

We thank David Poole for a valuable discussion. We also thank T. Chu for developing the network simulator and M. Hoyle for reviewing an earlier draft. This work is supported by Research Grant OGP0000982 and OGP0155425 from the Natural Sciences and Engineering Research Council (NSERC). The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada, NSERC, and the participation of PRECARN Associates Inc.

References

[1] R.R. Bouckaert. Properties of Bayesian belief network learning algorithms. In R. Lopez de Mantaras and D. Poole, editors, *Proc. of Tenth Conference on Uncertainty in Artificial Intelligence*, pages 102–109, Seattle, Washington, 1994. Morgan Kaufmann.

[2] W. Buntine. Classifiers: a theoretical and empirical study. In R. Lopez de Mantaras and D. Poole, editors, *Proc. of 1991 International Joint Conference on Artificial Intelligence*, pages 638–644, Sydney, 1991.

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

[4] E. Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1991.

[5] P. Cheeseman. Overview of model selection. In *Proc. of 4th International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL, 1993. Society for AI and Statistics.

[6] D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: serach methods and experimental results. In *Proc. of 5th Conference on Artificial Intelligence and Statistics*, pages 112–128, Ft. Lauderdale, FL, 1995. Society for AI and Statistics.

[7] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, (14):462–467, 1968.

[8] G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, (9):309–347, 1992.

[9] M.H. DeGroot. *Probability and Statistics.* Addison Wesley, 1975.

[10] M. Frydenberg and S.L. Lauritzen. Decomposition of maximum likelihood in mixed graphical interaction models. *Biometrika*, 76(3):539–555, 1989.

[11] R.M. Fung and S.L. Crawford. Constructor: A system for the induction of probabilistic models. In *Proc. of AAAI*, pages 762–769, Boston, MA, 1990. MIT Press.

[12] R.G. Gallager. *Information Theory and Reliable Communication.* John Wiley and Sons, 1968.

[13] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

[14] P. Hajek, T. Hovranek, and R. Jirousek. *Uncertain Information Processing in Expert Systems.* CRC Press, 1992.

[15] D. Heckerman. A tutorial on learning Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Mocrisoft, WA, March 1995.

[16] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *To appear in Machine Learning*, 1995.

[17] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 149–163. Elsevier Science Publishers, 1988.

[18] E.H. Herskovits and G.F. Cooper. Kutato: an entropy-driven system for construction of probabilistic expert systems from database. In *Proc. Sixth Conference on Uncertainty in Artificial Intelligence*, pages 54–62, Cambridge, Mass., 1990.

[19] F.V. Jensen. Junction tree and decomposable hypergraphs. Technical report, JUDEX, Aalborg, Denmark, February 1988.

[20] 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.

[21] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[22] W. Lam and F. Bacchus. Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.

[23] S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, (50):157–244, 1988.

[24] D. Maier. *The Theory of Relational Databases.* Computer Science Press, 1983.

[25] U. Manber. *Introduction to Algorithms: a Creative Approach.* Addison-Wesley, 1989.

[26] R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems.* John Wiley and Sons, 1990.

[27] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, (29):241–288, 1986.

[28] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[29] G. Rebane and J. Pearl. The recovery of causal ploy-trees from statistical data. In *Proc. of Workshop on Uncertainty in Artificial Intelligence*, pages 222–228, Seattle, Washington, 1987.

[30] S.L. Sclove. Small-sample and large-sample statistical model selection criteria. In P. Cheeseman and R.W. Oldford, editors, *Selecting Models from Data*, pages 31–39. Springer-Verlag, 1994.

[31] P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–73, 1991.

[32] S.K.M. Wong and Y. Xiang. Construction of a Markov network from data for probabilistic inference. In *Proc. Third International Workshop on Rough Sets and Soft Computing*, pages 562–569, San Jose, CA, 1994.

[33] S.K.M. Wong, Y. Xiang, and X. Nie. Representation of Bayesian networks as relational databases. In *Proc. Fifth International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems*, pages 159–165, Paris, 1994.

[34] Y. Xiang. Distributed multi-agent probabilistic reasoning with Bayesian networks. In Z.W. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems*, pages 285–294. Springer-Verlag, 1994.

[35] Y. Xiang, B. Pant, A. Eisen, M. P. Beddoes, and D. Poole. Multiply sectioned Bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine*, 5:293–314, 1993.

[36] Y. Xiang, D. Poole, and M. P. Beddoes. Multiply sectioned Bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence*, 9(2):171–220, 1993.