Learning Conditional Independence Relations from a Probabilistic Model

Y. Xiang and S.K.M Wong
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
E-mail: yxiang@cs.uregina.ca, wong@cs.uregina.ca
Tel: Xiang (306) 585-4088, Wong (306) 585-4597
Fax: (306) 585-4745

December 22, 1994

Abstract

We consider the problem of learning conditional independencies, expressed as a Markov network, from a probabilistic model. An efficient algorithm employing a greedy search has been developed earlier with promising empirical results. However, two issues were not addressed. First, the reason why the myopic search works so well globally has not been fully understood. Second, whether the algorithm can find a correct Markov network in all cases has not been formally established.

In this paper, we prove that, for any given probabilistic model, the algorithm will always produce a Markov network whose structure is an independence map of the underlying model and whose associated probability distribution is identical to the underlying model. The proof also offers deeper insight into the algorithm's working mechanism.

As the problem of learning a minimal independence map of a given probabilistic model is NP-hard in general, our polynomial time algorithm does not guarantee minimality in all cases. We show that, however, if the given probabilistic model belongs to a subclass that has a singly connected independence map, the algorithm will always produce a Markov network whose structure is a minimal independence map.

Keywords: Reasoning under uncertainty, learning, Markov networks, belief networks, knowledge acquisition, knowledge representation.

1 Introduction

In designing many traditional information systems and AI systems, a central task is to express our knowledge about the underlying domain in some form of dependency relations. In communication and pattern-recognition systems, it is more desirable that the relations be specified using a joint probability distribution (jpd) [2, 8], either in its entirety or in a factorized form. In machine learning [14, 11, 12, 21], to construct classification rules or decision trees is the primary objective. For probabilistic reasoning in AI, algorithms were developed to learn a Bayesian network from data samples [6, 3, 15, 13].

In this paper we consider the representation of our knowledge by a Markov network (MN) which is an undirected graph associated with a factorized probability distribution. We have developed an efficient algorithm to learn a MN of a probabilistic model (PM) from data samples produced by the model. Such a representation has a number of advantages as will be discussed in the paper.

A description of our algorithm and the preliminary results were presented in [16]. Though the experimental results are promising, two important issues were not addressed. The algorithm employs a greedy search. It was not fully understood why such a myopic search worked so well globally. Secondly, whether a *correct* MN can be found for any PM was not formally established.

In this paper, we formulate the problem as learning an independence map (I-map) and its associated distribution expressed as a MN of a given PM. We will prove that, for any given PM, the proposed algorithm returns a correct MN. The proof of this assertion also offers deeper insight into the algorithm's working mechanism. As the problem of learning a minimal I-map of a given PM is NP-hard in general [1], any polynomial time algorithm would not guarantee minimality in every case. We show that, however, for a subclass of PMs, the proposed algorithm will indeed return a MN whose structure is a minimal I-map.

Section 2 provides the background and terminologies. Section 3 introduces the algorithm. Section 4 shows that our method produces a Markov network as a correct dependency model. Section 5 presents a sufficient condition under which the algorithm learns a MN whose structure is a minimal I-map. Section 6 compares this work with related work in traditional estimation of jpds [2, 8], learning of classification rules [14, 11, 12, 21] and learning of Bayesian networks [6, 3, 15, 13].

2 Background and Terminologies

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. A *clique* of a graph is a maximal set of nodes pairwise linked. A *component* of a graph is a maximal subgraph that is connected.

Let G be a chordal graph. If G is connected, 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 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. Without confusion, we sometimes refer to a node C in F as a clique when the nodes of G contained in C are of our concern. The intersection of two adjacent cliques in F is called the *sepset* of the two cliques.

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 a JF, we use $< X|Z|Y >$ to mean that either Z is a clique on the unique path between the clique that contains X and the clique that contains Y or Z is a sepset on that path.

Let N be a set of discrete variables and $X \subseteq N$. A *configuration* $\overline{x}$ of X is an assignment of values to every variable $x \in X$. 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 joint probability distribution (jpd) over N. However, in practice, we often do not have the jpd, but can only obtain marginal distributions over subsets of N. Our algorithm does not require the jpd as long as there are some independencies among variables. Hence we treat the PM as a means to obtain the necessary marginals without using the jpd as an intermediate step. We only use the jpd as a conceptual entity. The entropy of N defined by a jpd P is $H(N) = -\sum_{\overline{x}} P(\overline{x}) \log(P(\overline{x}))$.

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 a PM M over N if there is an 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 $< X|Z|Y > \Rightarrow Ind(X, Z, Y)$. An I-map guarantees that variables graphically separated are independent. However, it does not guarantee that variables graphically connected are necessarily dependent. Conversely, G is a *D-map* of M, if $Ind(X, Z, Y) \Rightarrow < X|Z|Y >$. We call G a *perfect map* of M if it is both a I-map and a D-map. For a formal treatment of graphical representation of dependency models, see Pearl [13].

Among possible I-maps of a given PM, we only consider those that are chordal due to many desirable properties of chordal graphs [5, 13, 10, 7]. To quantify the strength of dependencies, we associate a chordal I-map with a probability distribution defined as follows: Let M be a PM over a set N of variables, $G = (N, E)$ be a chordal graph and F be a JF of G. Let C_i be a clique of F and S_j be a sepset of F. Let $P(C_i)$ and $P(S_j)$ be the marginal distributions over C_i and S_j, respectively, defined by M. The jpd $P = (\prod_i P(C_i))/(\prod_j P(S_j))$ is called the *projected distribution of M on G (or on F)*. The entropy of N defined by P can be obtained by $H(N) = \sum_i H(C_i) - \sum_j H(S_j)$ [16]. Whenever $< X|Z|Y >$ holds in G (or F), $Ind(X, Z, Y)$ must hold in P of M on G (or F). Therefore, we shall say that $Ind(X, Z, Y)$ is *implied* by G (or F).

Given a PM M over N and a chordal graph $G = (N, E)$, we shall call the

pair (G, P) a *Markov network* (MN) of M, where P is the projected distribution of M on G. We shall call G the structure of the MN, and P the distribution of the MN. A Markov network defines a probabilistic model[1].

3 The Learning Algorithm

Our objective is to learn a MN (G, P) of a given PM M over a set N of variables. Ideally, we would like G to be a *minimal* I-map [13] of M, i.e., an I-map that includes no superfluous links. Since the problem of learning a minimal I-map is NP-hard [1], we settle to develop an algorithm that learns *efficiently* (G, P) such that G is close to a minimal I-map. A description of the algorithm and the preliminary experimental results can be found in [16]. Here we briefly review the basic ideas and present the algorithm in pseudocode.

To measure the *closeness* of (G, P) to M, we adopt the Kullback-Leibler cross-entropy [9]: $K(P_M, P) = \sum_{\overline{x}} P_M(\overline{x}) \log(P_M(\overline{x})/P(\overline{x}))$, where P_M is the *true* jpd defined by M and $\overline{x}$ is a configuration of N. The MN that minimizes $K(P_M, P)$ will be taken as an approximation of M[2]. Since $K(P_M, P) = H(N) - H_M(N)$ [16], where $H(N)$ is the entropy of N defined by P and $H_M(N)$ defined by M, minimizing $K(P_M, P)$ can be achieved by simply minimizing $H(N)$. One can minimize $H(N)$ by an exhaustive search of all MNs, which is computationally intractable. Instead, we have developed the following greedy algorithm to find an approximate MN for M.

Algorithm 1
Input: A probabilistic model M over a set N of variables, and a threshold ϵ.
begin
    construct a graph $G = (N, E)$ with the set of links $E = \phi$;
    compute the projected distribution P of M on G;
    compute entropy h from P, $h := \sum_x H(\{x\})$;
    $h' := h$, $G' := G$, $P' := P$, done := true;
    repeat
        for each pair $x, y \in N$ such that $(x, y) \notin E$, do
            if $G'' = (N, E \cup \{(x, y)\})$ is chordal, then
                compute the junction forest F'' of G'';
                compute the projected distribution P'' of M on F'';
                compute entropy h'' from P'', $h'' := \sum_i H(C_i) - \sum_j H(S_j)$,
                    where C_i is a clique of F'' and S_j is a sepset of F'';
                if $h'' < h'$, then $h' := h''$, $G' := G''$, $P' := P''$;
        if $h - h' \geq \epsilon$, then $h := h'$,$G := G'$,$P := P'$,done := false;
        else done := true;
    until done = true or G becomes a complete graph;
    return G and P;
end

The algorithm starts with an empty graph. At each step, it searches all

[1] *Markov network* is used in [13] to denote a minimal I-map of a dependency model. We use the term to mean the graph and the associated distribution as a whole, since what we call *projected distribution* is sometimes referred to as *Markov distribution* in the literature [5]. We do not require the structure of a Markov network to be a *minimal* I-map.

[2] We will show that it is always possible to obtain $P = P_M$. The MN (G, P) is considered as an *approximation* in the sense that G may not be a perfect map of M.

possible links and adds to the current graph the link that minimizes the entropy. It terminates when no additional link can cause significant (larger than the threshold) decrease of the entropy. It can be easily verified that the algorithm runs in $O(n^4)$ time in the worst case, where n is the cardinality of N. Refer to [16] for how to choose an appropriate threshold ϵ.

To illustrate Algorithm 1, we use a given PM M whose perfect map is depicted in Figure 1. The two marginal distributions necessary to specify M is given in Table 1. The numbers (search step index) in Figure 1 indicate the order in which links are added by the algorithm.

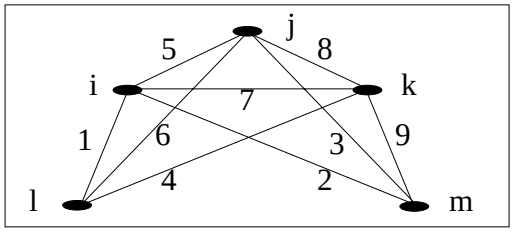

Figure 1: A Markov network.

$p(i_0$	&	l_0	&	j_0	&	$k_0)$	$= 0.47879$	$p(i_0$	&	m_0	&	j_0	&	$k_0)$	$= 0.45359$
$p(i_0$	&	l_0	&	j_0	&	$k_1)$	$= 0.11339$	$p(i_0$	&	m_0	&	j_0	&	$k_1)$	$= 0.11969$
$p(i_0$	&	l_0	&	j_1	&	$k_0)$	$= 0.04927$	$p(i_0$	&	m_0	&	j_1	&	$k_0)$	$= 0.03360$
$p(i_0$	&	l_0	&	j_1	&	$k_1)$	$= 0.00840$	$p(i_0$	&	m_0	&	j_1	&	$k_1)$	$= 0.00770$
$p(i_0$	&	l_1	&	j_0	&	$k_0)$	$= 0.02520$	$p(i_0$	&	m_1	&	j_0	&	$k_0)$	$= 0.05040$
$p(i_0$	&	l_1	&	j_0	&	$k_1)$	$= 0.01260$	$p(i_0$	&	m_1	&	j_0	&	$k_1)$	$= 0.00630$
$p(i_0$	&	l_1	&	j_1	&	$k_0)$	$= 0.00672$	$p(i_0$	&	m_1	&	j_1	&	$k_0)$	$= 0.02240$
$p(i_0$	&	l_1	&	j_1	&	$k_1)$	$= 0.00560$	$p(i_0$	&	m_1	&	j_1	&	$k_1)$	$= 0.00630$
$p(i_1$	&	l_0	&	j_0	&	$k_0)$	$= 0.15119$	$p(i_1$	&	m_0	&	j_0	&	$k_0)$	$= 0.14040$
$p(i_1$	&	l_0	&	j_0	&	$k_1)$	$= 0.02700$	$p(i_1$	&	m_0	&	j_0	&	$k_1)$	$= 0.02970$
$p(i_1$	&	l_0	&	j_1	&	$k_0)$	$= 0.00960$	$p(i_1$	&	m_0	&	j_1	&	$k_0)$	$= 0.00960$
$p(i_1$	&	l_0	&	j_1	&	$k_1)$	$= 0.00060$	$p(i_1$	&	m_0	&	j_1	&	$k_1)$	$= 0.00180$
$p(i_1$	&	l_1	&	j_0	&	$k_0)$	$= 0.06480$	$p(i_1$	&	m_1	&	j_0	&	$k_0)$	$= 0.07559$
$p(i_1$	&	l_1	&	j_0	&	$k_1)$	$= 0.02700$	$p(i_1$	&	m_1	&	j_0	&	$k_1)$	$= 0.02430$
$p(i_1$	&	l_1	&	j_1	&	$k_0)$	$= 0.01440$	$p(i_1$	&	m_1	&	j_1	&	$k_0)$	$= 0.01440$
$p(i_1$	&	l_1	&	j_1	&	$k_1)$	$= 0.00540$	$p(i_1$	&	m_1	&	j_1	&	$k_1)$	$= 0.00420$

Table 1: Marginal distributions for the two cliques in Figure 1.

4 I-mapness of the Algorithm

A fundamental question that one must answer is whether, given an arbitrary PM M, Algorithm 1 will return a MN (G, P) such that G is an I-map of M and $P = P_M$, where P_M is the jpd defined by M.

It seems possible that, at some stage while $P \neq P_M$, the algorithm can add no *single* link to the current G such that (1) the resultant G is chordal and (2) the entropy is decreased. The effects of (1) and (2) may only be obtained by adding *multiple* links at a time. The algorithm would terminate prematurely if that is the case.

In the following, we show that Algorithm 1 will always return a MN (G, P) such that G is an I-map of M and $P = P_M$. We will establish the results

through a series of propositions and a final theorem, which also lead to a better understanding of the working mechanism of the algorithm.

Proposition 2 shows that if the entropy of the current MN (G, P) is not minimum, it must contain a false independence relation.

Proposition 2 *Let M be a probabilistic model over a set N of variables. Let (G, P) be a Markov network of M. Let $H_M(N)$ be the entropy of N defined by M, and $H(N)$ be the entropy of N defined by P.*

If $H(N) > H_M(N)$, there exist three disjoint subsets X, Z and Y of N such that $Ind(X, Z, Y)$ holds in P but does not hold in M.

Proof:

Suppose $H(N) > H_M(N)$. The graph G is not completely connected, since otherwise we would have $P = P_M$ and $H(N) = H_M(N)$. This implies that G has more than one clique, or equivalently, the JF F of G 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 $S_0 = \phi$. Otherwise, let S_0 be the sepset of C_0 with its unique adjacent clique. 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 M, the proof is complete.

Suppose $Ind(C_0 \setminus S_0, S_0, N \setminus C_0)$ holds in M. Let N_0 denote the set $S_0 \cup (N \setminus C_0)$. We have $P_M = P_M(N_0)P_M(C_0)/P_M(S_0)$, which implies $H_M(N) = H_M(N_0) + H_M(C_0) - H_M(S_0)$. On the other hand, $H(N) = H(N_0) + H(C_0) - H(S_0) = H(N_0) + H_M(C_0) - H_M(S_0)$, where $H(N_0)$ is computed from the projected distribution P of M on the subgraph of F without the node C_0. Denote this subgraph by F_0 (a JF). The above implies $H(N_0) > H_M(N_0)$.

Since $H(N_0) > H_M(N_0)$, we can repeat the above procedure on F_0. Because F has only finite number of nodes, eventually an independence relation $Ind(X, Z, Y)$ in P will be found that does not hold in M. We will show that the contrary will lead to a contradiction.

Suppose such an $Ind(X, Z, Y)$ has not been found when F is reduced to *only* two nodes: C_1 and C_2 with the sepset $S_{1,2}$. By the above argument, we have $H(C_1 \cup C_2) > H_M(C_1 \cup C_2)$, where $H(C_1 \cup C_2)$ is computed from the projected distribution P of M on the subgraph of F with only nodes C_1 and C_2. Since P satisfies $Ind(C_1 \setminus S_{1,2}, S_{1,2}, C_2 \setminus S_{1,2})$, we obtain $H(C_1 \cup C_2) = H_M(C_1) + H_M(C_2) - H_M(S_{1,2})$. If $Ind(C_1 \setminus S_{1,2}, S_{1,2}, C_2 \setminus S_{1,2})$ holds in M as well, then marginalization produces $\sum_{N \setminus (C_1 \cup C_2)} P_M = P_M(C_1)P_M(C_2)/P_M(S_{1,2})$. This is equivalent to $H_M(C_1 \cup C_2) = H_M(C_1) + H_M(C_2) - H_M(S_{1,2})$, which implies $H(C_1 \cup C_2) = H_M(C_1 \cup C_2)$. Thus, we obtain a contradiction. □

The following lemma establishes a property of the well known average mutual information $I(U; V)$ between two sets U and V of variables. We will use this property in the proof of Proposition 4.

Lemma 3 *Let W be a set of variables and x and y be two distinct variables not contained in W. Then $I(W \cup \{x\}; \{y\}) - I(\{x\}; \{y\}) \geq 0$, and the equality holds iff (if and only if) W is independent of y given x.*

Proof:

For simplicity, we will write Wx to denote $W \cup \{x\}$.

$$I(Wx;y) - I(x;y) = \sum_{Wxy} P(Wxy) \log \frac{P(Wxy)}{P(Wx)P(y)} - \sum_{xy} P(xy) \log \frac{P(xy)}{P(x)P(y)}$$

$$= \sum_{Wxy} P(Wxy) \log \frac{P(Wxy)P(x)P(y)}{P(Wx)P(y)P(xy)} = \sum_{Wxy} P(Wxy) \log \frac{P(W|xy)}{P(W|x)} = I(W;y|x)$$

where $I(W;y|x)$ is the *average conditional mutual information* between W and y given x. It has been shown [4] that $I(W;y|x) \geq 0$ and the equality holds iff W is independent of y given x. □

Proposition 4 says that if a link can be added to the current MN to remove a false independence relation, then the addition must decease the entropy.

Proposition 4 *Let M be a probabilistic model over a set N of variables. Let X, Y, Z, $\{a\}$ and $\{b\}$ ($\{a\} \cup \{b\} \neq \phi$) be disjoint subsets of N. Let F_0 be a junction forest with two adjacent cliques $X \cup Z \cup \{a\} \cup \{b\}$ and $Y \cup Z \cup \{a\} \cup \{b\}$. Let F_1 be a junction forest with the same topology as F_0 except the above two cliques are modified into $X \cup Z \cup \{a\}$ and $Y \cup Z \cup \{b\}$. Let H_0 and H_1 be the entropies defined by the projected distributions of M on F_0 and F_1, respectively.*

If $Ind(X \cup \{a\}, Z, Y \cup \{b\})$ does not hold in M, but $Ind(X, Z \cup \{a\} \cup \{b\}, Y)$ does, then $H_0 < H_1$.

Proof:

Again, $Z \cup \{a\}$ is written as Za. Since $Ind(X, Zab, Y)$ is implied by F_0, we have $H_0 = H_M(XZab) + H_M(YZab) - H_M(Zab) + h$, where h is the entropy contribution from cliques other than the two mentioned in the proposition. Since $Ind(Xa, Z, Yb)$ is implied by F_1, we have $H_1 = H_M(XZa) + H_M(YZb) - H_M(Z) + h$. Because

$$\begin{aligned}
H_M(XZab) &= H_M(XZa) + H_M(b) - I_M(XZa;b) \\
H_M(YZab) &= H_M(YZb) + H_M(a) - I_M(YZb;a) \\
H_M(Zab) &= H_M(Z) + H_M(a) + H_M(b) - I_M(Za;b) - I_M(Z;a),
\end{aligned}$$

we obtain

$$\begin{aligned}
&H_1 - H_0 = [H_M(XZa) + H_M(YZb) - H_M(Z)] - [H_M(XZa) + H_M(b) - I_M(XZa;b) \\
&+H_M(YZb) + H_M(a) - I_M(YZb;a) - H_M(Z) - H_M(a) - H_M(b) + I_M(Za;b) + I_M(Z;a)] \\
&= (I_M(XZa;b) - I_M(Za;b)) + (I_M(YZb;a) - I_M(Z;a)).
\end{aligned}$$

By Lemma 3, we have $I_M(XZa;b) - I_M(Za;b) \geq 0$ and $I_M(YZb;a) - I_M(Z;a) \geq 0$, and therefore $H_1 - H_0 \geq 0$. The equality does not hold iff either $Ind(X, Za, b)$ does not hold or $Ind(Yb, Z, a)$ does not hold in M.

Using the contrapositive form of the following *contraction* condition [13] for probabilistic models $Ind(A, D, B) \& Ind(A, DB, E) \Longrightarrow Ind(A, D, BE)$ where A, B, D and E are disjoints sets, the assumption that $Ind(Xa, Z, Yb)$ does not hold in M implies that either $Ind(Yb, Z, a)$ does not hold or $Ind(Yb, Za, X)$ does not hold. $Ind(Yb, Za, X)$ does not hold implies that either $Ind(X, Za, b)$

does not hold or $Ind(X, Zab, Y)$ does not hold. Since $Ind(X, Zab, Y)$ does hold in M by assumption, that $Ind(Xa, Z, Yb)$ does not hold in M implies that either $Ind(Yb, Z, a)$ does not hold or $Ind(X, Za, b)$ does not hold in M. This completes the proof. □

Note that Proposition 4 is still valid if the two singleton sets $\{a\}$ and $\{b\}$ are replaced by general sets, though this generality is not needed in the paper.

Proposition 5 says that if a false independence relation can be removed by adding a single link, then the resultant MN structure must be chordal.

Proposition 5 *Let M be a probabilistic model over a set N of variables, (G, P) be a Markov network of M, and F be a junction forest of G. Let X, Y, Z, $\{a\}$ and $\{b\}$ be disjoint subsets of N such that at most one of $\{a\}$ and $\{b\}$ is empty if $Z = \phi$ and none is empty otherwise. Let $X \cup Z \cup \{a\}$ and $Y \cup Z \cup \{b\}$ be two adjacent cliques of F if $Z \neq \phi$.*

If $Ind(X \cup \{a\}, Z, Y \cup \{b\})$ does not hold in M, but $Ind(X, Z \cup \{a\} \cup \{b\}, Y)$ does, then a single link can always be added to G (resulting in a new graph G') such that $Ind(X, Z \cup \{a\} \cup \{b\}, Y)$ is implied by G', and G' is chordal.

Proof:

Suppose $Z = \phi$. Then cliques Xa and Yb are disconnected in F.

If $\{a\} \neq \phi$ and $\{b\} = \phi$, then it must be the case that $Y \neq \phi$. Otherwise, $Ind(Xa, Z, Yb) = Ind(Xa, \phi, \phi)$ is always valid, which contradicts the assumption that $Ind(Xa, Z, Yb)$ does not hold in M. If we form G' by connecting a to any node $y \in Y$, then G' implies $Ind(X, \{a\}, Y)$. Since the link (a, y) does not introduce any cycle, G' is chordal.

If $\{a\} \neq \phi$ and $\{b\} \neq \phi$, we can form G' by connecting a to b, and then G' implies $Ind(X, ab, Y)$. Since (a, b) does not introduce any cycle, G' is chordal.

Now suppose $Z \neq \phi$. Then cliques XZa and YZb are connected in F. Since Z is the sepset between the two cliques in F, the node a is connected to every node of Z in G, and so is b. Hence connecting a and b forms the clique Zab in G', which implies that the link (a, b) cannot create a cycle of length > 3 in G'. Therefore, G' is chordal. Since all paths from X to Y in G' are mediated through Zab, this implies $Ind(X, Zab, Y)$. □

Proposition 6 says that if the entropy of the current MN is not minimum, it is always possible to add a single link such that the resultant MN is chordal and the entropy is decreased.

Proposition 6 *Let M be a probabilistic model over a set N of variables. Let (G, P) be a Markov network of M. Let $H(N)$ be the entropy of N defined by P, and $H_M(N)$ defined by M. If $H(N) > H_M(N)$, there exists a link L not contained in G such that (1) the graph G' resulting from adding L to G is chordal; and (2) $H'(N) < H(N)$, where $H'(N)$ is the entropy of N defined by the projected distribution of M on G'.*

Proof:

Suppose $H(N) > H_M(N)$. By Proposition 2, an independence relation $Ind(X, Z, Y)$ that holds in P but not in M can be found.

For any disjoint sets X, Y and Z such that $Ind(X, Z, Y)$ is false in M, we can always find $A \subseteq X$ and $B \subseteq Y$ (at least one of A and B is nonempty) such that $Ind(X \setminus A, ZAB, Y \setminus B)$ is true. This is because $Ind(X \setminus X, ZXY, Y \setminus Y) = Ind(\phi, ZXY, \phi)$ is always true.

According to the *decomposition* condition of a PM [13], $Ind(X \setminus A, ZAB, Y \setminus B)$ implies $Ind(\phi, ZAB, \phi)$ which states that variables in ZAB are dependent on each other and thus ZAB must be a clique in any I-map of M. It follows that, for any pair of nodes $a \in A$ and $b \in B$, $Ind(\phi, Zab, \phi)$ holds in M.

Since a is not directly connected to Y in G and b is not directly connected to X, $Ind(\phi, Zab, \phi)$ does not hold in P. We therefore have found an independence relation $Ind(\phi, Zab, \phi)$ that does not hold in P but holds in M.

Denote the graph resulting from adding the link (a, b) to G by G'. Using Proposition 4, if we let $X = Y = \phi$, F_1 (F_0) be the JF of G (G'), $H(N)$ ($H'(N)$) be the entropy defined by the projected distribution of M on F_1 (F_0), then we have $H'(N) < H(N)$. By Proposition 5, it follows that G' is chordal. □

Proposition 7 claims that Algorithm 1 will never halt prematurely.

Proposition 7 *Let M be a probabilistic model over a set N of variables. Algorithm 1 halts and returns a Markov network (G, P) such that $H(N) = H_M(N)$, where $H(N)$ is the entropy defined by P, and $H_M(N)$ defined by M.*

Proof:

Algorithm 1 starts with a totally disconnected chordal graph G_0. If there exists a pair of nodes $x, y \in N$ ($x \neq y$) such that $Ind(x, \phi, y)$ does not hold in M, then $H_0(N) > H_M(N)$ by Proposition 4, where $H_0(N)$ is the entropy defined by the projected distribution of M on G_0. According to Proposition 6, a link can be added to decrease $H_0(N)$ to $H_1(N)$. If there is still a false independence relation in G_1, by applying Proposition 4 and Proposition 6, another link can be added to decrease $H_1(N)$ to $H_2(N)$.

By repeatedly applying Proposition 4 and Proposition 6, Algorithm 1 will capture more and more true dependencies. As soon as all the dependence relations in M are captured in G, Algorithm 1 halts and returns (G, P) such that $H(N) = H_M(N)$. There can be only finite number, say m, of links given a finite set of variables, and a completely connected graph G_m always satisfies $H_m(N) = H_M(N)$. Thus Algorithm 1 always halts with $H(N) = H_M(N)$. □

Let us now state the final result about Algorithm 1 by Theorem 8.

Theorem 8 *Let M be a probabilistic model over a set N of variables. Algorithm 1 halts and returns a Markov network (G, P) such that G is an I-map of M and $P = P_M$.*

Proof:

By Proposition 7, Algorithm 1 halts with $H(N) = H_M(N)$. The Kullback-Leibler cross entropy $K(P_M, P) = 0$ iff $P = P_M$. Since $K(P_M, P) = H(N) - H_M(N)$ (Section 3), $H(N) = H_M(N)$ implies $P = P_M$.

To show the I-mapness, suppose G is not an I-map of M. Then there exist three disjoint subsets X, Z and Y such that $< X|Z|Y >$ holds in G but $Ind(X, Z, Y)$ does not hold in M. $< X|Z|Y >$ in G implies that $Ind(X, Z, Y)$ holds in P. Hence $Ind(X, Z, Y)$ holds in P but not in M. According to Proposition 4, $H(N)$ can be further decreased below $H_M(N)$, which is a contradiction. □

To illustrate Proposition 2 through Theorem 8, we use the example PM M whose perfect map is depicted in Figure 1. Table 2 shows the false independence relation removed and the corresponding valid relation learned at each search step. The first column of Table 2 shows the search step index. The second column shows the false independence relation that holds in the current MN but not in M. The third column shows the independence relation that holds in M and is learned at the current step. The next five columns show the sets involved in the two independence relations. The last column shows the entropy after the current search step, which decreases as learning proceeds. We have used the simplified set notation in the table. Note that $Ind(\phi, il, \phi)$ implies that variables i and l are not conditionally independent of any set of variables.

Step	$Ind(Xa, Z, Yb)$	$Ind(X, Zab, Y)$	X	Y	Z	a	b	H
1	$Ind(i, \phi, l)$	$Ind(\phi, il, \phi)$	ϕ	ϕ	ϕ	i	l	2.3205
2	$Ind(l, \phi, k)$	$Ind(\phi, lk, \phi)$	ϕ	ϕ	ϕ	l	k	2.2757
3	$Ind(j, \phi, m)$	$Ind(\phi, jm, \phi)$	ϕ	ϕ	ϕ	j	m	2.2550
4	$Ind(k, \phi, l)$	$Ind(\phi, kl, \phi)$	ϕ	ϕ	ϕ	k	l	2.2480
5	$Ind(i, m, j)$	$Ind(\phi, imj, \phi)$	ϕ	ϕ	m	i	j	2.2436
6	$Ind(j, i, l)$	$Ind(\phi, jil, \phi)$	ϕ	ϕ	i	j	l	2.2330
7	$Ind(i, l, k)$	$Ind(\phi, ilk, \phi)$	ϕ	ϕ	l	i	k	2.2317
8	$Ind(j, il, k)$	$Ind(\phi, ijkl, \phi)$	ϕ	ϕ	il	j	k	2.2306
9	$Ind(k, ij, m)$	$Ind(\phi, ijkm, \phi)$	ϕ	ϕ	ij	k	m	2.2278

Table 2: False independence relation removed at each search step.

5 Learning a Minimal I-map

In general, Algorithm 1 does not necessarily produce a *minimal* I-map . Figure 2 (left) shows the perfect map of a PM M. The two marginal distributions of M are listed in the upper part of Table 3. Algorithm 1 learns the perfect map correctly. However, if we weaken the strength of dependency between b and d (b influences d with only one eighth the strength as before when $c = c_1$) by changing two items of the distribution as shown in the lower part of Table 3, Algorithm 1 learns the structure depicted in Figure 2 (right), which is not a minimal I-map of M. From our experience, such situation only occurs when dependence relations between some variables are very weak.

Since the problem of learning a minimal I-map is NP-hard [1], it is not surprising that the $O(n^4)$ time algorithm does not produce a minimal I-map in every case. However, we would like to know under what conditions Algorithm 1 does return a minimal I-map.

Proposition 9 says that, if variables a and b are dependent, and a and c are either marginally independent or independent given b, then Algorithm 1 will prefer to connect a and b rather than a and c.

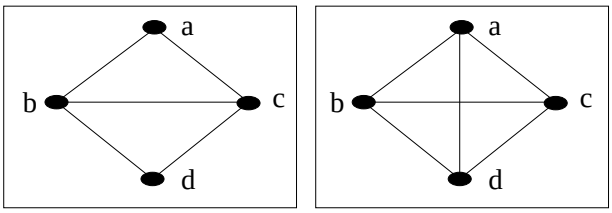

Figure 2: Comparison of two Markov networks.

$p(a_0$	&	b_0	&	$c_0)$	$= 0.0225$	$p(b_0$	&	c_0	&	$d_0)$	$= 0.0445$
$p(a_0$	&	b_0	&	$c_1)$	$= 0.0675$	$p(b_0$	&	c_0	&	$d_1)$	$= 0.0049$
$p(a_0$	&	b_1	&	$c_0)$	$= 0.0025$	$p(b_0$	&	c_1	&	$d_0)$	$= 0.1764$
$p(a_0$	&	b_1	&	$c_1)$	$= 0.0075$	$p(b_0$	&	c_1	&	$d_1)$	$= 0.0441$
$p(a_1$	&	b_0	&	$c_0)$	$= 0.0270$	$p(b_1$	&	c_0	&	$d_0)$	$= 0.0773$
$p(a_1$	&	b_0	&	$c_1)$	$= 0.1530$	$p(b_1$	&	c_0	&	$d_1)$	$= 0.0331$
$p(a_1$	&	b_1	&	$c_0)$	$= 0.1080$	$p(b_1$	&	c_1	&	$d_0)$	$= 0.0929$
$p(a_1$	&	b_1	&	$c_1)$	$= 0.6120$	$p(b_1$	&	c_1	&	$d_1)$	$= 0.5265$
*	**	**	**	**	**	**	**	**	**	**	**
$p(b_0$	&	c_1	&	$d_0)$	$= 0.02205$	$p(b_0$	&	c_1	&	$d_1)$	$= 0.19845$

Table 3: *Upper*: Marginal distributions for the two cliques in Figure 2 (left). *Lower*: Modified probabilities for the lower clique in Figure 2 (left).

Proposition 9 *Let M be a probabilistic model over a set N of variables. Let $a, b, c \in N$ be distinct such that $Ind(a, \phi, b)$ does not hold in M. Let $(G = (N, E), P)$ be a Markov network such that the component of G that contains a contains neither b nor c. Let $G_1 = (N, E \cup \{(a, b)\})$ and $G_2 = (N, E \cup \{(a, c)\})$. Let (G_1, P_1) and (G_2, P_2) be Markov networks of M, and $H_1(N)$ and $H_2(N)$ be entropies defined by P_1 and P_2, respectively.*

Then, we have $H_1(N) < H_2(N)$ if (1) $Ind(a, \phi, c)$ holds in M, or (2) $Ind(a, b, c)$ holds in M but $Ind(a, c, b)$ does not.

Proof:

That a and b are in different components implies that the addition of link (a, b) will form a new clique in G^1 that contains only a and b. Hence, $H_1(N) = H(N) + H_M(ab) - H_M(a) - H_M(b)$, where $H(N)$ is the entropy defined by P, and $H_M(ab)$ is the entropy of the new clique defined by M.

Similarly, that a and c are in different components implies that the addition of link (a, c) will form a new clique in G_2 that contains only a and c. Therefore, $H_2(N) = H(N) + H_M(ac) - H_M(a) - H_M(c)$. We have

$$\begin{aligned} &H_2(N) - H_1(N) = H_M(ac) - H_M(c) - H_M(ab) + H_M(b) \\ &= [H_M(a) + H_M(c) - I_M(a; c)] - H_M(c) - [H_M(a) + H_M(b) - I_M(a; b)] + H_M(b) \\ &= I_M(a; b) - I_M(a; c). \end{aligned}$$

If $Ind(a, \phi, c)$ holds in M, then $I_M(a; c) = 0$. Since $Ind(a, \phi, b)$ does not hold in M, we have $H_2(N) - H_1(N) = I_M(a; b) > 0$.

On the other hand, if $Ind(a, b, c)$ holds in M, then

$$I_M(a; b) - I_M(a; c) = [I_M(a; c) + I_M(a; b|c)] - I_M(a; c) = I_M(a; b|c) >= 0,$$

with equality iff $Ind(a, c, b)$ holds in M. Since $Ind(a, c, b)$ does not hold by assumption, $I_M(a; b) - I_M(a; c) > 0$. □

Lemma 10 *Let M be a probabilistic model with a minimal I-map $G_M = (N, E_M)$ such that G_M is a tree, and let P_M be strictly positive. Then, for any three distinct variables $a, b, c \in N$, $Ind(a, b, c)$ and $Ind(a, c, b)$ can not both be true in M.*

Proof:

$Ind(a, b, c)$ implies $P_M(a|bc) = P_M(a|b)$ and $Ind(a, c, b)$ implies $P_M(a|bc) = P_M(a|c)$. In order for $P_M(a|bc) = P_M(a|c) = P_M(a|b) > 0$ to hold for all possible combinations of values of b and c, it must be the case that $P_M(a|bc) = P_M(a)$. This means that a is marginally independent of b and c in M. Since G_M is a minimal I-map of M, a must be disconnected from b and c in G_M, which contradicts to the fact that G_M is a tree. □

Theorem 11 *Let M be a probabilistic model with a minimal I-map G_M such that G_M is singly connected (no more than one path between any pair of variables), and let P_M be strictly positive. Then Algorithm 1 returns a Markov network whose structure is G_M.*

Proof:

Because G_M is singly connected, for any two variables a and c in M, either $Ind(a, \phi, c)$ or $Ind(\phi, ac, \phi)$ or $Ind(a, b, c)$ where b is a distinct variable on the path between a and c in G_M.

Denote $G_M = (N, E_M)$. Algorithm 1 starts with $G = (N, \phi)$. Suppose (a, c) is selected as the first link to be added and $(a, c) \notin E_M$. Then either $Ind(a, \phi, c)$ or there exists a $b \in N$ such that $Ind(a, b, c)$. According to Lemma 10, in the second case, $Ind(a, c, b)$ must be false in M.

Let the MN resulting from adding (a, b) to G be (G_1, P_1) and the entropy defined by P_1 be $H_1(N)$. Let the MN resulting from adding (a, c) to G be (G_2, P_2) and the entropy defined by P_2 be $H_2(N)$. By Proposition 9, we have $H_1(N) < H_2(N)$. Thus (a, c) could not have been selected by Algorithm 1. We therefore conclude that the first link added must be in G_M.

Suppose the current graph $G = (N, E)$ satisfies $E \subset E_M$. We claim that, if the next link added by Algorithm 1 is (a, b), a and b must be in different components of G. Suppose a and b are in the same component of G. Since G_M is singly connected and $E \subset E_M$, adding (a, b) to G (resulting in $G_1 = (N, E \cup \{(a, b)\})$) must introduce a loop, which implies $(a, b) \notin E_M$. Since G is singly connected but G_1 is multiply connected and is chordal required by Algorithm 1, there must be a variable c in G such that $\{a, b, c\}$ is a clique in G_1. Since G is a subgraph of G_M and $(a, b) \notin E_M$, it must be true that $Ind(a, c, b)$ holds in M. This implies $H_1(abc) = H(ac) + H(bc) - H(c)$, where H_1 (H) is the entropy computed based on G_1 (G). This in turn implies that $H_1 = H$. Therefore, the link (a, b) could not have been selected.

Now, we only have to show that, if (a, b) is added next by Algorithm 1 and it connects two components of G, it must be the case $(a, b) \in E_M$. Otherwise,

either $Ind(a, \phi, b)$ holds in M, in which case a and b are disconnected in G_M, or there exists c such that $Ind(a, c, b)$ holds in M and $Ind(a, b, c)$ does not hold in M, in which case a and b are indirectly connected in G_M.

If $Ind(a, c, b)$ holds in M, let the MN resulting from adding (a, b) to G be (G_1, P_1), and the entropy defined by P_1 be $H_1(N)$. Let the MN resulting from adding (a, c) to G be (G_2, P_2), and the entropy defined by P_2 be $H_2(N)$. By Proposition 9, we have $H_2(N) < H_1(N)$. Hence, the link (a, b) could not have been selected by the algorithm.

If $Ind(a, \phi, b)$ holds in M, connecting a and b modifies the entropy by an amount $H_1(ab) - H_1(a) - H_1(b) = 0$, and therefore (a, b) could not have been selected by the algorithm. □

Since Algorithm 1 adds no superfluous links when learning a singly connected PM, and a singly connected graph of n nodes can have no more than $n - 1$ links, we can lower the computational complexity of the algorithm.

Corollary 12 *Let M be a PM with a minimal I-map G_M such that G_M is singly connected, and let P_M be strictly positive. Then Algorithm 1 returns a MN with structure G_M in $O(n^3)$ time.*

6 Related Work

Comparison with Chow and Liu

Chow and Liu [2] developed an $O(n^2)$ algorithm to approximate optimally a jpd by a projected distribution on a tree (connected and singly connected) for the purpose of storage savings. However, presuming a tree structure is inadequate for learning a PM in an arbitrary domain.

Our method uses the same Kullback-Leibler cross-entropy as a measure of closeness as Chow and Liu. Algorithm 1, however, can learn a disconnected and multiply connected PM. The price is the increased complexity $O(n^4)$. When the underlying PM is indeed a tree, Algorithm 1 produces the correct model in $O(n^3)$ time as stated in Corollary 12.

Comparison with Ku and Kullback

The method suggested by Ku and Kullback [8] allows any lower-order marginal distributions to be used in approximating a jpd with a convergent iterative procedure. The resultant jpd is more accurate than that obtained by the method of Chow and Liu. However, their method does not learn the dependency structure. Furthermore, since the probability of each configuration of N must be estimated at each iteration, the method requires execution time exponential in the cardinality of N.

Comparison with methods for learning a Bayesian network

Pearl [13] showed that directionality makes Bayesian networks (BNs) a richer language in expressing dependencies. For instance, an induced dependency can be expressed by a BN but not by a MN. Contrary to the methods that learn a BN [6, 3, 15, 13], our method does not discover directions of dependencies. On the other hand, one important application of BNs is to compute posterior

marginal probabilities. An elegant algorithm for doing that in a multiply connected BN is the junction tree method by Jensen et al.[7] whose underlying run time representation is a MN (in terms of its JT). In converting the original BN into a MN and then a JT, directionality is discarded. Therefore, as long as computing posterior marginal probabilities is concerned, a MN is equally expressive as a BN.

Jensen's method can be extended to probabilistic inference with multiply sectioned Bayesian networks in a single agent oriented system [20, 19] as well as a multiagent distributed interpretation system [18]. The run time representation is a set of MNs (in terms of a set of JTs). These approaches also highlight the usefulness of the MN representation.

It has been shown [17] 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 MN. This implies that once a PM is expressed in terms of a MN, probabilistic reasoning can be easily performed using standard relational DBMSs.

Comparison with Kutato

The procedure Kutato developed by Herskovits and Cooper [6] bears much resemblance to Algorithm 1. Kutato learns a BN. In contrast, Algorithm 1 learns a MN. Both employ greedy search, and have similar time complexity.

However, Algorithm 1 is based on the minimization of Kullback-Leibler cross-entropy. As claimed, "Kutato is an efficient system for approximating the *maximum-entropy* distribution of a database" [6]. The I-mapness of Kutato has not been formally established. We believe that it can be achieved with an analysis similar to what we have presented in this paper.

Comparison with learning classification rules or decision trees

Many machine learning systems [14, 11, 12, 21] aim at generating explicit classification rules or decision trees. Generation of rules or trees from observations can be more efficient than learning domain models, but the applicability of the results are also limited. Decision rules or trees are too rigid with respect to the decision variables as they are fixed once the rules or trees are generated. Decision rules and trees are also rigid in the number of variables whose values must be observed before a rule can be fired or a terminal node in a tree can be reached.

Systems that learn a MN or a BN, on the other hand, are more flexible. A MN or a BN captures the domain independence relations without being committed a priori to particular decision variables and particular patterns of observations.

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

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

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

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

[6] 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.

[7] 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.

[8] H.H. Ku and S. Kullback. Approximating discrete probability distributions. *IEEE Trans. Information Theory*, IT-15(4):444–447, 1969.

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

[10] 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.

[11] R.S. Michalski and R.J. Chilausky. Learning by being told and learning from examples. *Journal of Policy Analysis and Information Systems*, (4), 1980.

[12] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, (5):341–356, 1982.

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

[14] J.R. Quinlan. Learning efficient classification procedures and their application to chess end games. In *Machine Learning: An Artificial Intelligence Approach, Vol.1*, pages 463–482. Morgan Kaufmann, 1983.

[15] S. Srinivas, S. Russell, and A. Agogino. Automated construction of sparse bayesian networks for unstructured probabilistic models and domain information. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 295–308. North-Holland, 1990.

[16] 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.

[17] 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, July 1994.

[18] 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, Oct. 1994.

[19] 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.

[20] 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.

[21] W. Ziarko. The discovery, analysis, and representation of data dependencies in database. In G. Piatetsky-Sapiro and W.J. Frawley, editors, *Knowledge Discovery in Databases*, pages 195–209. AAAI/MIT, 1991.