Exact algorithms for MRE Inference

Xiaoyuan Zhu  XIAOYUAN.ZHU@QC.CUNY.EDU
Changhe Yuan  CHANGHE.YUAN@QC.CUNY.EDU

Queens College, City University of New York
65-30 Kissena Blvd., Queens, NY 11367

Abstract

Most Relevant Explanation (MRE) is an inference task in Bayesian networks that finds the most relevant partial instantiation of target variables as an explanation for given evidence by maximizing the Generalized Bayes Factor (GBF). No exact algorithm has been developed for solving MRE previously. This paper fills the void and introduces Breadth-First Branch-and-Bound (BFBnB) MRE algorithms based on two novel upper bounds on GBF. One bound is calculated by decomposing the computation of the score to a set of Markov blankets of subsets of evidence variables. Furthermore, another improved Markov blanket bound is proposed based on two new ideas. One is to split the Markov blankets that are too large by converting auxiliary nodes into pseudo-targets. The other is to perform summation instead of maximization on some of the targets in each Markov blanket. Our empirical evaluations show that the proposed BFBnB algorithms make exact MRE inference tractable in Bayesian networks that could not be solved previously.

1. Introduction

Bayesian networks are probabilistic models that capture the conditional independences between random variables as directed acyclic graphs, and provide principled approaches to explaining evidence, e.g., Most Probable Explanation (MPE), Maximum a Posterior (MAP), and Most Relevant Explanation (MRE). MPE (Pearl, 1988) finds the most likely instantiation of all the unobserved (target) variables as an explanation for the evidence. MAP (Pearl, 1988; Darwiche, 2009; Koller & Friedman, 2009) is a generalization of MPE by including auxiliary variables besides target and evidence variables. Both MPE and MAP are solved by optimizing the joint posterior probability of the target variables given the evidence. However, in explanation problems (Lacave & Díez, 2002; Nielsen, Pellet, & Elisseeff, 2008), MPE and MAP often produce overspecified explanations (Pacer, Lombrozo, Griffiths, Williams, & Chen, 2013; Yuan, Liu, Lu, & Lim, 2009), i.e., irrelevant target variables may be included. This is because maximizing the joint probability alone cannot exclude irrelevant target variables directly. To address the limitation of MPE and MAP, MRE (Yuan & Lu, 2007; Yuan et al., 2009) was proposed to find a partial instantiation of the target variables that maximizes the Generalized Bayes Factor (GBF) (Fitelson, 2007; Good, 1985) as an explanation for the evidence. GBF is a rational function of probabilities that is suitable for comparing explanations with different cardinalities. Theoretically, MRE is shown able to prune away independent and less relevant variables from the final explanation (Yuan, Lim, & Lu, 2011b); a recent human study also provided empirical evidence to support the claim (Pacer et al., 2013).

Due to the difficulty of finding a meaningful bound for GBF, no exact algorithms have been developed to solve MRE. Only local search and Markov chain Monte Carlo methods have been proposed previously (Yuan, Lim, & Littman, 2011a). In this paper, we are the first to introduce non-trivial exact MRE algorithms, i.e., Breadth-First Branch-and-Bound (BFBnB) algorithms. The
The key idea of the proposed methods is to decompose the whole Bayesian network into a set of overlapping subnetworks. Each subnetwork is characterized by a subset of evidence variables and its Markov blanket. An upper bound for the GBF of an MRE solution is derived by solving independent optimization problems in the subnetworks. To tighten the upper bound, we introduce a greedy algorithm for merging the Markov blankets that share target variables. Then, we further propose another improved upper bound based on two new ideas. First, the decomposition method often leads to large Markov blankets which prevent the application of branch-and-bound algorithm on large scale MRE problems. To address this problem, we propose to split large Markov blankets by converting auxiliary nodes into pseudo-targets. Second, we find that the upper bound can be tightened by identifying and summing out enclosed-targets from each Markov blanket, which does not change the decomposition of the whole network.

The proposed upper bounds are used in the BFBnB algorithms for pruning the search space. We evaluate the proposed algorithms in a set of benchmark diagnostic Bayesian networks. The experimental results show that the proposed BFBnB algorithms make exact MRE inference tractable in Bayesian networks that could not be solved previously.

2. Background

2.1 Bayesian networks and moral graph

A Bayesian network (Pearl, 1988; Darwiche, 2009; Koller & Friedman, 2009) is represented as a Directed Acyclic Graph (DAG). The nodes in the DAG represent random variables. The lack of arcs in the DAG define conditional independence relations among the nodes. If there is an arc from node \(Y\) to \(X\), we say that \(Y\) is a parent of \(X\), and \(X\) is a child of \(Y\) (We use upper-case letters to denote variables \(X\) or variable sets \(X\), and lower-case letters for values of scalars \(x\) or vectors \(x\)). A node \(Y\) is an ancestor of a node \(X\), if there is a directed path from \(Y\) to \(X\). Let \(an(X)\) denote all the ancestors of \(X\), then the smallest ancestral set \(An(X)\) of node set \(X\) is defined as \(An(X) = X \cup \bigcup_{X_i \in X} an(X_i)\). In directed graphs, \(d\)-separation describes the conditional independence relation between two set of nodes \(X\) and \(Y\), given a third set of nodes \(Z\), i.e., \(p(X|Z, Y) = p(X|Z)\). The Markov blanket of \(X\), denoted by \(MB(X)\), is the smallest node set which \(d\)-separates \(X\) from the remaining nodes in the network. The network as a whole represents the joint probability distribution of \(\prod_X p(X|PA(X))\), where \(PA(X)\) is the set of all the parents of \(X\).

The moral graph \(G^m\) of a DAG \(G\) is an undirected graph with the same set of nodes. There is an edge between \(X\) and \(Y\) in \(G^m\) if and only if there is an edge between them in \(G\) or if they are parents of the same node in \(G\). In an undirected graph, \(Z\) separates \(X\) and \(Y\), if \(Z\) intercepts all paths between \(X\) and \(Y\). Moral graph is a powerful construction to explain \(d\)-separation. Lemma 1 (Lauritzen, Dawid, Larsen, & Leimer., 1990) links \(d\)-separation in DAG to separation in undirected graphs.

**Lemma 1.** Let \(X\), \(Y\), and \(Z\) be disjoint subsets of nodes in a DAG \(G\). Then \(Z\) \(d\)-separates \(X\) from \(Y\) if and only if \(Z\) separates \(X\) from \(Y\) in \((G^m_{An(X \cup Y \cup Z)})\), where \((G^m_{An(X \cup Y \cup Z)})\) is the moral graph of the subgraph of \(G\) with node set \(An(X \cup Y \cup Z)\).

2.2 Most relevant explanation

In an inference problem in a Bayesian network, the nodes are often classified into three categories: target, evidence, and auxiliary. The target set \(M\) represents variables of inference interest. The
Evidence set $E$ represents observed information. The auxiliary set represents the variables that are not of interest in the inference. MRE (Yuan et al., 2011b), which finds a partial instantiation of $M$ as an explanation for given evidence $e$ in a Bayesian network, is formally defined as follows.

**Definition 1.** Let $M$ be a set of targets, and $e$ be the given evidence in a Bayesian network. Most Relevant Explanation is the problem of finding a partial instantiation $x$ of $M$ that has the maximum generalized Bayes factor score $GBF(x; e)$ as the explanation for $e$, i.e.,

$$MRE(M; e) = \arg\max_{x, \emptyset \subset X \subseteq M} GBF(x; e),$$

where $GBF$ is defined as

$$GBF(x; e) = \frac{p(e|x)}{p(e|\overline{x})}. \quad (2)$$

In the above definition, $x$ is the instantiation of $X$, and $\overline{x}$ represents all of the alternative explanations of $x$. Here, explanation refers to the explanation of evidence, whose goal is to explain why some observed variables are in their particular states using the target variables in the domain. In model based explanation problems, one commonly used measure is the probability of an explanation given the evidence, as used in MAP and MPE to find the most likely configuration of a set of target variables. Methods that use probability as the relevance measure do not have the intrinsic capability to prune the less relevant facts. Moreover, the probability measure is quite sensitive to the modeling choices, e.g., simply refining a model can dramatically change the best explanation. Different from MPE and MAP, MRE maximizes the rational function of probabilities $GBF$ in Equation 2. This makes it possible for MRE to compare explanations with different cardinalities and to prune the less relevant variables automatically in a principled way. The search space of MRE is exponential on the number of targets, and the complexity of MRE is conjectured to $NP^{PP}$ complete (Yuan et al., 2011a).

To further study the properties of generalized Bayes factor, we reformulate $GBF$ as follows.

$$GBF(x; e) = \frac{p(e|x)}{p(e|\overline{x})} = \frac{p(x|e)p(\overline{x})}{p(x)p(\overline{x}|e)} = \frac{p(x|e)(1 - p(x))}{p(x)(1 - p(x|e))}. \quad (3)$$

Based on the above formulation, $GBF$ is no longer a measure for comparing different hypotheses, but simply a measure that compares the posterior and prior probabilities of a single hypothesis. $GBF$ is hence able to overcome the drawback of Bayes factor (Jeffreys, 1961) in having to do pairwise comparisons between multiple hypotheses.

Belief update ratio is a useful concept. The belief update ratio of $X$ given $e$ is defined as follows.

$$r(X; e) = \frac{p(X|e)}{p(X)}. \quad (4)$$

$GBF$ can be calculated from the belief update ratio as follows.

$$GBF(x; e) = \frac{p(x|e)(1 - p(x))}{p(x)(1 - p(x|e))} = \frac{r(x; e) - p(x|e)}{1 - p(x|e)} = 1 + \frac{r(x; e) - 1}{1 - p(x|e)}. \quad (5)$$
Figure 1: The search tree of an MRE problem with three targets, i.e., A, B, and C.

2.3 Existing methods for solving MRE

Many local search methods have been applied to solve both MAP (MPE) (Park & Darwiche, 2001) and MRE (Yuan et al., 2011a), such as tabu search (Glover, 1990). To solve MRE, tabu search starts at an empty solution set. At each step, it generates the neighbors of the current solution by adding, changing, or deleting one target variable. Then tabu search selects the best neighbor which has the highest GBF score and has not been visited before. In tabu search, the best neighbor can be worse than the current solution. To stop tabu search properly, upper bounds are set on both the total number of search steps $M$ and the number of search steps since the last improvement $L$ as the stopping criteria. The local search methods can only provide approximate solutions which are not guaranteed to be optimal. Furthermore, the accuracy and efficiency of these methods are typically highly sensitive to tunable parameters.

2.4 Branch and bound-based inference algorithms

Branch-and-bound algorithms have been developed for solving MAP and MPE by using upper bounds derived based on the property of optimization criterion or the structure of Bayesian networks. A mini-bucket upper bound is proposed in (Dechter & Rish, 2003) and applied to And/Or tree search for solving MPE (Marinescu & Dechter, 2009). The work in (Choi, Chavira, & Darwiche, 2007) showed that the mini-bucket upper bound can be derived from a node splitting scheme. To solve MAP exactly, an upper bound is proposed in (Park & Darwiche, 2003) by commuting the order of max and sum operations in the MAP calculation. In (Huang, Chavira, & Darwiche, 2006), an exact algorithm is proposed for solving MAP by computing upper bounds on an arithmetic circuit compiled from a Bayesian network. No upper bound or exact algorithm has been proposed for solving MRE, however.

3. A Novel Markov Blanket Upper Bound for Solving MRE

It is difficult to solve MRE problems exactly because of both an exponential search space and the need for probabilistic inference at each search step. Generally, a naive brute-force search method can scale to Bayesian networks with at most 15 targets. In this work, we propose breadth-first branch-and-bound algorithms that use a suite of new upper bounds based on Markov blanket decomposition to prune the search space. The algorithm makes it possible to solve MRE problems with more targets exactly.
3.1 Search space formulation

Assuming there are \( n \) targets, and each target has \( d \) states, the search space of MRE contains \((d + 1)^n - 1\) possible joint states (or solutions). Each state contains values of a subset of the targets. We organize the search space as a search tree by instantiating the targets according to a total order \( \pi \) of the targets. The search tree has the empty state as the root. For a state \( S \) in the tree, \( V \) is defined as the expanded set of targets, and \( U = \{ U_i | U_i \in M; \forall V_j \in V; V_j \prec \pi U_i \} \) is the unexpanded set. Each child state of \( S \) instantiates one more unexpanded target. Figure 1 shows an example with three targets \( A = \{a, \bar{a}\}, B = \{b, \bar{b}\}, \) and \( C = \{c, \bar{c}\} \) in this particular order. Assuming the current state is \( \{b\} \), \( \{B\} \) is the expanded set of target(s), and \( \{C\} \) is the unexpanded set. Different branches of the search tree may have different numbers of layers, but all of the states with the same cardinality appear in the same layer.

It is possible to use dynamic ordering to expand the targets that can most improve the GBF score first. However, it was shown in (Yuan & Hansen, 2009) that a static ordering can actually make computing upper bounds and ultimately the search more efficient. We therefore simply ordered the targets according to their indices in this work.

3.2 An upper bound based on Markov blanket decomposition

For MRE inference, an upper bound of a state \( S \) should be greater than the GBF score of all the descendant states of \( S \). During search, we can keep track of the highest-scoring state and prune the whole subtree if the upper bound is less than the GBF of the current best solution.

We introduce the following novel upper bound for MRE inference. We first partition the evidence variables into a set of exclusive and exhaustive subsets, i.e., \( E = \bigcup_i E_i \), by using depth first search discussed later. These evidence subsets naturally decompose the whole network into overlapping subnetworks. Each of these subnetworks contains an evidence subset \( E_i \) and its target-Markov blanket \( MB(E_i) \) which is defined as follows.

**Definition 2.** The target-Markov blanket \( MB(E_i) \) is the set of target variables which d-separates \( E_i \) from the remaining targets and evidence variables in the network.

For simplicity, throughout this paper we use Markov blanket instead of target-Markov blanket to refer to \( MB(E_i) \). An upper bound on GBF is then derived by multiplying the upper bounds on the belief update ratios calculated on the subnetworks.

We first derive an upper bound on the belief update ratio in Theorem 1.

**Theorem 1.** Let \( M = \{X_1, X_2, \ldots, X_n\} \) be a set of targets, \( e \) be the evidence, and \( MB(E_i) \) be the Markov blanket of the \( i^{th} \) subset of evidence \( E_i \). Then, for any subset \( \emptyset \subseteq X \subseteq M \), the belief update ratio \( r(x; e) \) is upper bounded as follows.

\[
\max_{x, \emptyset \subseteq X \subseteq M} r(x; e) \leq \left( \prod_i \max_{X = MB(E_i)} r(x; e_i) \right) \cdot C, \tag{6}
\]

where \( C = \prod_i p(e_i)/p(e) \).
Proof. From the formulation of \( r(M; e) \), we have
\[
    r(M; e) = \frac{p(M|e)}{p(M)} = \frac{p(e|M)}{p(e)}
\]
\[
    = \prod_i p(e_i|MB(E_i))/p(e)
\]
\[
    = \prod_i \frac{p(MB(E_i)|e_i)p(e_i)}{p(MB(E_i))}/p(e)
\]
\[
    = \left( \prod_i r(MB(E_i); e_i)p(e_i) \right)/p(e).
\]
(7)

The third equality is based on the property of Markov blankets.

Thus, we have:
\[
    r(M; e) = \left( \prod_i r(MB(E_i); e_i) \right) \cdot C,
\]
(8)

where \( C = \prod_i p(e_i)/p(e) \).

From Equation 8, we have the following upper bound on \( r(m; e) \),
\[
    \max_m r(m; e) \leq \left( \prod_i \max_{x, X = MB(E_i)} r(x; e_i) \right) \cdot C.
\]
(9)

For any \( \emptyset \subset X \subset M, M = X \cup \overline{X} \), let \( S_i = MB(E_i) \cap \overline{X}, \overline{S}_i = MB(E_i) \cap X, I = \{ i : S_i \neq \emptyset \} \), and \( J = \{ i : S_i = \emptyset \} \), we have:
\[
    p(e|X) = \sum_{\overline{X}} p(e|M)p(\overline{X}|X)
\]
\[
    = \sum_{\overline{X}} \left( \prod_{i \in I} p(e_i|MB(E_i)) \right) p(\overline{X}|X) \cdot \prod_{i \in J} p(e_i|MB(E_i))
\]
\[
    \leq \left( \prod_{i \in I} \max_{s_i} p(e_i|\overline{S}_i \cup s_i) \right) \cdot \left( \prod_{i \in J} p(e_i|MB(E_i)) \right).
\]
(10)

Since \( p(e|X) = r(X; e)p(e) \), we have the following upper bound on \( r(x; e) \):
\[
    \max_x r(x; e) \leq \left( \prod_{i \in I} \max_{s_i} \max_{\overline{s}_i} r(s_i \cup \overline{s}_i; e_i) \right) \cdot \left( \prod_{i \in J} \max_{x, X = MB(E_i)} r(x; e_i) \right) \cdot C.
\]
(11)

Thus, for any \( \emptyset \subset X \subset M \), we obtain the final upper bound by combining Equations 9 and 11.
\[
    \max_{x, \emptyset \subset X \subset M} r(x; e) \leq \left( \prod_{i} \max_{x, X = MB(E_i)} r(x; e_i) \right) \cdot C.
\]
\(\square\)
In MRE inference, the evidence \( e \) is given, thus \( C \) is a constant. Theorem 1 assumes that the expanded target set \( V = \emptyset \), which is true at the beginning of search. During the search when \( V \neq \emptyset \), we have the following corollary to derive the upper bound on the belief update ratio.

**Corollary 2.** Let \( M = \{X_1, X_2, \ldots, X_n\} \) be a set of targets, \( e \) be the evidence, \( MB(E_i) \) be the Markov blanket of the \( i \)th subset of evidence \( E_i \). Let \( U \) and \( V \) be the unexpanded and expanded target sets of state \( S \). Let \( T_i = MB(E_i) \cap V \), and \( \bar{T}_i = MB(E_i) \setminus T_i \). Then, for any subset \( \emptyset \subset X \subseteq U \), the belief update ratio \( r(x \cup v; e) \) is upper bounded as follows.

\[
\max_{x, \emptyset \subset X \subseteq U} r(x \cup v; e) \leq \left( \prod_{i} \max_{z, Z = T_i} r(z \cup t_i; e_i) \right) \cdot C, \tag{12}
\]

where \( C = \prod_i p(e_i)/p(e) \).

Based on Corollary 2, we can derive the upper bound on GBF in Theorem 3:

**Theorem 3.** Let \( M = \{X_1, X_2, \ldots, X_n\} \) be a set of targets, \( e \) be the evidence, \( MB(E_i) \) be the Markov blanket of the \( i \)th subset of evidence \( E_i \). Let \( U \) and \( V \) be the unexpanded and expanded target sets of state \( S \). Let \( T_i = MB(E_i) \cap V \), and \( \bar{T}_i = MB(E_i) \setminus T_i \). Then, for any subset \( \emptyset \subset X \subseteq U \), the generalized Bayesian factor score \( GBF(x \cup v; e) \) is upper bounded as follows.

\[
\max_{x, \emptyset \subset X \subseteq U} GBF(x \cup v; e) \leq 1 + \frac{\prod_i \max_{z, Z = T_i} r(z \cup t_i; e_i) \cdot C - 1}{1 - p(v|e)}, \tag{13}
\]

where \( C = \prod_i p(e_i)/p(e) \).

**Proof.** First, we formulate GBF using the belief update ratio as in Equation 5.

\[
GBF(m; e) = 1 + \frac{r(m; e) - 1}{1 - p(m|e)}.
\]
For any subset \( \emptyset \subset X \subseteq U \),
\[
p(x \cup v|e) = p(x|v, e) \cdot p(v|e) \leq p(v|e).
\]
Thus we have
\[
\max_{x, \emptyset \subset X \subseteq U} GBF(x \cup v; e) \leq 1 + \frac{\max_{x, \emptyset \subset X \subseteq U} r(x \cup v; e) - 1}{1 - p(v|e)}.
\]
(14)

Then using Corollary 2, we obtain the following upper bound on GBF
\[
\max_{x, \emptyset \subset X \subseteq U} GBF(x \cup v; e) \leq 1 + \frac{\left( \prod_{i} \max_{z, z_{i} = T_{i}} r(z \cup t_{i}; e_{i}) \right) \cdot C - 1}{1 - p(v|e)},
\]
where \( C = \prod_{i} p(e_{i}) / p(e) \).

Using Equation 13, we can bound all the descendant solutions of the current state \( S \). Equation 13 shows that in an MRE problem (left), we need to search all of the subsets of targets \( M \) to find the best solution. However, to calculate an upper bound (right), we only need to search a fixed target set \( MB(E_{i}) \) of each subnetwork, which usually has a small size and is easy to handle.

3.3 Compiling Markov blankets

The above theorems are based on factorizing the conditional joint distribution \( p(e|M) \) into the product of a set of conditional distributions \( \prod_{i} p(e_{i}|MB(E_{i})) \). Thus partitioning the whole evidence set \( E \) into the subsets \( \bigcup_{i} E_{i} \) and compiling their Markov blankets is a key part of the proposed methods. The Markov blanket of a single node includes its parents, children, and children’s other parents. In MRE problems, some of these nodes may be auxiliary nodes. To derive the upper bound, however, we need to find the Markov blankets containing only targets. Thus the standard definition of Markov blanket cannot be directly used in our problem.

In the proposed methods, we first generate the smallest ancestral set containing the target set \( M \) and the evidence set \( E \), i.e., \( An(M \cup E) \). Then we compile a moral graph \( (G_{An(M \cup E)})^{m} \). Figure 2(b) illustrates an example moral graph with two evidence nodes. Using Lemma 1, for each evidence subset \( E_{i} \), we need to find a set of targets which separates \( E_{i} \) from other targets and evidence nodes in \( (G_{An(M \cup E)})^{m} \). This is achieved by doing a depth first graph traversal starting from each evidence node if it has not been visited before. There are three scenarios when a node is being visited.

**Case 1:** When an evidence node is visited, we add the evidence to the current evidence subset \( E_{i} \), mark it as visited, and continue to visit its unmarked neighbors.

**Case 2:** When a target is visited, we add the target to \( MB(E_{i}) \) and mark it as visited.

**Case 3:** When an auxiliary node is visited, we mark it as visited and continue to visit its unmarked neighbors.

When restarting the search on a new evidence node, we unmark all targets and auxiliary nodes, because the same targets may occur in different Markov blankets as shown in Figure 2(c). The algorithm stops when all evidence nodes have been visited. The depth first search algorithm automatically partitions the whole evidence set into subsets \( \bigcup_{i} E_{i} \) and finds the Markov blanket \( MB(E_{i}) \) of each subset.
3.4 Merging Markov blankets

When computing the upper bound, we maximize each belief update ratio $r(MB(E_i); e_i)$ according to $MB(E_i)$, independently. Thus the common targets of two different Markov blankets $MB(E_i)$ and $MB(E_j)$ may be set to inconsistent values based on the maximization. Too much inconsistency may result in a loose bound. We can tighten the upper bound by merging Markov blankets that share targets. On the other hand, if the number of targets in an individual Markov blanket is too large, it will make calculating the upper bound inefficient. We propose to merge Markov blankets to reduce the number of duplicates of a target under the constraint that the number of targets in a Markov blanket cannot exceed $K$.

We can use an undirected graph to represent the problem of merging Markov blankets. The nodes in the graph denote Markov blankets. If two Markov blankets share targets, there is an edge between the two corresponding nodes. The weight of the edge is the number of targets shared by the two Markov blankets. This formulation translates the problem of merging Markov blankets into a graph partition problem. More specifically, the merging problem can be addressed by recursively solving the minimum bisection problem (Feige & Krauthgamer, 2002), which partitions the vertices into two equal halves so as to minimize the sum of weights of the edges between the two partitions, on the undirected graph.

Minimum bisection problem is an NP-hard problem, however. We cannot afford to spend too much on computing the upper bound. We therefore use a hierarchical clustering-like greedy algorithm for merging the Markov blankets. We first merge any two Markov blankets if one of them covers the other. Then for all the remaining Markov blankets, we each time merge two Markov blankets that share the most number of targets as long as the number of targets in the resulting Markov blanket does not exceed $K$. The algorithm finishes when no Markov blankets can be merged.

4. Improving the Markov Blanket Bound

The above Markov blanket upper bound has two potential difficulties in scaling to large Bayesian networks with many target variables. First, decomposing Bayesian networks (in subsection 3.3) can lead to large subnetworks with belief ratio tables that are too large to build. Second, the upper bound can be loose.
4.1 Pruned-targets and enclosed-targets

We define two new types of targets before discussing how to tighten the Markov blanket upper bound. One is pruned-target set $P$ of a search state, which satisfies $M = U \cup V \cup P$ and is pruned from the subtree of the state, as illustrated in Figures 1 and 3. For the state $\{b\}$ in Figure 1, $\{A\}$ is the pruned-target set.

Another type of targets is enclosed-target set $H$ of a Markov blanket, which is defined as follows.

**Definition 3.** In MRE inference, enclosed-targets $H$ of a Markov blanket are the targets which are conditionally independent from the nodes outside the Markov blanket given the remaining nodes in the Markov blanket.

In other words, enclosed-targets are the targets which are blocked (d-separated) by the rest of the Markov blanket. From Definition 3, we can see that each enclosed-target only occurs in one Markov blanket. As an example in Figure 2(c), $I$ and $F$ are enclosed-targets in $MB(E_2) = \{A, F, I\}$.

4.2 Splitting large Markov blankets

The decomposition method in subsection 3.3 can lead to large Markov Blankets which prevent the application of BFBnB to large scale MRE problems. To address this problem, we notice that in MRE problems, each subnetwork contains three types of nodes, i.e., targets $MB(E_i)$, evidence $E_i$, and auxiliary nodes $A_i$, where $MB(E_i)$ marks the boundary of subnetworks, and $E_i$ and $A_i$ are conditionally independent from other nodes given $MB(E_i)$. Our idea here is to split large Markov blankets by converting auxiliary nodes into pseudo-targets which only act as targets in compiling Markov blankets and calculating belief ratio tables. These pseudo-targets add additional d-separation to the Bayesian network, thus split the large Markov blankets into smaller ones. Theorems 3 and 4 guarantee that after this operation, we can still find an upper bound on GBF.

**Theorem 4.** Let $MB(E_i)$ be the Markov blanket of the $i^{th}$ subset of evidence $E_i$, and $A_i$ be the set of auxiliary nodes in the $i^{th}$ subnetwork. Assuming that after converting any subset $\emptyset \subset Y \subseteq A_i$ into pseudo-targets, the $i^{th}$ subnetwork is decomposed into a set of Markov blankets $MB(E_{ij})$, where $\cup_j MB(E_{ij}) = MB(E_i) \cup Y$ and $\cup_j E_{ij} = E_i$. Then, the belief update ratio $r(x; e_i)$ is upper bounded as follows.

$$\max_{x, X = MB(E_i)} r(x; e_i) \leq \prod_j \max_{z, Z = MB(E_{ij})} r(z; e_{ij}) \cdot C,$$

where $C = \prod_j p(e_{ij})/p(e_i)$.

**Proof.** Let $X = MB(E_i)$, for any $\emptyset \subset Y \subseteq A_i$ we have,

$$p(e_i | X) = \sum_Y p(e_i | X \cup Y)p(Y | X)$$

$$= \sum_Y \prod_j p(e_{ij} | MB(E_{ij}))p(Y | X)$$

$$\leq \prod_j \max_{z, Z = MB(E_{ij})} p(e_{ij} | MB(E_{ij}) \setminus Z, z).$$

10
The second equality is based on the property of Markov blankets. Since \( p(e_i | X) = r(X; e_i) p(e_i) \), we have the following upper bound on \( r(x; e_i) \),

\[
\max_{x, X = MB(E_i)} r(x; e_i) \leq \prod_{j} \max_{z, Z = MB(E_{ij})} r(z; e_{ij}) \cdot C,
\]

where \( C = \prod_{i} p(e_{ij}) / p(e_i) \).

We introduce a greedy algorithm to convert auxiliary nodes into pseudo-targets incrementally until the size of each resulting Markov blanket does not exceed \( K \). In the algorithm, we split the Markov blankets whose sizes exceed \( K \) using the following steps.

**Step 1:** Calculate the degree (i.e., the number of neighboring nodes) of each auxiliary node \( A_{ij} \) in the moral graph, and sort them according to their degrees in descending order.

**Step 2:** Convert one auxiliary node into pseudo-target according to its order, and perform depth-first search in subsection 3.3 to compile Markov blankets.

**Step 3:** Stop if the size of each resulting Markov blanket does not exceed \( K \), or repeat Step 2 until all the auxiliary nodes have been converted.

An example of splitting Markov blanket is illustrated in Figure 2. The original Markov blanket \( MB(\{E_1, E_2\}) = \{I, F, B, D\} \) is split into \( MB(E_1) = \{A, B, D\} \) and \( MB(E_2) = \{A, F, I\} \) by converting auxiliary node \( A \) into pseudo-target.

### 4.3 Tightening the upper bound

By combining Theorems 3 and 4, we can see that the splitting process makes the upper bound loose because of the maximizations on pseudo-targets. In order to maintain or even improve the search efficiency, we need to re-tighten the bound. The key idea comes from Theorem 3, in which \( \bar{T}_i \) contains two types of variables, i.e., pruned-targets and unexpanded targets. In Equation 13, both pruned-targets and unexpanded targets of state \( S \) are maximized to derive the upper bound. Since the pruned-targets do not occur in the subsequently generated states (solutions), maximizing pruned-targets makes the upper bound loose.

However, directly **summing** the pruned-targets in the subnetworks will change the structure of Markov blanket decomposition, thus makes the upper bound intractable. In this work, we found that summations can be performed instead of maximizations on the enclosed-targets in individual subnetworks \( MB(E_i) \), which also preserves the structure of Markov blanket decomposition. Based on this observation, we have Theorem 5 to tighten the existing upper bound.

**Theorem 5.** Let \( M = \{X_1, X_2, \ldots, X_n\} \) be a set of targets, \( e \) be the evidence, \( MB(E_i) \) be the Markov blanket of the \( i^{th} \) subset of evidence \( E_i \). Let \( U, V, \) and \( P \) be the unexpanded, expanded, and pruned target sets of state \( S \). Let \( T_i = MB(E_i) \cap V \), and \( \bar{T}_i = MB(E_i) \setminus T_i \). Let \( H_i \) be the enclosed-target set in \( MB(E_i) \) and \( H_{is} = H_i \cap P \). Then, for any subset \( \emptyset \subset X \subseteq U \), the belief update ratio \( r(x \cup v; e) \) is upper bounded as follows.

\[
\max_{x, \emptyset \subset X \subseteq U} r(x \cup v; e) \leq \prod_{i} \max_{z, Z = T_i \setminus H_{is}} r(z \cup t_i; e_i) \cdot C, \tag{17}
\]

where \( C = \prod_{i} p(e_i) / p(e) \).
Proof. For any \( \emptyset \subset X \subseteq U \), let \( W = X \cup V \) and \( \bar{W} = M \setminus W \). Let \( S_i = MB(E_i) \cap \bar{W} \), \( I = \{ i : S_i \neq \emptyset \} \), and \( J = \{ i : S_i = \emptyset \} \), we have,

\[
p(e|W) = \sum_{W} p(e|M)p(\bar{W}|W)
\]

\[
= \sum_{W} \left( \prod_{i \in I} p(e_i|MB(E_i)) \right) p(\bar{W}|W) \cdot \prod_{i \in J} p(e_i|MB(E_i)).
\]

Equation 18 is based on the property of Markov blankets.

Assuming \( \bar{X} = U \setminus X \), we have \( \bar{W} = P \cup \bar{X} \) and \( H_{is} \subseteq S_i \subseteq \bar{W} \). Since \( H_{is} \) only occurs in \( MB(E_i) \), we perform summation on non-empty \( H_{is} \) in Equation 18 as follows.

\[
\sum_{H_{is}} p(e_i|MB(E_i))p(H_{is}|M \setminus H_{is})
\]

\[
= \sum_{H_{is}} \frac{p(e_i,H_{is}|MB(E_i))p(H_{is}|MB(E_i))}{p(H_{is}|MB(E_i))} \cdot p(H_{is}|M \setminus H_{is})
\]

\[
= \sum_{H_{is}} p(e_i,H_{is}|MB(E_i)) \cdot p(H_{is}|M \setminus H_{is})
\]

\[
= p(e_i|MB(E_i)) \cdot p(H_{is}|M \setminus H_{is}),
\]

where \( p(H_{is}|MB(E_i)) = p(H_{is}|M \setminus H_{is}) \), since \( H_{is} \) is conditionally independent with other targets given \( MB(E_i) \).

Based on Equation 19, we obtain an upper bound by summing out \( H_{is} \) from Equation 18 and replacing the summation on \( \cup_{i}(S_i \setminus H_{is}) \) with maximization.

\[
p(e|W) \leq \prod_{i \in I} \max_{z,z = S_i \setminus H_{is}} p(e_i|MB(E_i) \setminus S_i,z) \cdot \prod_{i \in J} p(e_i|MB(E_i)).
\]

Since \( p(e|W) = r(W;e)p(e) \), we have the following upper bound on \( r(x \cup v;e) \) by re-organizing the partition of \( MB(E_i) \).

\[
\max_{x,\emptyset \subseteq X \subseteq U} r(x \cup v;e) \leq \prod_{i \in I} \max_{z,z = T_i \setminus H_{is}} r(z \cup t_i;e_i) \cdot \prod_{i \in J} \max_{z,z = T_i} r(z \cup t_i;e_i) \cdot C,
\]

where \( C = \prod_i p(e_i)/p(e) \).

Thus, for any \( \emptyset \subset X \subseteq U \), we obtain the tighter upper bound on \( r(x \cup v;e) \).

\[
\max_{x,\emptyset \subseteq X \subseteq U} r(x \cup v;e) \leq \prod_{i \in I} \max_{z,z = T_i \setminus H_{is}} r(z \cup t_i;e_i) \cdot C.
\]

By substituting Equation 17 into 14, we have the tightened upper bound on GBF in Theorem 6.

Theorem 6. Let \( M = \{ X_1, X_2, \ldots, X_n \} \) be a set of targets, \( e \) be the evidence, \( MB(E_i) \) be the Markov blanket of the \( i^{th} \) subset of evidence \( E_i \). Let \( U, V, \) and \( P \) be the unexpanded, expanded, and pruned target sets of state \( S \). Let \( T_i = MB(E_i) \cap V \), and \( \bar{T}_i = MB(E_i) \setminus T_i \). Let \( H_i \) be
Figure 4: Subproblem graph of compiling belief ratio tables for the incremental algorithm.

The enclosed-target set in $MB(E_i)$ and $H_{i|s} = H_i \cap P$. Then, for any subset $\emptyset \subset X \subseteq U$, the generalized Bayesian factor score $GBF(x \cup v; e)$ is upper bounded as follows.

$$\max_{x, \emptyset \subset X \subseteq U} GBF(x \cup v; e) \leq 1 + \frac{\prod_{z, \max_{T_i \setminus H_{i|s}}} r(z \cup t_i; e_i) \cdot C - 1}{1 - p(v|e)},$$

where $C = \prod_i p(e_i)/p(e)$.

The main difference between Theorems 3 and 6 is that Theorem 3 maximizes $H_{i|s}$ in the upper bound while Theorem 6 sums out $H_{i|s}$ which results in a tighter upper bound.

### 4.4 Compiling belief ratio tables

The belief ratio tables, which contain the belief update ratios of all the configurations of a series of target sets generated based on $MB(E_i)$, are used to calculate the upper bounds during MRE inference. To compile belief ratio tables, we find the set of enclosed-targets $H_i$ by converting each target in $MB(E_i)$ into an auxiliary node individually. If this does not add new targets or evidence nodes into the original subnetwork, we add the target into $H_i$. All the subsets of $H_i$ are used to build $2|H_i|$ belief ratio tables of $MB(E_i)$, where $|H_i|$ denotes the size of $H_i$. Let $H_{ij}$ be a subset of $H_i$, then the target set of the belief ratio table is $MB(E_i) \setminus H_{ij}$. Since the number of belief ratio tables increases exponentially according to $|H_i|$, we limit $|H_i|$ to be no larger than $N$ for both time and space reasons. Enclosed-targets contribute differently to tighten the upper bound, thus we select the top $N$ enclosed-targets from $H_i$ by sorting the enclosed-targets $H_{ij}$ according to their belief update ratios $\max_{h_{ij}} r(e_i|h_{ij})$ in descending order.

To compile belief ratio tables, we need to calculate the belief update ratios of all the configurations of target set $MB(E_i) \setminus H_{ij}$. A straightforward method is to compile each of the $2|H_i|$ belief ratio tables independently. But this is slow because of redundant computation. In this work, we propose an incremental algorithm which compiles the belief ratio tables gradually from the smallest target set $H_i = MB(E_i) \setminus H_i$. In Figure 4, assuming there are four enclosed-targets, i.e., $H_i = \{A, B, C, D\}$, organized in alphabetical order, and each node in the subproblem graph represents a target. We first compile the belief ratio table of $\tilde{H}_i$, then traverse the subproblem graph in breadth-first order. At each state (node), we compile the belief ratio table by adding the target.
into the complied belief ratio table of its parent state. After traversing the subproblem graph, the algorithm compiles all of the belief ratio tables.

In the proposed algorithm, we calculate a hash key $HK_i$ of each Markov blanket based on $H_i$. The belief ratio tables of each Markov blanket are stored in a hash table and the hash keys are calculated based on the corresponding subset of enclosed-targets $H_{ij}$.

5. Breadth-First Branch-and-Bound Algorithms Based on Markov Blanket Upper Bounds

In MRE inference, all of the search nodes are potential solutions except the root node. We can choose from a variety of search methods to explore the search tree, e.g., depth-first search, best-first search, and breadth-first search. We chose breadth-first search in MRE inference for two reasons. First, it is convenient to identify the convergence of MRE by monitoring the number of states in the search queue. Usually, the size of the search queue first increases and then decreases in the BFBnB algorithm. Second, since MRE prunes away independent and less relevant targets, usually the number of targets in the optimal solution is not large. Thus breadth-first search may reach the optimal solutions faster than other search strategies. The breadth-first search costs more memory to store the unexpanded states in a layer, however.

By using the derived upper bounds, we propose BFBnB algorithms which contain two parts, i.e., preprocessing and search. The preprocessing part includes compiling Markov blankets, merging Markov blankets sharing targets, splitting large Markov blankets, and generating the belief ratio tables of each $MB(E_i)$. A suite of upper bounds can be derived by including part of the above preprocessing modules. Based on our empirical studies, the Markov blanket upper bounds in Theorems 3 and Theorems 6 show excellent performance.

The search part of BFBnB explores the search tree layer by layer while keeping track of the highest-scoring state, and prunes state $S$ if the upper bound is less than the current best GBF. The use of different upper bounds proposed in this work leads to two versions of the BFBnB algorithms, MavBnd and TitBnd. MavBnd is based on the upper bound in Theorem 3, which maximizes on both the unexpanded targets and pruned targets. Thus, in MavBnd, each Markov blanket $MB(E_i)$ has one belief ratio table which is derived by calculate the belief update ratios of all configurations of $MB(E_i)$. For each state $S$, we search for a configuration of each Markov blanket that is consistent with the expanded targets $t_j$ and has the highest belief update ratio. We then calculate the upper bound of $S$ using Theorem 3.

TitBnd is based on the upper bound in Theorem 6, which integrates part of the pruned-targets based on the structure of Bayesian networks. Thus, in TitBnd, each Markov blanket $MB(E_i)$ has a set of belief ratio tables each of which is derived by calculate the belief update ratios of all configurations of $MB(E_i) \setminus H_{ij}$. For each state $S$, we calculate a hash key $SK$ based on the pruned-targets and use $SK$ to $HK_i$ to index the belief ratio table of each Markov blanket. Then, we search for a configuration of each selected belief ratio table that is consistent with the expanded targets $t_j$ and has the highest belief update ratio. Finally, we calculate the upper bound of $S$ using Theorem 6.

To speed up the search process, we sort each belief ratio table of a Markov blanket in descending order, i.e., higher belief update ratios are closer to the front of the table. Furthermore, in the search tree of MRE, the expanded targets of a state are guaranteed to be included in its descendant states. Thus in the proposed method, we record the indices of the best belief update ratios, one for each
belief ratio table, for each expanded state. To calculate the upper bound of a current state, we only need to search the best belief update ratio from the recorded indices of its parent.

6. Experiments

The proposed algorithms are evaluated on six benchmark diagnostic Bayesian networks listed in Table 1, i.e., Alarm (Ala), Carpo (Car), Hepar (Hep), Insurance (Ins), Emdec6h (Emd), and CPCS179 (Cpc) (Beinlich, Suermondt, Chavez, & Cooper, 1989; Binder, Koller, Russell, & Kanazawa, 1997; Onisko, 2003; Pradhan, Provan, Middleton, & Henrion, 1994). Among them, Alarm, Carpo, Hepar, and Insurance are networks with fewer than 100 nodes. Emdec6h and CPCS179 are larger networks with more than 100 nodes. We listed the number of nodes (Nodes), the number of leaf nodes (Leaves), the average number of node states (States), and the number of arcs (Arcs) of the Bayesian networks in Table 1. The experiments were performed on a 2.67GHz Intel Xeon CPU E7 with 512G RAM running a 3.7.10 Linux kernel.

6.1 Experimental design

Since the proposed methods, i.e., MavBnd and TitBnd, are the first nontrivial exact MRE algorithms, we had to use a naive Breadth-First Brute-Force search algorithm (BFBF) as the baseline; basically BFBF is BFBnB with the bound set to be infinity. We also included the results of tabu search to indicate the difficulty of MRE problems. Both TitBnd and MavBnd are exact algorithms for solving MRE inference. In TitBnd and MavBnd, we set the maximum number of targets in a Markov blanket $K$ to be 18. In TitBnd, we set the maximum number of enclosed-targets in a Markov blanket $N$ to be 7. In tabu search, we set the number of search steps since the last improvement $L$ and the number of search steps $M$ according to different network settings. For the networks with 12 targets, we set $L$ to be 20 and $M$ to be $\{400, 800, 1600, 3200, 6400\}$. For the networks with about 20 targets, we set $L$ to be 80 and $M$ to be $\{12800, 25600, 51200\}$. To evaluate search performance, we compared the solutions of tabu search and TitBnd, and counted the number of test cases on which tabu search achieved optimal solutions.

BFBF search can only solve test cases with fewer than 15 targets. To compare to BFBF, we randomly generated five test settings of each network, each setting consisting of all leaf nodes as evidence, 12 of the remaining nodes as targets, and others as auxiliary nodes. Then for each setting, we randomly generated 20 configurations of evidence (test cases) by sampling from the prior distributions of the networks.

<table>
<thead>
<tr>
<th>Networks</th>
<th>Nodes</th>
<th>Leaves</th>
<th>States</th>
<th>Arcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>37</td>
<td>11</td>
<td>2.84</td>
<td>46</td>
</tr>
<tr>
<td>Carpo</td>
<td>61</td>
<td>43</td>
<td>2.23</td>
<td>74</td>
</tr>
<tr>
<td>Hepar</td>
<td>70</td>
<td>41</td>
<td>2.31</td>
<td>123</td>
</tr>
<tr>
<td>Insurance</td>
<td>27</td>
<td>6</td>
<td>3.30</td>
<td>52</td>
</tr>
<tr>
<td>Emdec6h</td>
<td>168</td>
<td>117</td>
<td>2.00</td>
<td>261</td>
</tr>
<tr>
<td>CPCS179</td>
<td>179</td>
<td>151</td>
<td>2.29</td>
<td>239</td>
</tr>
</tbody>
</table>

Table 1: Benchmark diagnostic Bayesian networks used to evaluate the proposed algorithm.
To compare MavBnd and TitBnd, we randomly generated five test settings of each network, each setting consisting of all leaf nodes as evidence, about 20 of the remaining nodes as targets, and others as auxiliary nodes. Then for each setting, we randomly generated 20 configurations of evidence (test cases) by sampling from the prior distributions of the networks.

We also compare TitBnd and MavBnd on three Bayesian networks, i.e., Hepar, Emdec6h, and CPCS179, with gradually increasing number of targets from 17 to 25 with an increment of 2. For each target number \( i \), we randomly generated four test settings and 5 test cases of each setting. In the Bayesian networks, we set all leaf nodes as evidence, \( i \) of the remaining nodes as targets, and others as auxiliary nodes.

### 6.2 Importance of splitting large Markov blankets

To show the importance of splitting large Markov blankets, we calculated the percentage of test cases that need splitting operation at various target settings on three Bayesian networks, i.e., Hepar, Emdec6h, and CPCS179. All the Markov blankets in Figure 5 can be split into smaller ones with size less than \( K=18 \) by using the proposed splitting algorithm. The results in Figure 5 show that the ratio increases, and then decreases with the increasing numbers of targets. The reason is that initially the increase in the number of targets leads to large Markov blankets. But as the number of targets keeps increasing, densely distributed targets tend to introduce more d-separation to the Bayesian network and result in smaller Markov blankets. In Figure 5, the peaks of curves located at 22 approximately. On the networks with a large number of non-leaf nodes, e.g., Emdec6h, the ratio is higher than 0.5 on a significant number of target settings. This means that without splitting large Markov blankets most of MRE problems cannot be solved on the networks with these target settings.

### 6.3 Evaluation of MavBnd and TitBnd on networks with 12 targets

In Table 2, we compared the proposed MavBnd and TitBnd algorithms with BFBF and tabu search on the test cases with 12 targets. For tabu search, we listed the accuracy (top) and running time (bottom) at different numbers of search steps. MavBnd, TitBnd and BFBF were able to solve the
Table 2: Comparison of TitBnd, MarBnd, BFBF, and tabu on running time and accuracy on Bayesian networks with 12 targets.

<table>
<thead>
<tr>
<th></th>
<th>Ala</th>
<th>Car</th>
<th>Hep</th>
<th>Ins</th>
<th>Emd</th>
<th>Cpc</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFBF</td>
<td>1.7e3</td>
<td>66.1</td>
<td>270</td>
<td>1.6e5</td>
<td>212</td>
<td>1.3e4</td>
</tr>
<tr>
<td>MavBnd</td>
<td>17.0</td>
<td>3.5</td>
<td>14.6</td>
<td>1.6e3</td>
<td>15.1</td>
<td>815</td>
</tr>
<tr>
<td>TitBnd</td>
<td>2.6</td>
<td>1.5</td>
<td>9.0</td>
<td>72.7</td>
<td>13.1</td>
<td>377</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>T6400</th>
<th>T3200</th>
<th>T1600</th>
<th>T800</th>
<th>T400</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>85</td>
<td>79</td>
<td>76</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>17.3</td>
<td>9.1</td>
<td>4.7</td>
<td>2.4</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>25.4</td>
<td>13.2</td>
<td>6.9</td>
<td>3.6</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>91</td>
<td>86</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>44.8</td>
<td>22.6</td>
<td>11.5</td>
<td>5.9</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>77</td>
<td>74</td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>26.3</td>
<td>13.7</td>
<td>7.0</td>
<td>3.6</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>89.3</td>
<td>45.0</td>
<td>23.2</td>
<td>11.8</td>
<td>6.1</td>
</tr>
<tr>
<td></td>
<td>108</td>
<td>55.1</td>
<td>28.0</td>
<td>14.1</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Test cases exactly. MavBnd is shown to be significantly faster than BFBF because of the pruning by the upper bound. The running time of TitBnd is further reduced by using the tightened upper bound. T400 is the fastest algorithm with the worst accuracy. With the increase of $M$, the running time of tabu search increased significantly. However, in most of the networks, tabu search could not solve all the test cases optimally, even using more running time than MavBnd and TitBnd.

To compare MavBnd, TitBnd and BFBF in detail, we computed the averaged running time of individual test settings for MavBnd, TitBnd and BFBF. Then we illustrated the running time pair, i.e., (BFBF, MavBnd) and (MavBnd, TitBnd), in logarithm as a point in Figure 6 and Figure 7, respectively. We also drew the contour lines to mark the difference between the logarithm running time in Figure 6 and Figure 7. For example, the contour line marked by -3 in Figure 6 contains the points on which MavBnd is 1000 times faster than BFBF. The results showed that although the averaged running time may change significantly, the ratios of running times between BFBF and MavBnd, and between MavBnd and TitBnd are relatively stable. MavBnd is roughly 10 to 100 times faster than BFBF. TitBnd is roughly 3 to 4 times faster than MavBnd.

6.4 Evaluation of MarBnd and TitBnd on networks with about 20 targets

In Table 3, we compared the proposed TitBnd algorithm with MavBnd and tabu search on the test cases with about 20 targets. We listed the number of targets of each network in the first row of Table 3. For tabu search, we listed both accuracy (top) and running time in minutes (bottom) on each network. Increasing $M$ from 12800 to 51200 is not helpful in preventing tabu search from getting stuck in the local optima. Moreover, the performance of tabu search varies greatly on different test networks. TitBnd is shown to be significantly faster than MavBnd on most of the networks due to the tightened upper bound. In TitBnd, we need to compile a series of belief ratio tables for each Markov blanket, which may consume significant amount of running time. For example, in CPCS179 although the running time is 419 minutes, the search time is only 182.4 minutes on average. Also
from the results, we can see that the running time of MavBnd and TitBnd depends on not only the number of targets, but also the tightness of upper bound and the Bayesian network structures, which control the number of pruned states and the size of belief table in each Markov blanket $MB(E_i)$, respectively.

To compare TitBnd and MavBnd in more detail, we computed the averaged running time of individual test settings for both of them, and illustrated each running time pair in logarithm as a point in Figure 8. We also drew the contour lines to mark the difference between TitBnd and
Table 3: Comparison of TitBnd, MavBnd, and tabu on running time and accuracy on Bayesian networks with about 20 targets.

MavBnd. The results showed that the data points of each network form a cluster. TitBnd is roughly 3 to 4 times faster than MavBnd.

The results in both Figure 7 and Figure 8 showed that for some test cases the running times of MavBnd and TitBnd are close. There are two reasons behind this. First, for some network structures, the enclosed-target sets may not exist in the individual Markov blankets. Thus in these cases TitBnd will degenerate to MavBnd. Second, the running time of each test case consists two parts, compiling belief ratio tables and performing search. For some test cases, compiling belief ratio tables may take significant among of time, which may make TitBnd slow. For the second reason, we can make the tradeoff between compiling time and search time by adjusting the maximum number of enclosed-targets $N$ in a Markov blanket.

6.5 Scalability of TitBnd

We also evaluated the scalability of TitBnd on the test cases with gradually increasing number of targets. In the experiments, we set the time limit of running to be 800 minutes and use OT to indicate the test cases running out of time limit in Table 4. The results show that the pruning of upper bound slowed down the exponential growth of running time significantly. TitBnd can handle more complex MRE problems which are out of the reach of MavBnd. From the results, we can also see that the running time of TitBnd is affected by the tightness of upper bound and the structures of Bayesian networks, which no longer heavily depends on the number of targets.

6.6 Effect of summing out enclosed-targets

In this section, we take Hepar, Emdec6h, and CPCS179 as examples to illustrate how summing out enclosed-targets tightens the Markov blanket upper bound. We calculated the difference between the upper bound $UpBnd$ and the maximum GBF $MaxGBF$ in each subtree rooted at a search state for the test cases with 12 targets. Normalized histograms of the differences for both using and not using the tightening are plotted in Figure 9. It is clear from the graph that summing out the enclosed-targets makes the bound much tighter.
Figure 8: Distributions of logarithm running time pairs of TitBnd and MavBnd on Bayesian networks with about 20 targets.

Table 4: Comparison of TitBnd and MavBnd on running time on networks with gradually increasing number of targets.

<table>
<thead>
<tr>
<th>Min</th>
<th>Targets</th>
<th>17</th>
<th>19</th>
<th>21</th>
<th>23</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>MavBnd</td>
<td>Hep</td>
<td>64.7</td>
<td>101.0</td>
<td>411.3</td>
<td>OT</td>
<td>OT</td>
</tr>
<tr>
<td></td>
<td>Emd</td>
<td>63.6</td>
<td>182.4</td>
<td>643.4</td>
<td>OT</td>
<td>OT</td>
</tr>
<tr>
<td></td>
<td>Cpc</td>
<td>483.7</td>
<td>OT</td>
<td>OT</td>
<td>OT</td>
<td>OT</td>
</tr>
<tr>
<td>TitBnd</td>
<td>Hep</td>
<td>24.0</td>
<td>15.0</td>
<td>66.9</td>
<td>243.1</td>
<td>711.9</td>
</tr>
<tr>
<td></td>
<td>Emd</td>
<td>36.3</td>
<td>100.8</td>
<td>444.2</td>
<td>OT</td>
<td>OT</td>
</tr>
<tr>
<td></td>
<td>Cpc</td>
<td>130.7</td>
<td>94.1</td>
<td>499.6</td>
<td>OT</td>
<td>OT</td>
</tr>
</tbody>
</table>

6.7 Convergence of upper bound

To gain a better perspective on how the tightened upper bound in Theorem 6 improves over time, we calculated both the maximum upper bound \( \text{MaxBound} \) in each layer of the search tree and the current maximum GBF \( \text{LocMax} \). We also recorded the running time when we finished each search layer. Figure 10 shows the convergence curves of \( \text{MaxBound} \) (dot line) and \( \text{LocMax} \) (circle line) against the search time for one typical test case of each of the three networks, i.e., Hepar, Emdec6h, and CPC179, under the 17-target setting in Table 4. Figure 10(A) shows that the upper bound dropped sharply at the end of the search and became tight quickly. Figure 10(B) shows that the upper bound dropped gradually during the search. Figure 10(C) shows that the upper bound dropped quickly at both the beginning and the end of the search. These results demonstrate different behaviors how the bound is tightening during the search.
EXACT ALGORITHMS FOR MRE INFERENCE

Figure 9: Tightness of upper bounds. We evaluate the effect of summing out enclosed-targets using normalized histogram on different Bayesian networks, i.e., Hepar (A), Emdec6h (B), and CPCS179 (C).

Figure 10: Convergence of upper bound and local maximum on different Bayesian networks, i.e., Hepar (A), Emdec6h (B), and CPCS179 (C).

7. Discussions and Conclusions

The main contributions of this paper are two BFBnB algorithms, i.e., MavBnd and TitBnd, for solving MRE exactly using upper bounds based on Markov blanket decomposition. These are the first exact algorithms for solving MRE in Bayesian networks. The key idea of the proposed method is to decompose the conditional joint probability $p(E|M)$ into a set of marginal probabilities $p(E_i|MB(E_i))$. Each marginal probability is related to a subnetwork characterized by a Markov blanket of a subset of evidence nodes. An upper bound on GFB is derived by maximizing belief update ratio on the fixed target set $MB(E_i)$ of each subnetwork separately. The upper bound can be tightened by merging the Markov blankets which sharing the same set of targets.

Furthermore, another improved upper bound is proposed based on two ideas. First, we proposed to split large Markov blankets by converting auxiliary nodes into pseudo-targets, which enables solving MRE problems on complex Bayesian networks with many targets. Second, we tightened the upper bound on GFB by identifying and summing out enclosed-targets from each Markov blan-
ket $MB(E_i)$, which does not change the structure of Markov blanket decomposition of the whole network. The new upper bound reduces the search space dramatically. The experimental results show that TitBnd is significantly faster than MavBnd and BFBF algorithms. The proposed TitBnd and MavBnd can solve MRE inference exactly in Bayesian networks which could not be solved previously.

The proposed upper bounds can be calculated efficiently for two reasons. First, each Markov blanket $MB(E_i)$ is usually much smaller than the whole target set $M$. Second, in the original MRE problem, we need to search all the subsets of target set $M$ to find the best solution. However, to calculate upper bound, we only need to search on a fixed target set $MB(E_i)$ of each subnetwork.

In MRE inference, the proposed upper bound of a search state consists of four parts which come from different sources of relaxations, i.e., (1) the relaxation from $p(x, v|e)$ to $p(v|e)$, (2) the maximal GBF of all the descendant states, (3) the inconsistent values of the sharing nodes of different Markov blankets, and (4) the maximization on the pruned-targets of individual search states. The optimal upper bound can be achieved should be equal to (2). When the size of expanded target set increase, both $p(x, v|e)$ and $p(v|e)$ become much smaller than 1, and the relaxation of (1) becomes very tight. Thus, to achieve the optimal upper bound we only need to minimize the impact of (3) and (4). In this work, we minimized the effect of (3) by merging the Markov blankets sharing the same set of targets, and minimized the effect of (4) by identifying and summing out the enclosed-target sets from each Markov blanket.

Different from the brute-force algorithm, the search time of MavBnd and TitBnd is no longer mainly dependent on the size of search space (i.e., the number of targets and the number of states of each target), but also on the tightness of the upper bound and the structure of Bayesian networks. For Bayesian networks with a large number of targets, an upper bound can be efficiently generated as long as the the number of targets in each Markov blanket is small. In TitBnd, the tightness of upper bound depends on the number of enclosed-targets and the quality of each enclosed-target $H_{ij}$ measured by its belief update ratio $\max_{h_{ij}} r(e_i|h_{ij})$. If there are no enclosed-targets, the proposed method TitBnd degenerates to MavBnd.

In this work, the splitting algorithm is designed to split Markov blankets, whose sizes are larger than $K$, based on the separation property of undirected graphs by converting auxiliary nodes into pseudo-targets. If an evidence node is connected directly with many targets, we may decompose it using the methods such as node splitting (Choi et al., 2007). This is also one of our future research directions.

Acknowledgements This work was supported by NSF grants IIS-0953723, IIS-1219114, and a PSC-CUNY enhancement award. Part of this research has previously been presented in AAAI-15 (Zhu & Yuan, 2015).

References


