Computational Complexity and Approximation Methods of Most Relevant Explanation

Heejin Lim & Changhe Yuan
Department of Computer Science and Engineering
Mississippi State University
Mississippi State, MS 39762
hl204@msstate.edu & cyuan@cse.msstate.edu

Abstract

Most Relevant Explanation (MRE) is a new approach to generating explanations for given evidence in Bayesian networks. MRE has a solution space containing all the partial instantiations of target variables and is extremely hard to solve. We show in this paper that the decision problem of MRE is NP^{PP}-complete. For large Bayesian networks, approximate methods may be the only feasible solutions. We observe that the solution space of MRE has a special lattice structure that connects all the potential solutions together. The connectivity motivates us to develop several efficient local search methods for solving MRE. Empirical results show that these methods can efficiently find the optimal MRE solutions for majority of the test cases in our experiments.

1 Introduction

Most Relevant Explanation (MRE) is a new approach to generating explanations for given evidence in Bayesian networks (Yuan & Lu 2008; Yuan *et al.* 2009). Its main idea is to traverse a trans-dimensional space that contains all the partial instantiations of the target variables and find an instantiation that maximizes a relevance measure called *generalized Bayes factor* (Fitelson 2007). The approach was shown theoretically and empirically in (Yuan *et al.* 2009) to be able to find precise and concise explanations. Readers are referred to (Yuan *et al.* 2009) for the theoretical properties and empirical results.

The solution space of MRE is huge for large Bayesian networks. MAP only considers the full instantiations of the target variables and was already shown to be an NP^{PP}-complete problem (Park 2002). MRE has an even larger solution space and is believed to be extremely difficult to solve. We show in this paper that the decision problem of MRE is also NP^{PP}-complete. Therefore approximation methods may be the only feasible solutions for large Bayesian networks. A method based on Reversible Jump MCMC (Green 1995) and simulated annealing was proposed to solve MRE approximately in (Yuan & Lu 2008). Although theoretically this algorithm will converge to an optimal solution given enough time, the amount of time needed is unpredictable. The algorithm can still be inefficient on some networks.

In this paper, we develop several local search methods for solving MRE. Local methods have been shown to be effective in solving MAP problems in Bayesian networks (Park & Darwiche 2001). We further observe that the solution space of MRE has a special lattice structure that connects all the potential solutions, which motivates us to design several local search strategies to find high-quality solutions efficiently. Empirical results show that these local search methods are not only efficient but also able to find the optimal solutions for most of the test cases in our experiments.

The remainder of the paper is structured as follows. Section 2 reviews the formulation of Most Relevant Explanation. Section 3 proves that the decision problem of MRE is NP^{PP}-complete. Section 4 develops several local search methods for solving MRE, including forward search, forward-backward search, and partial exhaustive search. Finally, Section 5 presents the empirical evaluations of these local MRE methods on a set of benchmark Bayesian networks.

2 Most Relevant Explanation

Most Relevant Explanation is a method developed for Bayesian networks to automatically identify the most relevant target variables in explaining given evidence (Yuan & Lu 2008; Yuan *et al.* 2009). First, *explanation* in Bayesian networks is formally defined as follows.

Definition 1. *Given a set of target variables* $\mathbf{X}$ *in a Bayesian network and evidence* $\mathbf{e}$ *on the remaining variables, an* explanation $\mathbf{x_{1:k}}$ *for the evidence is a* k*-variate partial instantiation of the target variables, i.e.,* $\mathbf{X_{1:k}} \subseteq \mathbf{X}$ *and* $\mathbf{X_{1:k}} \neq \emptyset$.

Most Relevant Explanation (MRE) is defined as follows.

Definition 2. *Let* $\mathbf{X}$ *be a set of target variables, and* $\mathbf{e}$ *be the evidence on the remaining variables in a Bayesian network.* Most Relevant Explanation *is the problem of finding an explanation* $\mathbf{x_{1:k}}$ *that has the maximum* Generalized Bayes Factor *score* $GBF(\mathbf{x_{1:k}}; \mathbf{e})$*, i.e.,*

$$MRE(\mathbf{X}, \mathbf{e}) \equiv \arg\max_{\mathbf{x_{1:k}}, \mathbf{X_{1:k}} \subseteq \mathbf{X}, \mathbf{X_{1:k}} \neq \emptyset} GBF(\mathbf{x_{1:k}}; \mathbf{e}) \ , \tag{1}$$

where GBF *is defined as*

$$GBF(\mathbf{x_{1:k}}; \mathbf{e}) \equiv \frac{P(\mathbf{e}|\mathbf{x_{1:k}})}{P(\mathbf{e}|\overline{\mathbf{x_{1:k}}})}. \tag{2}$$

3 Computational Complexity of MRE

This section proves the computational complexity of MRE. We first introduce a new decision problem called EP-MAJSAT (exists partial MAJSAT), which generalizes the NP^{PP}-complete E-MAJSAT problem (exists MAJSAT) (Littman, Goldsmith, & Mundhenk 1998). Recall that E-MAJSAT is defined as follows: given a Boolean formula over n variables $X_1, ..., X_n$, we need to find out whether there is an instantiation of the first k variables $X_1, ..., X_k$ such that the majority of the assignments to $X_{k+1}, ..., X_n$ satisfy the Boolean expression. EP-MAJSAT is defined analogously except that we want to find out whether there is a partial instantiation $y_1, ..., y_i$ with a given cardinality i ($i \leq k$) of $X_1, ..., X_k$ such that the majority of the assignments to the remaining variables $\{X_1, ..., X_n\} \setminus \{Y_1, ..., Y_i\}$ satisfy the Boolean expression.

If $i = k = n$, EP-MAJSAT becomes the classic NP-complete Boolean satisfiability problem, which asks whether there exists an assignment to all the variables that makes the formula true (Cook 1971). If $i = 0$, it becomes the PP-complete MAJSAT problem, which asks whether the majority of the complete assignments make the formula true (Gill 1977). If $i = k$, this is simply the E-MAJSAT problem. Otherwise for the general case when $1 \leq i \leq k$, EP-MAJSAT involves first an NP-type task to pick a partial instantiation of $X_1, ..., X_k$ and then a PP-type task to see whether the majority of assignments to the remaining variables make the Boolean formula true. The following theorem shows that EP-MAJSAT is NP^{PP}-complete.

Theorem 1. *EP-MAJSAT is NP^{PP}-complete.*

Proof. Membership in NP^{PP} is clear. We can first guess a partial instantiation of $X_1, ..., X_k$ using a nondeterministic machine in polynomial time. For each such instantiation, we can query a PP MAJSAT oracle to see if the majority of the assignments to all the remaining variables make the Boolean formula true.

To prove hardness, we reduce E-MAJSAT to EP-MAJSAT. The reduction is rather trivial. For a Boolean formula over n variables $X_1, ..., X_n$, E-MAJSAT over $X_1, ..., X_k$ is satisfied if and only if EP-MAJSAT over the same set of variables (cardinality i is equal to k) is satisfied. □

The solution space of MRE consists of all the partial instantiations of the target variables. Intuitively, MRE has a solution space much larger than MAP. Let there be n target variables, each with d states. The size of the solution space for MAP is d^n. The size of the solution space of MRE is $(d+1)^n - 1$ because

$$\sum_{k=1}^{n} C(n,k) * d^k = (d+1)^n - 1, \tag{3}$$

where $C(,)$ is the combination function. So MRE has a solution space with a size equal to that of a MAP problem in an augmented space. It was proven in (Park 2002; Park & Darwiche 2004) that MAP is NP^{PP}-complete by reducing from E-MAJSAT. We prove in this section that MRE has the same computational complexity as MAP by reducing from EP-MAJSAT. We consider the following decision problem of MRE in proving its computational complexity. The decision problem asks whether there is a partial instantiation $y_1, ..., y_i$ with given cardinality i of variables $X_1, ..., X_n$ such that $GBF(y_{1:i}; e) > (2^i - 1)/(2^{i+1}P(e) - 1)$, where $P(e)$ is the probability of evidence e.

Theorem 2. *MRE is NP^{PP}-complete.*

Proof. Membership in NP^{PP} is clear. Computing the probability of any instantiation of a set of variables given evidence on the other variables is shown to be PP-complete (Littman, Majercik, & Pitassi 2001). So we can first guess a partial instantiation using a nondeterministic machine in polynomial time. We then query a PP-oracle to get the prior and posterior probabilities of the partial instantiation, which can be used to compute its GBF score and check whether the score is greater than the given threshold.

To show hardness, we reduce EP-MAJSAT to MRE. For any Boolean formula over n variables $X_1, ..., X_n$, we create a Bayesian network using a method similar to the one presented in (Park 2002; Park & Darwiche 2004). We create a root node with uniform prior probability in the Bayesian network for each boolean variable X_i. We also create a variable for each boolean operator with parent variables corresponding to the operands and with a CPT encoding the truth table of the operator. By doing so, we obtain a variable in the Bayesian network that corresponds to the truth value of the whole Boolean formula. We denote it as V. We set $V = T$ as the evidence to the Bayesian network. The creation of the Bayesian network can be done in polynomial time.

We claim that a solution $y_1, ..., y_i$ of $MRE(X_{1:n}; e)$ on the Bayesian network has a GBF score greater than $(2^i - 1)/(2^{i+1}P(e) - 1)$ if and only if the the same instantiation makes EP-MAJSAT satisfied in the original problem. $P(e)$ is the probability of $V = T$ and can be obtained from a PP-oracle.

First, let $y_1, ..., y_i$ be the MRE solution on the Bayesian network and $GBF(y_{1:i}; e) > (2^i - 1)/(2^{i+1}P(e) - 1)$. Since for any $y_1, ..., y_i$, its prior probability is $P(y_{1:i}) = 1/2^i$ due to the uniform priors on all the variables $X_{i:n}$, we have

$$\begin{aligned}
GBF(y_{1:i}; e) &> \frac{2^i - 1}{2^{i+1}P(e) - 1} \\
\Leftrightarrow \frac{P(e|y_{1:i})}{P(e|\overline{y_{1:i}})} &> \frac{2^i - 1}{2^{i+1}P(e) - 1} \\
\Leftrightarrow \frac{P(y_{1:i}|e)(1 - P(y_{1:i}))}{P(y_{1:i})(1 - P(y_{1:i}|e))} &> \frac{2^i - 1}{2^{i+1}P(e) - 1} \\
\Leftrightarrow \frac{P(y_{1:i}|e)(1 - P(y_{1:i}))}{P(y_{1:i})(1 - P(y_{1:i}|e))} &> \frac{\frac{1}{2^{i+1}P(e)}(1 - \frac{1}{2^i})}{\frac{1}{2^i}(1 - \frac{1}{2^{i+1}P(e)})} \\
\Leftrightarrow P(y_{1:i}|e) &> \frac{1}{2^{i+1}P(e)} \\
\Leftrightarrow P(y_{1:i}, e) &> \frac{1}{2^{i+1}}.
\end{aligned}$$

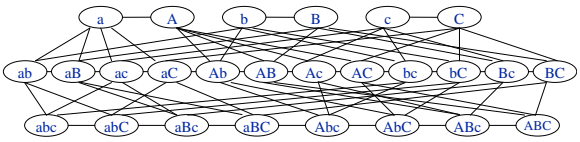

Figure 1: Solution space of Most Relevant Explanation

Then $y_{1:i}$ must make EP-MAJSAT satisfied, because

$$\begin{aligned} P(y_{1:i}, e) &= \sum_{\{X_{1:n}\}\backslash\{Y_{1:i}\}} P(X_{1:n}, e) \\ &= (\#satisfied\ assignments)/2^n \\ &= percentage\ satisfied/2^i. \end{aligned}$$

Second, if there exists a partial instantiation $y_1, ..., y_i$ that makes EP-MAJSAT satisfied, the number of satisfying assignments must be greater than 2^{n-i}. The probability $P(y_{1:i}, e)$ on the Bayesian network must be greater than $1/2^{i+1}$. It follows that $GBF(y_{1:i}; e) > (2^i - 1)/(2^{i+1}P(e) - 1)$. □

4 Local Search Methods for Solving MRE

MRE traverses a solution space that contains all the partial instantiations of a set of target variables to find an optimal solution. An exhaustive search algorithm is impractical for large Bayesian network models with many target variables. More efficient methods, including approximate methods, need to be developed to handle these large domains.

We develop several efficient local search methods for solving MRE. These methods are motivated from the similarity between MRE and feature selection problem (Molina, Belanche, & Nebot 2002) and our observation on the structure of the solution space.

The goal of feature selection is to choose a small subset of features of a dataset to preserve as much predictive power as possible from all the features. Feature selection enables a more compact and robust model to be built, plus it can make the resulting model more intuitive. Since it is often impractical to search all the subsets and find the optimal subset, many feature selection methods are approximation methods, such as the forward and backward search methods (Molina, Belanche, & Nebot 2002). These methods typically start with a candidate subset of features and greedily improves the subset by incorporating the best new feature or deleting the worst existing feature until some stopping criterion is achieved.

In MRE, we need to select not only a subset of variables but also their states to maximize the GBF score. The solution space of MRE is much larger than feature selection. For example, in selecting from a total of n binary variables, the number of possible combinations for feature selection is $\sum_{k=1}^{n} C(n,k)$ or $2^n - 1$. However for MRE, it is $\sum_{k=1}^{n} C(n,k)2^k$ or $3^n - 1$. Therefore MRE has a much larger solution space given the same number of variables than feature selection. The development of local search methods is even more critical for solving MRE problems in large Bayesian networks. In this paper, we adapt local search methods such as forward and backward search to solve MRE.

The solution space of MRE has an interesting structure similar to the graph in Figure 1 for three binary variables. Each node in the graph is a potential solution, and the links show the connectivity between the solutions. Two solutions are connected if they differ only by one variable. Either the state of the variable is different, or one solution has one more variable than the other. The solutions range from singletons to full configurations of the target variables. The forward and backward search methods we are to develop essentially search the graph either top-down or bottom-up.

4.1 Forward Search

In forward search, we first generates one or more initial starting solutions. Then for each starting solution, we greedily improve the solution by either *adding* an additional variable with some state or *changing* the state of one variable in the current solution. Essentially, we are searching the graph in Figure 1 top down. The forward search algorithm is outlined in Algorithm 1.

Algorithm 1 The forward search algorithm

Input: Bayesian network **B** with a set of target variables **X**, and a set of evidence variables **E**
Output: An MRE solution over **X**

```
1: Initialize the starting solution set i with an initialization rule
2: Initialize the current best solution x_best = ∅
3: for each starting solution s in i do
4:    x = s
5:    i = i-{s}
6:    repeat
7:       x_locbest = x
8:       Find the neighboring solution set n of x by either adding one target variable with any state or changing one variable to another state
9:       for each solution x_n in n do
10:         if GBF(x_n) > GBF(x_locbest) then
11:            x_locbest = x_n
12:         end if
13:      end for
14:      if GBF(x_locbest) > GBF(x) then
15:         x = x_locbest
16:      end if
17:   until GBF(x) stops increasing
18:   if GBF(x) > GBF(x_best) then
19:      x_best = x
20:   end if
21: end for
22: return x_best
```

Our forward search algorithm has two rules for initializing the staring solution set. The initialization rules are shown in Figure 2.

```
1: s = ∅
2: i = {s}
3: return i
```

(a) Empty instantiation initialization

```
1: i = ∅
2: for j = 1 to n do
3:     s = {X_j = arg max_{x_j} P(X_j = x_j|E)}
4:     i = i ∪ {s}
5: end for
6: return i
```

(b) Marginal pivot initialization

Figure 2: Two initialization rules of the forward search.

Empty Instantiation Initialization We start the forward search algorithm from an empty instantiation and greedily improve the solution through the search steps.

Marginal Pivot Initialization When using the empty instantiation initialization, we observe that some cases that fall into local maxima are caused by wrong guidance of the target variable added in the very first step by the forward search. Because one misguided greedy step can cause the entire solution to be wrong, we try to avoid this problem by setting each single target variable to its most likely state, called a *pivot*, as a set of starting points of the search algorithm. The pivots can be obtained using any belief updating algorithm, such as the jointree algorithm (Lauritzen & Spiegelhalter 1988). There are n pivoted starting points (n is the number of target variables in the network) in this initialization rule.

4.2 Forward-Backward Search

We also develop several backward search algorithms for MRE. A backward search greedily improve a candidate solution by *changing* the state of one target variable or *deleting* an existing variable from the current solution at each step until no further improvement can be achieved. We search the graph in Figure 1 bottom up in a backward search. The backward search algorithm is given in Algorithm 2.

Starting solutions of the backward search can be full configurations of the target variables generated by combining the marginal pivots, selecting random states, or solving MAP. The first two approaches can generate full instantiations with minimal cost. However, they often generate inconsistent instantiations because they do not consider the interdependence between the target variables. These inconsistent starting points typically lead backward search algorithms to low-quality results. We also tried MAP solution as a starting point of the backward search algorithm because MAP finds the most probable instantiation of target variables given evidence. However, MAP itself has a high computational complexity. It is shown to be a NP^{PP}-complete (Park 2002) problem. Using MAP as an initialization method is not practical.

Instead, we use the results of the forward search to initialize the backward search. Due to the intrinsic pruning capability of MRE (Yuan *et al.* 2009), forward search can already find the global optimal MRE solutions for many cases. For those cases that forward search that only finds sub-optimal solutions, the results are typically not too far off either. We observe that many of them have a couple of more target variables than the optimal solution or the number of target variables is same but some states of the target variables are different. Applying variable deletion and/or state changing is likely to improve these solutions and may even lead to optimal solutions. Therefore, we propose to apply backward search on top of forward search results to further improve the results. The resulting algorithm is a mixed algorithm of the forward search and the backward search, which we call the forward-backward search.

Algorithm 2 The backward search algorithm

Input: Bayesian network **B** with a set of target variables **X**, and a set of evidence variables **E**
Output: An MRE solution over **X**

```
Initialize the starting solution set i with an initialization rule
Initialize the current best solution x_best = ∅
for each starting solution s in i do
    x = s
    i = i-{s}
    repeat
        x_locbest = x
        Find the neighboring solution set n of x by deleting one existing variable or changing one variable to another state
        for each solution x_n in n do
            if GBF(x_n) > GBF(x_locbest) then
                x_locbest = x_n
            end if
        end for
        if GBF(x_locbest) > GBF(x) then
            x = x_locbest
        end if
    until GBF(x) stops increasing
    if GBF(x) > GBF(x_best) then
        x_best = x
    end if
end for
return x_best
```

Because the forward search with two initialization rules generate two starting points of the backward search, the forward-backward search algorithm also has two versions.

4.3 Partial Exhaustive Search

As we mention in the previous subsection, even though the results of forward search are sub-optimal, they are often not off by too much. Many of them have a couple of more target variables than the optimal solution or the number of target variables is same but the states of some target variables are different. Backward search is one way to improve the re-

Network	Targets	Size of solution space	Metrics	F_e	F_{mp}	FB_e	FB_{mp}	EX_e	EX_{mp}
Alarm	12	944,783	*Accuracy*	0.9400	0.9750	0.9500	0.9950	0.9500	0.9950
			Efficiency	0.0001	0.0015	0.0001	0.0015	0.0044	0.0079
Carpo	10	59,048	*Accuracy*	0.9200	1.0000	0.9200	1.0000	0.9200	1.0000
			Efficiency	0.0009	0.0128	0.0010	0.0129	0.0113	0.0361
Hepar	9	34,991	*Accuracy*	0.8250	0.9650	0.8350	0.9650	0.8600	0.9850
			Efficiency	0.0017	0.0192	0.0018	0.0194	0.0183	0.0466
Insurance	5	1,199	*Accuracy*	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Efficiency	0.0325	0.1776	0.0367	0.1827	0.0634	0.2093
Munin	4	599	*Accuracy*	0.8950	0.8800	0.9000	0.9650	0.9150	0.9900
			Efficiency	0.0785	0.2304	0.0935	0.2487	0.3272	0.4992

Table 1: Performance of the forward search algorithm (F), the forward-backward search algorithm (FB), and the partial exhaustive search algorithm (EX) with an empty instantiation initialization (e) and a marginal pivot initialization (mp) on a set of benchmark Bayesian networks. "Targets" shows the number of target variables in the models. "Size of solution space" is the number of candidate solutions to choose from. *Accuracy* is defined as the percentage of cases solved correctly. *Efficiency* is the percentage of solution space searched.

sults but still may result in sub-optimal solutions. Since the forward search typically narrows down the set of candidate target variables significantly, we propose to run the exhaustive search algorithm on the variables identified by forward search, which we call partial exhaustive search. If the results of forward search find all the relevant target variables, the partial exhaustive search algorithm guarantees to find the optimal solutions because it searches all the partial instantiations of a starting solution.

If the results of forward search happen to include a majority or all of the target variables, partial exhaustive search may reduce to the the full exhaustive search algorithm. However, our empirical results show that the number of target variables found is typically small for large Bayesian networks.

5 Empirical Results

In this paper, we propose three local search methods: the forward search algorithm (F), the forward-backward search algorithm (FB), and the partial exhaustive search algorithm (EX). In case of the forward search algorithm, it generates two versions by applying two initialization rules: an empty instantiation initialization (e) and a marginal pivot initialization (mp). The forward-backward search algorithm and the partial exhaustive search algorithm use these two results from the forward search algorithm and also have two possible versions each. Finally, we have six possible algorithms out of the three methods, denoted as F_e, F_{mp}, FB_e, FB_{mp}, EX_e, and EX_{mp}.

5.1 Experimental Design

We tested three local search algorithms on a set of benchmark Bayesian networks, including Alarm, Carpo, Hepar, Insurance, and Munin. We chose these models because we have the diagnostic versions of these networks, whose variables have been annotated into three categories: $target$, $observation$, and $auxiliary$. For generating the test cases, we used the networks as generative models and sampled without replacement from their prior probability distributions. We only kept those test cases with at least one abnormal observation. Munin, Insurance, Hepar, Carpo, and Alarm have 4, 5, 9, 10, and 12 target variables respectively. We collected 200 test cases for each network.

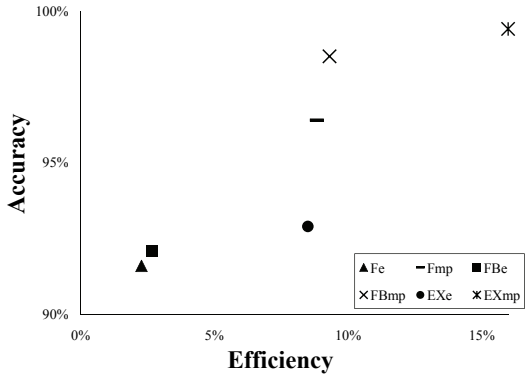

Figure 3: The average *Accuracy* and *Efficiency* of each algorithm in Alarm, Carpo, Hepar, Insurance, and Munin networks.

5.2 Results and Analyses

Our experiments compared results of local MRE methods with the optimal MRE solution obtained by the systematic search. Each algorithm is scored with two metrics, accuracy and efficiency. They are defined as follows:

$$Accuracy = \frac{\#Test\ cases\ solved\ correctly}{\#All\ test\ cases} \quad (4)$$

$$Efficiency = \frac{\#Search\ steps}{Size\ of\ the\ entire\ solution\ space}. \quad (5)$$

The results are shown in Table 1. The average of the results are shown in Figure 3. We did not report running times because they are proportional to the efficiency measure for any given model. We observe that the algorithms provides a wide-range of trade-off between accuracy and efficiency.

- The forward search algorithm finds a solution very quickly, but its accuracy is lower than the other two algorithms.
- The forward-backward search algorithm provides slightly better accuracy and worse efficiency than the forward search algorithm. Because the forward-backward search starts from the result of the forward search, the accuracy result of the forward-backward search is guaranteed to be no worse than forward search.
- Two initialization rules of the forward search algorithm have different performances. The empty-instantiation initialization method is fast but with lower accuracy. In comparison, the marginal pivot initialization method demonstrates higher accuracy but searches larger space than the empty-instantiation initialization method.
- The partial exhaustive search is the most accurate algorithm among three. It is not surprising because it exhaustively searches all the variables found by forward search. It shows pretty high accuracy on all the benchmark models. Because it sacrifices speed for accuracy, however, its efficiency is worse than the other two algorithms. Especially on Munin network, the efficiency of the partial exhaustive search reaches about 50%. For the Munin network, there are only four target variables. The optimal solution often contains a majority of the target variables. In that case, not many target variables are excluded from the forward search either. The partial exhaustive search algorithm often reduces to the systematic search algorithm because the number of target variables given to the partial exhaustive search algorithm is not partial any more. That is why the efficiency on the Munin network is poor. In comparison, when the network has relatively many target variables, say Alarm, the performance of local search algorithms is outstanding.

6 Concluding Remarks

In this paper, we show that the computational complexity of Most Relevant Explanation (MRE) is NP^{PP}-complete, which motivates us to develop efficient approximate methods for solving MRE. According to the similarity between MRE and feature selection, we develop three local search strategies for solving MRE approximately, including the forward search algorithm, the forward-backward search algorithm and the partial exhaustive search algorithm.

Our experimental results show that the proposed local search methods can often solve MRE efficiently and accurately. In case of the network having the relatively large number of target variables, say Alarm, local search methods search only 0.01 - 0.79% of the entire solution space, but they solve 94.00 - 99.50% of test cases optimally.

By introducing three local search strategies with two initialization rules each, totally six possible algorithms, we provide a wide selection of algorithms. The different algorithms provide a trade-off between accuracy and efficiency. If a fast algorithm is needed, the forward search algorithm and the forward-backward search algorithm can be used. If accuracy is more important, the partial exhaustive search algorithm can be used to achieve the highest accuracy.

7 Acknowledgements

This research was supported by the National Science Foundation grants IIS-0842480 and EPS-0903787. All experimental data have been obtained using SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory at University of Pittsburgh and available at `http://genie.sis.pitt.edu`.

References

Cook, S. 1971. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 151158.

Fitelson, B. 2007. Likelihoodism, Bayesianism, and relational confirmation. *Synthese* 156(3).

Gill, J. 1977. Computational complexity of probabilistic turing machines. *SIAM Journal on Computing* 6(4):675–695.

Green, P. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrica* 82:711–732.

Lauritzen, S. L., and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)* 50(2):157–224.

Littman, M.; Goldsmith, J.; and Mundhenk, M. 1998. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research* 9:1–36.

Littman, M. L.; Majercik, S. M.; and Pitassi, T. 2001. Stochastic Boolean satisfiability. *J. Autom. Reasoning* 27(3):251–296.

Molina, L. C.; Belanche, L.; and Nebot, A. 2002. Feature selection algorithms: a survey and experimental evaluation. *Proceedings of 2002 IEEE International Conference on Data Mining* 306–313.

Park, J. D., and Darwiche, A. 2001. Approximating MAP using local search. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI–01)*, 403–410.

Park, J. D., and Darwiche, A. 2004. Complexity results and approximation strategies for MAP explanations. *J. Artif. Intell. Res.(JAIR)* 21:101–133.

Park, J. D. 2002. MAP complexity results and approximation methods. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI–02)*, 388–396.

Yuan, C., and Lu, T.-C. 2008. A general framework for generating multivariate explanations in Bayesian networks. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence (AAAI-08)*.

Yuan, C.; Liu, X.; Lu, T.-C.; and Lim, H. 2009. Most Relevant Explanation: Properties, algorithms, and evaluations. In *Proceedings of 25th Conference on Uncertainty in Artificial Intelligence (UAI-09)*.