Finding a Path is Harder than Finding a Tree

Christopher Meek
Microsoft Research
Redmond WA, 98052-6399
meek@microsoft.com

Abstract

This note shows that the problem of learning an optimal chain graphical model from data is NP-hard for the Bayesian, maximum likelihood, and minimum description length approaches. This hardness result holds despite the fact that the problem is a restriction of the polynomially solvable problem of finding the optimal tree graphical model.

Keywords : Model selection, NP-hardness, Learning algorithms, Graphical models, Learning chains, Learning trees, Learning total orders.

1 Introduction

The problem of learning graphical models has received much attention. In this note, I present a negative result on learning optimal chain graphical models.

The main positive results on learning graphical models are on learning tree graphical models. These have been presented for maximum likelihood (ML) criterion (Edmonds, 1967; Chow and Liu, 1968) and adapted to a Bayesian criterion by Heckerman, Geiger, and Chickering (1995). Two NP-hardness results for learning graphical models have appeared in the literature. Those are the NP-hardness of finding the optimal Bayesian network structure with in-degree greater than or equal to two using a Bayesian optimality criterion (Chickering, 1996) and the problem of finding the ML optimal polytree (Dasgupta, 1999).

In this note, proofs of the hardness of finding an optimal chain graphical models are presented for the maximum likelihood (ML) criterion, the minimum description length (MDL) criterion, and a Bayesian criterion. Unlike the ML hardness result of Dasgupta, I explicitly construct a polynomial sized dataset for the reduction and, unlike the Bayesian hardness result of Chickering (1996), I use a common "non-informative" prior.

The negative result for learning optimal chain graphical models stands in contrast to the positive result on learning tree graphical models. While polynomial learning algorithms exist for the class of tree graphical models, by restricting the class of graphical models to chains, one can make the learning problem become NP-hard.

2 Optimal Graphical Models

Graphical models can be used to obtain an approximate joint distribution over a set of variables from data. In this note, I focus on directed graphical models for a set of discrete variables $\{X_1, \ldots, X_n\}$. One component of a directed graphical model is its directed graphical structure that describes dependencies between the variables. A directed graphical model represents a family of distributions that factor according to the graphical structure G of the directed graphical model, more specifically,

$$P_G(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa_G(X_i))$$

where $pa_G(X_i)$ denotes the possibly empty set of parents of vertex X_i in graph G. The subscript G is omitted when it is clear from context. The most common methods guiding the choice distribution from a family of distributions are maximum likelihood estimation and Bayesian estimation. Given a graphical structure and a set of cases for the variables (also a prior distribution over the distributions in the case of the Bayesian method), these methods provide an approximate joint distribution. For more details on graphical models and estimation see Heckerman (1995).

This leaves open the question of how one should choose the appropriate graphical structure. In the remainder of this section we describe the maximum likelihood (ML), the minimum discrimination length (MDL), and a Bayesian criterion for evaluating directed graphical models given a set of cases D. A value of X_i is denoted by x_i and a value of $pa(X_i)$ is denoted by $pa(x_i)$. The number of cases in D in which $X_i = x_i$ and $pa(X_i) = pa(x_i)$ is denoted by $N(x_i, pa(x_i))$ and the total number of cases in D is denoted by $N = N(\emptyset)$.

The log maximum likelihood score for a graphical model is

$$\begin{aligned} S_{ML}(G, D) &= N \sum_i H_D(X_i | pa(X_i)) \\ &= \sum_i LocalScore_{ML}(X_i, pa(X_i)) \end{aligned}$$

where $H_D(X_i|pa(X_i))$ is the empirical conditional entropy of X_i given its parents, and is equal to

$$-\sum_{X_i, pa(X_i)} \frac{N(x_i, pa(x_i))}{N} \log \frac{N(x_i, pa(x_i))}{N(pa(x_i))}$$

One practical shortcoming of the ML score is that in comparing two models with graphical structure G and G' where G contains a proper subset of the edges of G' the ML score will never favor G. Thus, when using an ML score to choose among models without restricting the class of graphical structures, a fully connected structure is guaranteed to have a maximal score. This is problematic due to the potential for poor generalization error when using the resulting approximation. This problem is often called "overfitting". When using this principle it is best to restrict the class of alternative structures under consideration in some suitable manner.

The minimum description length score can be viewed as a penalized version of the ML score

$$\begin{aligned} S_{MDL}(G, D) &= N \sum_i H_D(X_i|pa(X_i)) - \frac{d \log N}{2} \\ &= \sum_i LocalScore_{MDL}(X_i, pa(X_i)) \end{aligned}$$

where $d = \sum_i (\#(pa(X_i)) \times (\#(X_i) - 1))$ and $\#(Y)$ is used to denote the number of possible distinct assignments of values for a set of variables Y and $\#(\emptyset) = 1$. The penalty term leads to more parsimonious models, thus, alleviating the shortcoming described for the ML score.

Finally, we present a log Bayesian score. We consider a restricted type of prior where we assume a uniform prior on alternative graphs, $P(G) \propto 1$, and the "uninformative" prior over distributions from Cooper and Herskovits (1992).

$$\begin{aligned} S_{Bayes}(G, D) &= \log P(D|G) + \log P(G) \\ &\propto \sum_{i=1}^{n} \log \prod_{pa(x_i)} \frac{(\#(X_i) - 1)!}{(\#(X_i) - 1 + N(pa(x_i)))!} \\ &\qquad \prod_{x_i} N(x_i, pa(x_i))! \\ &\propto \sum_i LocalScore_{Bayes}(X_i, pa(X_i)) \end{aligned}$$

Although not as apparent as in the MDL score, the Bayesian score also has a built-in tendency for parsimony that alleviates the problems of overfitting. The hardness results presented below can be extended to a variety of alternative types of priors including the BDe prior with an empty prior model (see Heckerman et al. 1995).

The problem of finding the optimal directed graphical model for a given class of structures $\mathcal{G}$ and data D is the problem of finding the structure $G \in \mathcal{G}$ that maximizes $S(G, D)$. The important feature of each of the scores is that they can be calculated in terms of a local score for each variable. The structure of the graphical model determines which particular variables are involved in the computation of a local score. Finally, the local score for a variable X_i is only a function of the number of possible assignments of values to the variables X_i and $pa(X_i)$ and the joint counts for X_i and $pa(X_i)$ in the set of cases D.

3 NP-Hardness of finding optimal chains

In this section, we demonstrate that the problem of finding the optimal directed graphical model when we restrict the class of structures to be chains is NP-hard. A chain is a spanning tree in which no vertex has degree higher than two. This result stands in stark contrast to the positive results provided by Edmonds (1967) and Chow and Liu (1968) who show that one can learn the ML optimal tree in polynomial time. Heckerman et al. (1995) have extended these results to finding the Bayesian optimal tree.

To demonstrate the hardness of finding optimal chains we need to formulate the problem as a decision problem. The decision problem version of finding the optimal chain directed graphical model is as follows

> The optimal chain (OC) decision problem: Is there a chain graphical model with score greater than or equal to k for dataset D?

In this section we prove the following theorem.

Theorem 1 *The optimal chain problem is NP-Hard.*

To show this, we reduce the Hamiltonian Path (HP) decision problem to the OC decision problem.

> The HP decision problem: Is there a Hamiltonian path in an undirected graph G?

A Hamiltonian path for an undirected graph G is a non-repeating sequence of vertices such that each vertex in G occurs on the path and for each pair of adjacent vertices in the sequence there is an edge in G. Let the undirected graph $G = \langle V, E\rangle$ have vertex set $V = \{X_1, \ldots, X_n\}$ and edge set E.

The HP decision problem is NP-complete. Loosely speaking, this means that the HP decision problem is as computationally difficult as a variety of problems for which no known algorithm exists that runs in time that is a polynomial function of the size of the input. Theorem 1 indicates that the OC decision problem is at least as difficult as any NP-complete problem. For more information about the HP decision problem and NP-completeness see Garey and Johnson (1979).

We reduce the HP decision problem for G to the OC decision problem by constructing a set of cases D with the following properties;

$$\#(X_i) = \#(X_j) \quad (1)$$

$$LocalScore(X_i, \emptyset) = LocalScore(X_j, \emptyset) = \gamma \quad (2)$$

$$LocalScore(X_i, \{X_j\}) \in \{\alpha, \beta\} \qquad \alpha < \beta \quad (3)$$

$$LocalScore(X_j, \{X_i\}) = LocalScore(X_i, \{X_j\}) \quad (4)$$

$$LocalScore(X_i, \{X_j\}) = \beta \ \text{ iff } \ \{X_i, X_j\} \in E \quad (5)$$

For such a dataset, the problem of the existence of a Hamiltonian path is equivalent to the existence of a chain graphical model with score equal to $k = \gamma + (|V| - 1) \times \beta$ where $|V| = n$ is the number of vertices in the undirected graph G. Thus, if we can efficiently construct a polynomial sized dataset with these properties, we have reduced the HP problem to the OC problem. In other words, we have transformed a general HP decision problem into an OC decision problem. Because the size of the input to the OC problem is a polynomial function of the size of the input for the HP problem, if one can find an algorithm solve the OC problem in polynomial time then all NP-complete problems can be solved in polynomial time.

We construct a dataset for graph G assuming that each variable is ternary to satisfy condition 1. For each pair of vertices X_i and X_j $(i < j)$ for which there is an edge in G we add the following 8 cases in which every variable X_k $(k \neq i, j)$ is zero.

$X_1 \ldots X_{i-1}$	X_i	$X_{i+1} \ldots X_{j-1}$	X_j	$X_{j+1} \ldots X_n$
0...0	1	0...0	1	0...0
0...0	1	0...0	1	0...0
0...0	1	0...0	1	0...0
0...0	1	0...0	2	0...0
0...0	2	0...0	1	0...0
0...0	2	0...0	2	0...0
0...0	2	0...0	2	0...0
0...0	2	0...0	2	0...0

For each pair of vertices X_i and X_j $(i < j)$ for which there is not an edge in G we add the following 8 cases.

$X_1 \ldots X_{i-1}$	X_i	$X_{i+1} \ldots X_{j-1}$	X_j	$X_{j+1} \ldots X_n$
0...0	1	0...0	1	0...0
0...0	1	0...0	1	0...0
0...0	1	0...0	2	0...0
0...0	1	0...0	2	0...0
0...0	2	0...0	1	0...0
0...0	2	0...0	1	0...0
0...0	2	0...0	2	0...0
0...0	2	0...0	2	0...0

For a set of cases constructed as described above, the pairwise counts for a pair of variables X_i and X_j connected by an edge in G are

		X_i		
		0	1	2
X_j	0	$4(n^2 - 5n + 6)$	$4(n-2)$	$4(n-2)$
	1	$4(n-2)$	3	1
	2	$4(n-2)$	1	3

The pairwise counts for a pair of variables X_i and X_j not connected by an edge in G are

		X_i		
		0	1	2
X_j	0	$4(n^2 - 5n + 6)$	$4(n-2)$	$4(n-2)$
	1	$4(n-2)$	2	2
	2	$4(n-2)$	2	2

The marginal counts for each variable are identical, thus, condition 2 is satisfied. There are two types of pairwise count tables, thus, there are at most two values for a given type of pairwise *LocalScore*. It is easy to verify that these two values are not equal to show condition 3 is satisfied. It follows from the symmetry in the two types of pairwise tables and condition 2 that condition 4 is satisfied. Finally, we have constructed the cases to satisfy condition 5. Furthermore, the set of cases is efficiently constructed and has a size which is polynomially bounded by the size of the graph G proving the result.

4 Conclusion

The hardness result presented in this note highlights one potential source of the hardness of NP-Hard problems. By choosing an inappropriate subclass of models one can make an easy problem difficult. Perhaps, by carefully choosing a broader class of models than tree graphical models one can identify interesting classes of graphical models for which the problem of finding an optimal model is tractable.

It is important to note that good heuristics exist for the problem of finding weighted Hamiltonian paths (Karp and Held, 1971). These heuristics can be easily used to identify good quality chain models. In addition, the optimal tree model will have a score at least as large as any chain model so the optimal tree score can be used as a bound for the optimal chain score. This bound can be useful for searching for good chain models.

Finally, the problem of finding an optimal chain graphical model is a version of a general problem called a similarity ordering problem. Given score for pairs of objects $Score(X_i, X_j)$, the *similarity ordering problem* is the problem of identifying a total order on a set of objects such that the sum of the scores for objects adjacent in the total ordering is maximized. A solution to the similarity ordering problem is potentially useful for the visualization of quantitative and qualitative information. When viewing the problem of finding the optimal chain graphical model as a similarity ordering problem, the variables are the objects and the chain corresponds to a total order. One can use the optimal chain graphical model for a dataset to choose an ordering of the variables in a visual display of that dataset. By ordering the variables according to the optimal chain graphical model, one can potentially visually detect and inspect statistically related quantities.

References

Chickering, D. (1996). Learning Bayesian networks is NP-complete. In Fisher, D. and Lenz, H., editors, *Learning from Data*, pages 121–130. Springer-Verlag.

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

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

Dasgupta, S. (1999). Learning polytrees. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence,* Stockholm, Sweden, pages 134–141. Morgan Kaufmann.

Edmonds, J. (1967). Optimum branching. *J. Res. NBS*, 71B:233–240.

Garey, M. and Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman, New York.

Heckerman, D. (1995). A tutorial on learning Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA. Revised November, 1996.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

Karp, R. and Held, M. (1971). The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical Programming*, 1:6–25.