Construction of Large-Scale Bayesian Networks by Local to Global Search

Kyu-Baek Hwang[1], Jae Won Lee[2], Seung-Woo Chung[1], and Byoung-Tak Zhang[1]

[1] School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea
{kbhwang, swchung, btzhang}@bi.snu.ac.kr
[2] School of Computer Science and Engineering, Sungshin Women's University, Seoul 136-742, Korea
jwlee@cs.sungshin.ac.kr

Abstract. Most existing algorithms for structural learning of Bayesian networks are suitable for constructing small-sized networks which consist of several tens of nodes. In this paper, we present a novel approach to the efficient and relatively-precise induction of large-scale Bayesian networks with up to several hundreds of nodes. The approach is based on the concept of *Markov blanket* and makes use of the divide-and-conquer principle. The proposed method has been evaluated on two benchmark datasets and a real-life DNA microarray data, demonstrating the ability to learn the large-scale Bayesian network structure efficiently.

1 Introduction

Bayesian networks [7] are useful tools for classification and data mining. They are particulary helpful because of their ability to give an insight into the underlying nature of the domain that generates the data. Accordingly, Bayesian networks have been applied to many data mining tasks, such as medical diagnosis [9] and microarray data analysis [5]. However, there are some obstacles that prevent the prevalent use of Bayesian networks as data mining tools for real-world problems. One is the scalability of structural learning algorithms. Most existing algorithms are inappropriate to learning large-scale Bayesian networks with hundreds of nodes because of their time and space complexities. In this paper, an efficient structural learning algorithm for such Bayesian networks is suggested.

There have been two kinds of approaches to learning Bayesian networks from data. The first is based on dependency analysis among variables. In this approach, the underlying network structure that produces the data is discovered by some conditional independence tests on variables [12]. The other approach solves the structural learning problem in the viewpoint of optimization [3]. Here, the learning algorithm searches for the network structure which best matches the data. The fitness of a structure is measured by some scoring metrics (e.g. the minimum description length (MDL) score and the Bayesian Dirichlet (BD) score). The search strategy is the core of the second approach. The search space

M. Ishizuka and A. Sattar (Eds.): PRICAI 2002, LNAI 2417, pp. 375–384, 2002.

is super-exponential in the number of variables of the Bayesian network and the problem of finding the best structure is known to be NP-hard [1]. So, heuristic methods such as greedy hill-climbing are used in a general way. The greedy search algorithm [3] is not guaranteed to find the best solution but only a local maximum. Nevertheless, the greedy search algorithm has proven to be quite effective in practice.

Each approach has its own drawbacks. In the former, the conditional independence test may produce the incorrect results and mislead the structural learning process. The latter approach has a tendency to find a dense network structure which may represent the improper causal relationships among variables without careful tuning of the scoring metrics (e.g. the proper assignment of prior probabilities for the BD score or the proper representation of the penalizing term for the MDL score). The exponential time complexity is the problem of both approaches especially in the case of dealing with hundreds of variables.

The recent work in the approach based on dependency analysis is best described in [8]. With the benefit of the concept of *Markov blanket*, unnecessary dependency tests are avoided here. The algorithm described in [8] is very effective for the precise construction of the network structure in polynomial time with restrictions on the maximum size of the Markov blanket. In addition, its randomized variant shows an acceptable performance for more general cases. In the framework of score-based approach, [4] suggested the "sparse candidate" algorithm. This algorithm reduces the size of search space by restricting the possible parents of each node before performing the greedy search and makes it possible to learn the Bayesian network structure with hundreds of variables efficiently. For finding the optimal Bayesian network structure from data, [13] suggested an efficient algorithm using the branch and bound technique.

The proposed algorithm in this paper ("local to global search" algorithm) belongs to the second approach (score-based search). To reduce the global search space, the local structure around each node is constructed before performing the greedy search algorithm. Thus, the proposed algorithm is based on the divide-and-conquer principle. The boundary of the local structure of each node is based on the concept of Markov blanket.

The paper is organized as follows. In Section 2, the concept of the Markov blanket structure as local structural boundary is described. The "local to global search" algorithm is explained in Section 3. We evaluate the proposed algorithm in comparisons with other structural learning algorithms in Section 4. Finally, concluding remarks are given in Section 5.

2 The Markov Blanket Structure

In this section, the Markov blanket and the concept of the *Markov blanket structure* are described. In the following, $\mathbf{X} = \{X_1, ..., X_n\}$ denotes the set of all the variables of interest. The Markov blanket [10] of X_i, $\mathbf{MB}(X_i)$ is the subset of $\mathbf{X} - X_i$ which satisfies the following equation.

$$P(X_i|\mathbf{X} - X_i) = P(X_i|\mathbf{MB}(X_i)). \tag{1}$$

In other words, $\mathbf{MB}(X_i)$ isolates X_i from all the other variables in $\mathbf{X} - X_i$. More than one Markov blanket for a variable could exist. The minimal Markov blanket is called Markov boundary. In this paper, Markov blanket denotes Markov boundary. In the Bayesian network structure G, $\mathbf{MB}_G(X_i)$ is composed of all the parents of X_i, all the children of X_i, and all the parents of each child of X_i excluding X_i itself. We designate the subgraph structure consisting of X_i and $\mathbf{MB}_G(X_i)$ in G as the Markov blanket structure of X_i. Fig. 1 shows an example Markov blanket structure of a node in the Bayesian network.

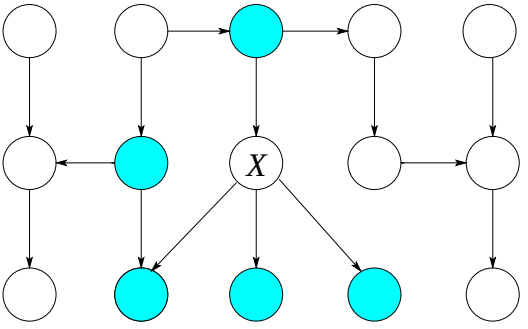

Fig. 1. An example Bayesian network structure. The shaded nodes correspond to the members of $\mathbf{MB}(X)$. The subgraph consisting of X and all the shaded nodes is the Markov blanket structure of X.

3 The "Local to Global Search" Algorithm

Before giving an explanation of the "local to global search" algorithm, we briefly describe the general greedy search algorithm for structural learning of the Bayesian network. The general greedy search algorithm proceeds as follows.

- Generate the initial Bayesian network structure G_0 (an empty structure or a randomly generated structure).
- For $m = 1, 2, 3, ...$, until convergence.
  - Among all the possible local changes (insertion of an edge, reversal of an edge, and deletion of an edge) in G_{m-1}, the one that leads to the largest improvement in the score is performed. The resulting graph is G_m.

The stopping criterion is when the score of G_{m-1} is equal to the score of G_m. At each iteration of the greedy search algorithm when learning the Bayesian network with n nodes, about $O(n^2)$ local changes should be evaluated to select the best one. In the case of hundreds of variables, the cost for these evaluations becomes very expensive.

The key idea of the "local to global search" algorithm is to reduce the search space considered in the general greedy search algorithm by constructing the

Markov blanket structure of each node in advance. The "local to global search" algorithm proceeds as follows.

- Generate the initial Bayesian network structure G_0 (an empty graph).
- Loop for $m = 1, 2, 3, ...$ until convergence.
  1. Construct the Markov blanket structure of each node based on G_{m-1} and the data D.
  2. Merge all the Markov blanket structures into a graph H_m (H_m could have directed cycles).
  3. In H_m, all the edges that do not constitute any directed cycle are fixed as valid edges in G_m.
  4. Perform the general greedy search algorithm to find a good network structure G_m which is a subgraph of H_m.

In the above algorithm, the general greedy search algorithm (Step 4) and the construction of the Markov blanket structure of each node (Step 1) are performed alternately. The stopping criterion of the "local to global search" algorithm is when the score of G_{m-1} is greater than or equal to the score of G_m.

The method of constructing the Markov blanket structure for X_i in Step 1 is as follows.

- For all $X_j \subset \mathbf{X} - X_i - \mathbf{MB}_{G_{m-1}}(X_i)$, the *conditional mutual information*, $I(X_i; X_j | \mathbf{MB}_{G_{m-1}}(X_i))$ is calculated to select the candidate members for the Markov blanket of X_i.
- The general greedy search algorithm is performed on the selected candidate members and X_i to construct the Markov blanket structure of X_i. When performing the general greedy search algorithm, the Markov blanket structure of each node in G_{m-1} is preserved.

The conditional mutual information [2] measures the dependency between X_i and X_j given the value of $\mathbf{MB}_{G_{m-1}}(X_i)$ and is calculated as follows:

$$\begin{aligned} &I(X_i; X_j | \mathbf{MB}_{G_{m-1}}(X_i)) \\ &= I(X_i; X_j, \mathbf{MB}_{G_{m-1}}(X_i)) - I(X_i; \mathbf{MB}_{G_{m-1}}(X_i)). \end{aligned} \tag{2}$$

In (2), $I(X_i; \mathbf{MB}_{G_{m-1}}(X_i))$ is unnecessary because it does not depend on X_j and the only necessary term to select the candidate members is calculated as

$$\begin{aligned} I(X_i; X_j, \mathbf{MB}_{G_{m-1}}(X_i)) \; &= \sum \hat{p}(X_i, X_j, \mathbf{MB}_{G_{m-1}}(X_i)) \\ &\cdot \log \frac{\hat{p}(X_i, X_j, \mathbf{MB}_{G_{m-1}}(X_i))}{\hat{p}(X_i)\hat{p}(X_j, \mathbf{MB}_{G_{m-1}}(X_i))}. \end{aligned} \tag{3}$$

Here, $\hat{p}(\cdot)$ denotes the empirical probability calculated from the data. The selection of candidate members may be guided by some threshold on the conditional mutual information value or by some restrictions on the size of the Markov blanket. Once the candidate members for the Markov blanket are selected, the general greedy search algorithm searches for the good Markov blanket structure.

The key point of the "local to global search" algorithm lies in Step 2 and 3. Merging all the Markov blanket structures and fixing the edges that do not form any directed cycle reduce the great amount of the evaluation cost of the greedy search algorithm in Step 4. However, fixing the edges found in Step 1 can also lead to the deterioration of the score. And the above algorithm is not guaranteed to find even the local maxia. The reason is as follows. The greedy search algorithm usually employs the decomposable scoring metric such as the BD (Bayesian Dirichlet) score [6] and the MDL (minimum description length) score [3]. The decomposable score has the property such that,

$$Score(G, D) = \Sigma_i Score(X_i|\mathbf{Pa}_G(X_i), D), \tag{4}$$

where G is the Bayesian network structure, D is the data, and $\mathbf{Pa}_G(X_i)$ is the parents of X_i in G. Because the score of the network can be decomposed into the score of each node, the increase in the score of each node also increases the score of the network. In the "local to global search" algorithm, the score of each node can be degraded through merging the Markov blanket structures into the global network structure H_m because of the possible changes of its parents. This is the trade-off between speed and accuracy. However, in many domains, the initial Bayesian network structure is an empty graph due to the lack of the domain knowledge and a small number of iterations of the "local to global search" algorithm could find the appropriate Bayesian network structure rapidly.

Time Complexity

We now describe the time complexity of the "local to global search" algorithm. In Step 1, in order to determine the candidate members for the Markov blanket of X_i, the conditional mutual information test is done for all other nodes. This takes time of $O(n \cdot (M \cdot |X_i| \cdot |X_j| \cdot |\mathbf{MB}_{G_{m-1}}(X_i)|))$. Here, M is the number of instances in the data D, and $|\cdot|$ denotes the cardinality of a variable or the set of variables. $|X_i| \cdot |X_j| \cdot |\mathbf{MB}_{G_{m-1}}(X_i)|$ can be regarded as a constant with the appropriate restrictions on the maximum size of the Markov blanket, k. And the selection of candidate members for the Markov blanket is done for all the nodes $\{X_1, ..., X_n\}$. Therefore, the selection of candidate members for the Markov blanket takes time of $O(n^2 M)$. The next is the greedy search procedure to determine the Markov blanket structure of a node. At each iteration of the greedy search algorithm, the number of possible local changes is bounded by $O(k^2)$. With the assumption of the moderate maximum size of the Markov blanket ($k \leq 20$), the greedy search procedure takes not so much time.

Step 2 and 3 merges all the Markov blanket structures into a global network and the time bound is $O(n)$.

In Step 4, the greedy search algorithm evaluates $O(n^2)$ local changes at each iteration. For large n (e.g. $n \geq 800$), the conventional greedy search algorithm takes very much time to search for the good network structure. In the "local to global search" algorithm, this step takes not so much time because of the greatly reduced search space through Step 1, 2, and 3.

4 Experimental Evaluation

To evaluate the performance of the "local to global search" algorithm, two kinds of artificial datasets and a real-life microarray data were used. In the experiments, we compared the "local to global search" algorithm with the general greedy search algorithm and the "sparse candidate" algorithm [4] in the respect of the learning speed and the accuracy of the learned results. As the scoring metric, the BD (Bayesian Dirichlet) metric with uninformative prior [6] was adopted. To ensure fairness in the comparison, the same basic module for the greedy search procedure was used in all three algorithms. The experiments were performed on a PentiumIII 1GHz machine with 256MB RAM. In the "local to global search" algorithm, the maximum size of the Markov blanket was restricted by a constant k.

Artificial Datasets

Table 1 shows the characteristics of two Bayesian networks that produced artificial datasets. Although these networks have only tens of nodes, the experiments on the artificial datasets make it possible to compare the accuracy of the algorithms in the respect of reconstructing the precise network structure. Five artificial datasets of 10000 instances were generated from each of these Bayesian networks.

Table 2 compares three structural learning algorithms in the respect of the learning speed. The "sparse candidate" algorithm is much faster than other two algorithms. The "local to global search" algorithm with $k = 5$ is faster than the general greedy search algorithm. However, the "local to global search" algorithm with $k = 10$ is not so much faster than the general greedy search algorithm and even slower on the Hailfinder datasets. This is due to the conditional mutual information test for the selection of the candidate Markov blanket members in the "local to global search" algorithm. Most variables in the Hailfinder network have more categorical values than the variables in the ALARM network. The Hailfinder network even has a variable which has 11 categorical values. Accordingly, the conditional mutual information tests on the Hailfinder datasets take extremely long time.

Table 3 compares three structural learning algorithms in the respect of the accuracy of the learned results (in the likelihood score). The general greedy

Table 1. The characteristics of two Bayesian networks used to generate artificial datasets. These networks are generally used as benchmarks to test the performance of structural learning algorithm for Bayesian networks.

Network	# of nodes	# of edges
ALARM network	37	46
Hailfinder network	56	66

Table 2. The comparison of the learning speed (in seconds) of three structural learning algorithms. The left (right) table represents the results on the five datasets generated from the ALARM (Hailfinder) network. (**GG** = general greedy search algorithm, **SC**k = "sparse candidate" algorithm with maximum candidate parents size of k, **LG**k = "local to global search" algorithm with Markov blanket size of k)

	GG	**SC5**	**SC10**	**LG5**	**LG10**
A1	143	26	39	42	134
A2	146	30	41	37	172
A3	129	27	39	46	128
A4	150	26	41	45	118
A5	136	27	40	41	133
Avg.	140.8	27.2	40.0	42.2	137.0

	GG	**SC5**	**SC10**	**LG5**	**LG10**
H1	352	55	89	191	959
H2	351	56	89	196	1625
H3	370	56	90	276	2015
H4	366	59	88	278	1663
H5	362	57	90	290	1450
Avg.	360.2	56.6	89.2	246.2	1542.4

Table 3. The comparison of the accuracy (in the likelihood score of the learned networks) of three structural learning algorithms. The left (right) table represents the results on the five datasets generated from the ALARM (Hailfinder) network. (**GG** = general greedy search algorithm, **SC**k = "sparse candidate" algorithm with maximum candidate parents size of k, **LG**k = "local to global search" algorithm with Markov blanket size of k)

	GG	**SC5**	**SC10**	**LG5**	**LG10**
A1	-9.483	-9.764	-9.577	-9.686	-9.577
A2	-9.524	-9.847	-9.597	-9.610	-9.765
A3	-9.536	-9.847	-9.609	-9.750	-9.630
A4	-9.466	-9.790	-9.546	-9.519	-9.597
A5	-9.541	-9.815	-9.589	-9.639	-9.614

	GG	**SC5**	**SC10**	**LG5**	**LG10**
H1	-49.685	-49.762	-49.714	-49.772	-49.813
H2	-49.707	-49.790	-49.742	-49.805	-49.788
H3	-49.695	-49.777	-49.724	-49.764	-49.847
H4	-49.677	-49.751	-49.721	-49.861	-49.789
H5	-49.748	-49.829	-49.779	-49.819	-49.845

search algorithm shows the best accuracy in the respect of the likelihood score of the learned Bayesian networks. The "sparse candidate" algorithm and the "local to global search" algorithm are slightly worse than the general greedy search algorithm. Fig. 2 compares the accuracy of each algorithm in the respect of the ability of reconstructing the correct network structure. In every case, the general greedy search algorithm finds the more accurate structures than other two algorithms. On the ALARM datasets, the performances of the "sparse candidate" algorithm and the "local to global search" algorithm are comparable. On the Hailfinder datasets, the "sparse candidate" algorithm shows a slightly better performance than the "local to global search" algorithm.

DNA Microarray Data

To test the ability of the "local to global search" algorithm for the construction of large-scale Bayesian networks, the NCI60 dataset [11] was used. The NCI60 dataset consists of 60 human cancer cell lines from 9 kinds of cancers, that is, colorectal, renal, ovarian, breast, prostate, lung, and central nervous system

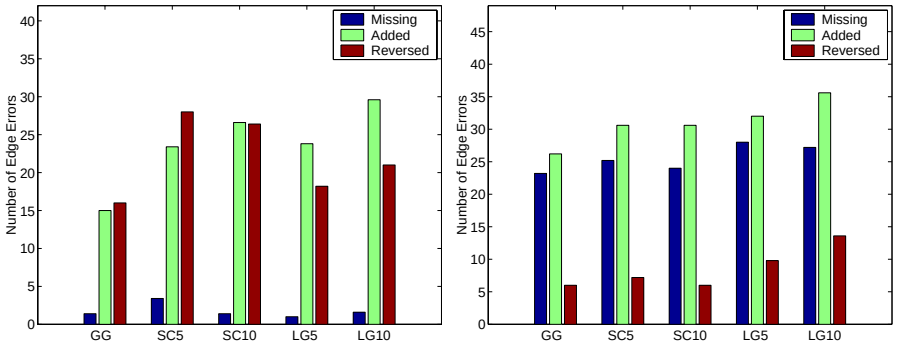

Fig. 2. The average edge errors for the ALARM network datasets (left) and the Hailfinder network datasets (right). In every case, the general greedy search algorithm finds the better structures than other two algorithms. On the ALARM datasets, the performances of the "sparse candidate" algorithm and the "local to global search" algorithm are comparable. In the case of the Hailfinder datasets, the "sparse candidate" algorithm shows a slightly better performance than the "local to global search" algorithm. (**GG** = general greedy search algorithm, **SC**k = "sparse candidate" algorithm with maximum candidate parents size of k, **LG**k = "local to global search" algorithm with Markov blanket size of k)

origin cancers, as well as leukaemias and melanomas. On each cell line, the gene expression pattern is measured by cDNA microarray of 9703 genes including ESTs. And 1400 chemical compounds were tested on the 60 cell lines. From these attributes, genes and drugs (chemical compounds) that have more than 3 missing values across 60 samples as well as unknown ESTs were eliminated for robust analysis. Consequently, the analyzed NCI60 dataset includes 60 samples with 890 attributes (805 gene expression levels, 84 drug activities, and one additional variable for the kind of cancer). Hence, the task is to learn the structure of Bayesian network with 890 nodes. All the attribute values are continuous and discretized into three categorical values (low, normal, and high) according to their mean values and the standard deviations across 60 data samples. With varying threshold values for discretization, three datasets (NCI1, NCI2, and NCI3) were made.

Table 4 shows the comparison of the "sparse candidate" algorithm and the "local to global search" algorithm on these datasets. The general greedy search algorithm was inapplicable to the NCI60 dataset because of its time and space complexity. The "local to global search" algorithm with $k = 5$ is the fastest. In the respect of the accuracy, the "local to global search" algorithm with $k = 8$ shows a slightly better performance than others.

5 Conclusion

We presented a novel method for structural learning of large-scale Bayesian networks. It is used as a component learning algorithm in the framework of

Table 4. The comparison of the learning speed (in seconds) and the accuracy (in the likelihood score of the learned results) of the "sparse candidate" algorithm and the "local to global search" algorithm on the NCI60 dataset. The left (right) table represents the learning time (the likelihood score). (**SC**k = "sparse candidate" algorithm with maximum candidate parents size of k, **LG**k = "local to global search" algorithm with Markov blanket size of k)

	SC5	**SC8**	**LG5**	**LG8**
NCI1	7568	8678	7123	10987
NCI2	7542	8443	7089	10223
NCI3	7345	8789	7343	11092

	SC5	**SC8**	**LG5**	**LG8**
NCI1	-777.35	-777.01	-779.08	-773.12
NCI2	-768.54	-768.34	-769.76	-768.23
NCI3	-787.35	-788.01	-787.20	-787.10

greedy hill-climbing, and avoids many unnecessary computations by constructing the Markov blanket structure of each node in advance. Since most of the network structure is learned through economic local search procedures, the space and time complexities of global search is greatly reduced.

Comparative analysis shows that the proposed method significantly outperforms the conventional greedy search algorithm in terms of the learning speed. The accuracy of the learned results in terms of the likelihood score or the ability of reconstructing the original structure is not so much degraded compared to the general greedy search algorithm. One disadvantage of the "local to global search" algorithm is that the performance is severely degraded when dealing with the dataset which has variables of large cardinalities. In comparisons with the state-of-the-art techniques for learning large-scale Bayesian networks (e.g. the "sparse candidate" algorithm [4]), the proposed method shows a slightly better performance in the accuracy and the learning speed although the experiments are confined on one real-life dataset. The choice of the parameter k of the "local to global search" algorithm does not seem to affect much the accuracy of the learned results.

As a conclusion, the proposed method is suitable for learning the large-scale Bayesian network with hundreds of variables efficiently from the data which has variables with moderate cardinalities ($2 \sim 4$).

Acknowledgements. This work was supported by the Korean Ministry of Education and the Ministry of Science and Technology under the BK21-IT, BrainTech, and IMT-2000 Programs.

References

1. Chickering, D.M.: Learning Bayesian networks is NP-complete. In: Fisher, D., Lenz, H.-J. (eds.): Learning from Data: Artificial Intelligence and Statistics V. Springer-Verlag, Berlin Heidelberg New York (1996) 121-130
2. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)

3. Friedman, N., Goldszmidt, M.: Learning Bayesian networks with local structure. In: Jordan, M.I. (ed.): Learning in Graphical Models. MIT Press, Cambridge (1999) 421-459
4. Friedman, N., Nachman, I., Pe'er, D.: Learning Bayesian network structure from massive datasets: the "sparse candidate" algorithm. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI) (1999) 206-215
5. Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. Proceedings of the Fourth Annual International Conference on Computational Biology (RECOMB) (2000) 127-135
6. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning **20**(3) (1995) 197-243
7. Heckerman, D.: A tutorial on learning with Bayesian networks. In: Jordan, M.I. (ed.): Learning in Graphical Models. MIT Press, Cambridge (1999) 301-354
8. Margaritis, D., Thrun, S.: Bayesian network induction via local neighborhoods. Advances in Neural Information Processing Systems 12 (2000) 505-511
9. Nikovski, D.: Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. IEEE Transactions on Knowledge and Data Engineering **12**(4) (2000) 509-516
10. Pearl, J: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
11. Scherf, U. et al.: A gene expression database for the molecular pharmacology of cancer. Nature Genetics **24** (2000) 236-244
12. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. 2nd edn. MIT Press, Cambridge (2000)
13. Suzuki, J.: Learning Bayesian belief networks based on the minimum description length principle: an efficient algorithm using the B & B technique. Proceedings of the Thirteenth International Conference on Machine Learning (ICML) (1996) 462-470