MML Inference of Large Margin Oblique Decision Trees

Peter J. Tan and David L. Dowe

School of Computer Science and Software Engineering, Monash University,
Clayton, Vic 3800, Australia
ptan@bruce.csse.monash.edu.au

Abstract. Univariate decision trees only permit internal decision node splits based on the value of one attribute. Due to the greater expressibility and flexibility of having multivariate functions in internal nodes, multivariate decision trees have the potential to render complex underlying concepts in a neater and more concise tree than univariate trees. But the dramatical increase of function space also poses new search challenges in constructing multivariate tree inference schemes. Improving generalization ability is amongst the major challenges in inferring multivariate trees. In this paper, we propose a multivariate decision tree inference scheme by using the minimum message length (MML) principle (Wallace and Boulton, 1968; Wallace and Dowe, 1999). The scheme uses MML coding as an objective (goodness-of-fit) function on model selections and a simple evolution strategy to perform the search. We test our multivariate tree inference scheme on data sets from the UCI machine learning repository and compare with the decision tree programs C4.5 and C5. The preliminary results show that on average and on most data-sets, MML oblique trees clearly perform better than both C4.5 and C5 on both "right"/"wrong" accuracy and probabilistic prediction - furthermore, manage to do this with smaller trees i.e., less leaf nodes.

1 Introduction

Decision tree algorithms have been successfully implemented in solving a wide range of machine learning and data mining problems. While there are a number of excellent decision tree learning algorithms such as CART [3] and C4.5 [14], much research effort has been continuously directed to finding new and improved tree induction algorithms. Decision tree learning algorithms recursively partition the data into a finite number of homogeneous subsets (leaf nodes) which have separate inference models. Most decision tree algorithms only test on one attribute at internal nodes, and these are often referred to as univariate trees. One of the obvious limitations of univariate trees is that their internal nodes can only separate the data with hyperplanes perpendicular to the co-ordinate axes. This limitation reduces the expressive power of decision trees. If the number of training data is limited, it can often lead to the comparatively poor rendering of many classes of concepts. Multivariate decision tree algorithms attempt to generate decision trees by employing discriminant functions at internal nodes with

more than one attribute. With these discriminant functions, multivariate trees are able to partition the instance space with hyperplanes having arbitrary slopes - rather than only parallel to the co-ordinate axes. Such flexibility alleviates the problem of inefficient representation. As a result, the resultant multivariate trees tend to be more concise than their univariate counterparts. The difference between multivariate decision trees and univariate trees is illustrated in Fig. 1. It demonstrates a case where, to classify the data purely, a univariate tree would need at least three cuts while a multivariate tree would require only a single cut.

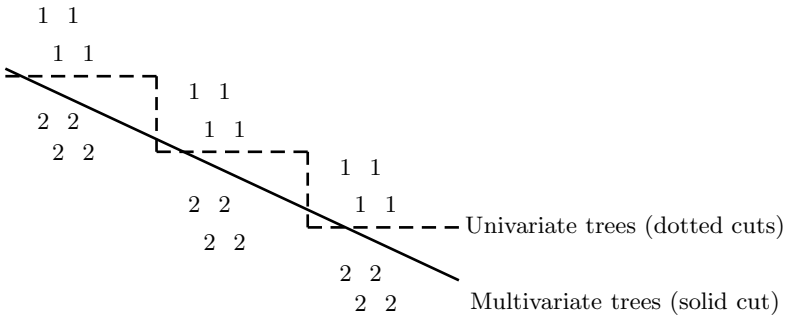

Fig. 1. The difference between univariate and multivariate decision trees

To keep the computational time reined in, most of the multivariate tree systems only allow the use of linear functions - such trees are often referred to as oblique decision trees. We propose a oblique decision tree inference scheme by using the minimum message length (MML) principle [22, 24, 23, 19]. The test results show that our new oblique decision tree inference algorithms are able to find smaller trees while maintaining better or comparable accuracy compared to the standard univariate decision tree schemes C4.5 and C5 [14].

The sections are organized as follows. Related works on multivariate decision trees and the motivation for the new scheme for multivariate trees are discussed in the next section. Sections 3 and 4 briefly describes MML inference of decision and the details of the new, MML oblique decision tree inference scheme. Section 5 shows the experimental results. Finally, we analyze the test results and give a conclusion.

2 Existing Multivariate Decision Tree Schemes

Although most of the current decision tree learning algorithms concentrate on univariate trees, the benefits of multivariate decision trees have long been known and many multivariate decision tree schemes have appeared in the machine learning literature. Most of them are oblique decision trees, whose internal nodes test

on (only) linear combinations of attributes. CART [3] was introduced by Breiman et al. and probably is the first oblique decision tree system. CART implemented a deterministic search algorithm which can be summarized as follows.

To find a split of the form $\sum_{i=1}^{d} w_i x_i \leq w_{d+1}$ at a leaf Node L,

Normalize data for all d attributes, set T=0 and given a fitness function G(L) and real number ϵ

1. T=T+1
2. Let i=0, $v = \sum_{i=1}^{d} w_i x_i$, $\gamma \in \{-0.25, 0, 0.25\}$ and $\delta \in R$
3. Search for the δ, γ that maximizes the fitness function G(L) of the split $v - \delta(w_i + \gamma) \leq w_{d+1}$
4. Let δ^* and γ^* be the optimal values from step 3, update w_i and w_{d+1} so that $w_i = w_i - \delta^*$, $w_{d+1} = w_{d+1} - \delta^* \gamma^*$.
5. $i = i + 1$; if $i \leq d$, go to step 3
6. Search for the best w_{d+1} that maximizes G(L)
7. If $G(L)^T - G(L)^{T-1} > \epsilon$, go to step 1

The idea of the above heuristic is to perturb the hyperplane in one dimension at each iteration to search for the optimal hyperplane. However, as pointed out in [9], there are two limitations to the CART algorithm. Firstly it is deterministic so that when the algorithm is trapped in a local minimum, it is unable to escape. Secondly, G(L) is not guaranteed to converge, so the search process may run indefinitely unless a pre-set stopping condition is imposed. To resolve these problems, several algorithms take new randomized approaches for inferring oblique decision trees. Simulated Annealing of Decision Trees (SADT) [7] implemented simulated annealing to search for optimal oblique splits at each tree node. Murthy proposed a new refined search algorithm called OC1 [9], in which random perturbations of the hyperplane are performed to escape from local minima. The OC1 algorithm can be summarized as follows.

Given a leaf node L, a fitness function G(L) and a (run bound) integer J,
Set T=0

1. Choose a random hyperplane, H
2. Search for the optimal hyperplane, H, by using a deterministic search algorithm similar to CART
3. Repeat step 2 until value of fitness function G(L) does not improve
4. Randomly perturb the hyperplane, H, in one dimension
5. T=T+1; if $T \leq J$ then go to step 2

SADT and OC1 combine deterministic searches and random processes to avoid the local minima traps. The main drawback of these approaches is the substantial increase of time complexity.

Another important issue of multivariate decision tree schemes is to select a subset of attributes included in the test at each node. To decide the optimal subsets of features in tests at each node, the algorithms must find the trade-off between the complexity and the goodness of fit. Due to the considerable increase

of candidate splits compared to cut-points in univariate trees, the main challenge for these oblique tree schemes is to differentiate the complexities of splits and quantify the trade-offs. While most of the schemes focus on the search issue, they rely on pruning to improve the generalization accuracy and take some ad hoc approaches to the complexity. Bennett and Blue [1] proposed to infer multivariate decision trees by using support vector machines (SVMs) [21] at internal nodes. Their algorithm[1] used the structural risk minimization (SRM) principle [21] to find the optimal splits. The scheme generates very simple decision trees - one with three non-linear multivariate decisions so that they are more like hybrids of SVMs than decision trees. However, the trade-off between the topological complexity of the tree and the complexity of the decisions remains unresolved in their scheme. We address this problem by using MML inference. We use the sum of the message length of a split and the encoded data given the split to find the trade-off between the complexity and the goodness of fit. The details of our scheme is detailed in the following sections. This paper is also built upon our previous work [20].

3 MML Inference of Multivariate Decision Trees

MML inference [22, 24, 23, 19] has been successfully implemented in [25] to infer univariate decision trees (refining [15]) and in [11, 18, 19] to infer univariate decision graphs, with the most recent decision graphs [18, 19] clearly out-performing *both* C4.5 and C5 [14] on *both* real-world and artificial data-sets on a range of test criteria - we had better "right"/"wrong" accuracy, substantially better probabilistic score and [19, Table 4] fewer leaf nodes. In this paper, we use the MML principle to infer multivariate decision trees. The multivariate decision tree scheme proposed here generalizes earlier MML decision tree work and should avail us of advantages described in Section 2 and illustrated in Fig. 1.

3.1 Minimum Message Length (MML) Principle

The Minimum Message Length (MML) Principle [22, 24, 23, 19] provides a guide for inferring the best model given a set of data. If a set of data is to be transmitted, it can either be transmitted directly (as it is); or alternatively a theory can be inferred from the data, then the set of data is transmitted with the help of the theory. Thus, the transmitted message is composed of the following two parts:

I. the description of the theory, or hypothesis, H

II. the data, D, explained with help of the theory: D given H, or $D \mid H$.

From Bayes's theorem,

we know that Pr(D)Pr(H|D) = Pr(H&D) = Pr(H)Pr(D|H)

So Pr(H|D) $= \frac{Pr(H)Pr(D|H)}{P(D)} \ldots\ldots (1)$,

where Pr(H&D) is the joint probability of D and H, Pr(H) is the prior probability of the hypothesis H, Pr(D) is the marginal probability of the data D, Pr(D|H)

is the likelihood function of D given H and Pr(H|D) is the posterior probability of H given observed data D. From (1), we get that

$$-\log Pr(H \mid D) = -\log Pr(D \mid H) - \log Pr(H) + \log Pr(D)$$

To maximize Pr(H|D) is equivalent to minimizing $-\log Pr(H \mid D)$.
Because $\log Pr(D)$ is a constant, we can ignore it and seek a minimum of

$$-\log P(D \mid H) - \log P(H).$$

Thus the hypothesis with the minimum two-part message length can be said to be the model of best fit for the given data. For details of the MML Principle, see [22, 24, 23, 11]. For a comparison between MML and the subsequent Minimum Description Length (MDL) principle[16], see, e.g., [23] and other articles in that 1999 special issue of the *Computer Journal*.

MML and the subsequent Minimum Description Length (MDL) principle [16, 8] (see also [23] for a survey) are widely used for model selection in various machine learning problems, and both can be thought of as operational forms of Ockham's razor [10]. In practice, MML and MDL work very well on inference of decision trees. Among efforts that have been put into the development of tree-based classification techniques in recent years, Quinlan and Rivest [15] proposed a method for inferring decision trees using MDL. Wallace and Patrick subsequently [25] presented a refined coding scheme for decision trees using MML in which they identified and corrected some errors in Quinlan and Rivest's derivation of the message length, including pertaining to the issue of probabilistic prediction. Wallace and Patrick also introduced a "Look Ahead" heuristic of arbitrarily many ply for selecting the test attribute at a node. We re-use the Wallace and Patrick decision tree coding [25] as part of the coding scheme for our new oblique decision tree program. For further details of the implementation, please see [18].

3.2 Encoding an internal split using a linear discriminant function

To infer oblique decision trees by the MML principle, we extend the Wallace and Patrick decision tree coding scheme. The new MML decision tree coding scheme is able to encode an internal split using a linear discriminant function. Firstly, the data falling at an internal node is scaled and normalized so that every data item falls within a D-dimensional unit hyper-cube, where D is the number of input attributes. A linear decision function d(w, x, b)=0 is written as $(\sum_{i=1}^{D} w_i x_i) + b = w \cdot x + b = 0$ where $w, x \in R^D$, $\cdot$ denotes the dot (or scalar) product, and the scalar b is often called the bias. The data is divided into two mutually exclusive sets by the following rules.

If $d(w, x_j, b) > 0, j \in [1, N]$, then x_j is assigned to set I (denoted '1' or '+').
If $d(w, x_j, b) < 0, j \in [1, N]$, then x_j is assigned to set II (denoted '2' or '−').

To encode the hyperplane is equivalent to transmitting the vector w and the bias b. Suppose the desired value of the vector w is w_c. If we state w_c exactly (to infinite precision), it will cost infinitely many bits of information in the first part of the message. So instead, we attempt to state a set of vectors $\Lambda(\theta), \theta \in (0, \frac{\pi}{2})$, which is defined as

$\Lambda(\theta) = \{w : \arccos(\frac{w \cdot w_c}{\|w\| \cdot \|w_c\|}) < \theta\}$

This is the set of vectors which form an angle less than θ with the optimal vector w_c as illustrated in Fig. 2. The probability that a randomly picked vector falls into the set is given by $\frac{V_\theta}{V_T}$, where V_θ is the volume of a partial sphere of radius θ and V_T is the total volume of the unit sphere. The value of $\frac{V_\theta}{V_T}$ is given [17] by $(\sin\theta)^{2(D-1)}$, so the information required to specify the set of the vectors is $-\log((\sin\theta)^{2(D-1)})$.

By specifying one data point on each side of the hyperplane h_c, two hyperplanes which are parallel to the decision surface d(w,x,b)=0 are also defined. We denote these two hyperplanes as h_+ and h_-. These(h_+ and h_-) and the other boundaries of the unit cube form a hyper-rectangle as shown in Fig. 2a.

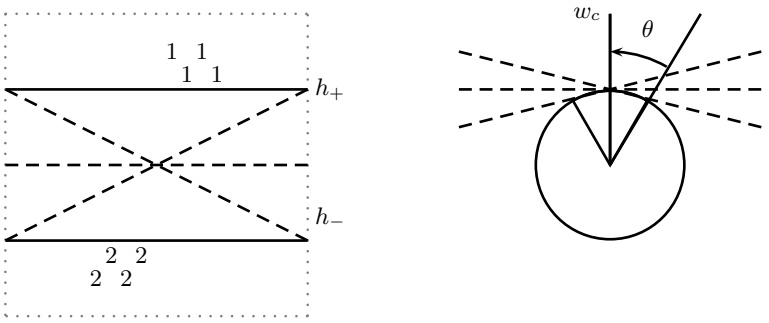

Fig. 2. The set of hyperplanes (Fig. 2a) defined by vector $w \in \Lambda(\theta)$ and (Fig. 2b) a partial sphere of radius θ formed by $w \in \Lambda(\theta)$

We want to work out the value of the θ so that the hyperplanes specified by vectors in the set $\Lambda(\theta)$ do not intersect with the hyperplane h_+ and h_- within the hypercube. We can imagine a rectangle whose length of one side is the distance between h_+ and h_- and whose length of the other side is $\sqrt{D}$, which is the longest diagonal in a D-dimensional unit cube. As {x: kwx+kb=0} $\equiv$ {x: wx+b=0} for any non-zero k, we can choose the w so that the margin between h_+ and h_- is equivalent to $\frac{2}{\|w\|}$. As shown in the Figure 3, given the margin $\frac{2}{\|w\|}$, if $\theta < \alpha$, where $\alpha = \arcsin(\frac{2}{\sqrt{D\|w\|^2+4}})$, it is easy to show that the hyperplane h_w defined by the vector w does not intersect with hyperplanes h_+ and h_- within the D-dimensional hyper cube (from Fig. 2a). As such, the message length needed to define a representative optimal hyperplane is: $M = -2(D-1)\log(\frac{2}{\sqrt{D\|w\|^2+4}}) + \log\binom{N}{2}$, where N is the number of data.

3.3 Search for the optimal hyperplane

In order to perform faster searches for optimal multivariate splits, we do not use the search heuristic used in OC1 [9] and SADT [7]. Instead, we implement a

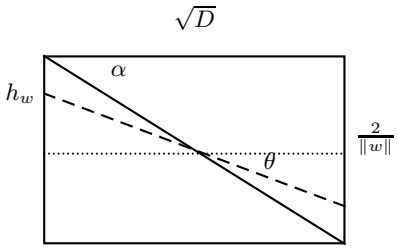

Fig. 3. The upper bounds of θ

simple evolution strategy as the preliminary search heuristic for our scheme. A similar approach has appeared in [4], in which promising results were reported. The search process in our scheme can be summarized as follows. Assuming the linear discriminant function in our scheme takes the form $\sum_{i=1}^{d} w_i x_i < w_{d+1}$, for each leaf node L, let M(unsplit) denote the message length of the node L while the node is unsplit, and let M(T) denote the message length of the subtree when node L is split by vector w^T at round T. The algorithm searches for the best vector w in the following steps:

Initialize T=0, input R, MaxP, M(unsplit)
1. Re-scale the coefficients of the vector w such that $\sum_{i=1}^{d} w_i^2 = 1$.
2. With $v \sim N(0,1)$, randomly pick $j \in [1, d+1]$, $w_j^{T+1} = w_j^T + v$.
3. if $M(T+1) < M(T)$, go to step 5
4. $w_j^{T+1} = w_j^T$
5. T=T+1; if $T < R$, go to step 1.
6. Randomly select d (in this paper, d is limited to 2 or 3) attributes
7. P=P+1; if $P < MaxP$, go to step 1
8. if $M(R) < M(unsplit)$, return w and M(R), otherwise return null and M(unsplit).

It is easy to see (from steps 2 and 7) that the search process is non-deterministic, and thus our algorithms are able to generate many different trees. As such, our algorithms can be extended to take advantage of this by choosing the best one among these trees or averaging results from these trees.

4 Experiments

To evaluate our new oblique decision tree scheme, we run experiments on nine data sets selected from the UCI Machine Learning Repository [2]. The performance of our scheme is compared with those of the C4.5 and C5 [14]. In addition to the traditional right/wrong accuracy, we are also keen to compare the probabilistic performance [19, sec 5.1] [6, 5, 10, 18] of the learning algorithms. In a

lot of domains, like oncological and other medical data, not only the class predictions but also the probability associated with each class is essential. In some domains, like finance, (long term) strategies heavily rely on accurate probabilistic predictions. We discuss using decision trees as probabilistic models in the following section.

4.1 Comparing and scoring probabilistic predictions

Decision trees are often used as classifiers in many machine learning problems. In the case in which the target attribute is multinomial, each leaf node in a tree is given a class label corresponding to the class with the highest inferred probability for this node. However, the multinomial distribution given by a model in each leaf node can also be interpreted as a probabilistic (prediction) models. In this way, decision trees are not only classifiers, but they can also serve as probabilistic prediction model. Provost and Domingos [13] showed that with some modifications, tree inductions programs can produce very high quality probability estimation trees (PETs). Perlich, Provost and Simonoff [12] also observed that for large data sets, tree induction often produces probability-based rankings that are superior to those generated by logistic regression. To compare probabilistic prediction performances, we propose a new metric called the related (test data) code length (RCL),which is defined as

$$RCL = -\frac{\sum_{i=1}^{n} \log(p_i)}{n \log(M)}$$

, where n is the total number of the test data, M is the number of classes in the target attribute and p_i is the probability assigned to the real class associated with the test instance i by the model.

This metric can be interpreted as follows. For each instance of the test data set, a probability distribution for the target classes is given by the model. Then each instance is encoded with a code length corresponding to the probability assigned to the real class associated with the test instance by the model. If there are M classes of the target attribute, assuming a uniform prior, each instance in the test data set will be encoded with log(M) bits by the null theory, i.e., not attempting to infer a probability distribution and using an identical code length for each class. The related test data code length (RCL) is equivalent to the average code length of the test data encoded by a model divided by the average code length encoded by the null theory. Thus the smaller the RCL is, the better the performance of a model on probabilistic prediction. If RCL is larger than 1, a model performs worse than the null theory on probabilistic prediction.

4.2 Data sets

Nine data sets from the UCI Machine Learning Repository [2] are used in this test. The purpose of the experiment is to have our algorithms perform on the real world data, especially on the oncological and the medical data, which are

described briefly below. **Bupa:** This contains blood test results of 345 male patients. The task is to predict whether a patient has a propensity for a liver disorder.
Breast Cancer: This breast cancer data was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. It is one of three oncology data sets frequently used in the machine learning literature.
Wisconsin: The Wisconsin breast cancer data is another oncology data-set repeatedly used to benchmark the performance of learning algorithms.
Lung Cancer: The set contains only 32 instances with 56 attributes, which is quite typical of medical data-set. Such types of data bring great challenges to machine learning algorithms.
Cleveland: The heart disease data set was compiled by Dr. Detrano and was collected at the Cleveland Clinic Foundation. The task is to predict whether the patient has heart disease.

The summary of the data sets can be found in Table 1.

To further test the robustness and accuracy of the learning algorithms and to eliminate bias in the data set, for each data set, a tenfold cross validation was performed. The tenfold cross validation test was repeated ten times, each with a different random partition of the data. As such, for each machine learning algorithm and each of the nine data sets, 10 x 10 = 100 independent tests were done consisting of training and modelling from 90% of the data and testing goodness of fit on the remaining 10%.

Table 1. Summary of Data Sets

Data-set Name	size	Discrete Attributes	Continuous Attributes	Number of Classes
Balance	625	4	0	3
Bupa	345	0	6	2
Breast Cancer	286	9	0	2
Wisconsin	699	9	0	2
Credit	1000	0	24	2
Lung Cancer	32	56	0	3
Cleveland	303	7	6	2
Sonar	208	0	60	2
Wine	178	0	13	3

4.3 Test Results

5 Discussions

We compare the MML oblique tree scheme to C4.5 and C5. The results from Table 4 clearly suggest that the MML oblique trees are much smaller than the

Table 2. "Right"/"Wrong" predictive accuracy

Name	C4.5	C5	MML Oblique Tree
Balance	77.8 ±4.3	77.8 ± 4.5	88.5 ± 4.0
Bupa	65.5 ± 7.4	65.5 ± 7.8	65.1 ± 8.1
Breast Cancer	71.2 ± 8.7	71.1 ± 8.4	72.8 ± 8.0
Wisconsin	94.6 ± 2.5	94.8 ± 2.5	96.0 ± 2.3
Credit	73.2 ± 4.3	73.3 ± 3.8	75.4 ± 4.7
Lung Cancer	40.0 ± 23.3	40.7 ± 24.8	46.8 ± 22.4
Cleveland	77.1 ± 7.6	77.2 ± 7.9	77.2 ± 7.8
Sonar	72.8 ± 9.2	73.9 ± 10.0	76.0 ± 9.2
Wine	93.6 ± 5.7	93.2 ± 5.8	93.2 ± 6.1

Table 3. Related test data code length (RCL)

Name	C4.5	C5	MML Oblique Tree
Balance	0.93±0.12	0.92 ±0.11	0.33 ± 0.08
Bupa	1.07±0.22	1.07 ±0.21	0.96 ± 0.15
Breast Cancer	0.88±0.17	0.88 ±0.17	0.84 ± 0.14
Wisconsin	0.26±0.10	0.25 ±0.12	0.21 ± 0.10
Credit	0.88±0.08	0.88 ±0.08	0.79 ± 0.09
Lung Cancer	1.83±0.50	1.86 ±0.65	0.94 ± 0.30
Cleveland	0.80±0.24	0.81 ±0.21	0.76 ± 0.22
Sonar	1.07±0.37	1.06 ±0.42	0.98 ± 0.33
Wine	0.42±0.30	0.44 ±0.29	0.28 ± 0.18

Table 4. Size of resultant trees - number of leaf nodes

Name	C4.5	C5	MML Oblique Tree
Balance	81.6±9.7	41.7 ±4.6	10.4 ±0.9
Bupa	49.2±9.8	27.3 ±5.4	6.7 ± 2.6
Breast Cancer	24.2±8.3	13.1 ±4.2	3.0 ± 0.6
Wisconsin	23.7±5.3	12.3 ±2.8	5.5 ± 0.9
Credit	151.4±17.7	77.6 ±9.1	6.5 ± 2.4
Lung Cancer	12.2±2.3	6.6± 1.1	2.2 ± 0.4
Cleveland	36.7±7.2	20.0 ±4.2	7.3 ± 1.8
Sonar	28.2±3.1	14.9 ±1.6	11.6 ± 9.3
Wine	9.6±1.3	5.4 ±0.7	3.6 ± 0.5

univariate trees from C4.5 and C5. That can be explained in the following two points. Firstly, the oblique decision trees are able to partition the data more efficiently by using multivariate linear functions at splits. Secondly, our algorithm does not over grow the tree and rely on pruning to control the complexity. The MML oblique trees perform significantly better than C4.5 and C5 on all datasets, as shown in table 2. As reported in [18, 19], MML inference is resistant to over-fitting. Another reason is that univariate trees generated by C4.5 and C5 have more leaf nodes with less data, thus more skewed distributions are often inferred from leaf nodes. MML oblique trees are also have higher "right"/"wrong" accuracy than C4.5 and C5 except on the Bupa and the wine data, which suggest that more works need to be done on the search algorithms. As expected, none of the algorithms have good results from the lung cancer data. Learning from small set of data with a great number of attributes remains a great challenge for machine learning algorithms.

6 Conclusion and Future Research

We have introduced a new oblique decision tree inference scheme by using the MML principle, as indicated by the experimental results. Our algorithm produces very simple trees with excellent performance on both "right"/"wrong" accuracy and probabilistic prediction. By encoding the multivariate splits and extending the MML decision tree coding scheme, our scheme has success in finding the optimal trade-off between the complexity of the model and the goodness of fit. Although the scheme is still in the initial state, the results are promising. However, more research need to be done in the future. Firstly, the search heuristic may need to be improved as an efficient search algorithm is crucial for any multivariate tree scheme. Secondly, as pointed out in section 3.3, the performance of the system may be enhanced by using multiple tree averaging. Further down the track, to use MML coding for internal nodes with SVMs or nonlinear splits is also an interesting research topic.

References

1. K. P. Bennett and J. A. Blue. A support vector machine approach to decision trees. In *IJCNN*, pages 2396–2401, Anchorage, Alaska, 1998.
2. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
3. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification And Regression Trees*. Wadsworth & Brooks, 1984.
4. Erick Cantu-Paz and Chandrika Kamath. Using evolutionary algorithms to induce oblique decision trees. In *Proc.Genetic and Evolutionary Computation Conference*, pages 1053–1060, Las Vegas, Nevada, USA, 2000. Morgan Kaufmann.
5. D.L. Dowe, G.E. Farr, A.J. Hurst, and K.L. Lentin. Information-theoretic football tipping. In N. de Mestre, editor, *Third Australian Conference on Mathematics and Computers in Sport*, pages 233–241. Bond University, Qld, Australia, 1996. http://www.csse.monash.edu.au/∼footy.

6. D.L. Dowe and N. Krusel. A decision tree model of bushfire activity. In *(Technical report 93/190) Dept. Comp. Sci., Monash Uni., Clayton, Australia*, 1993.
7. David G. Heath, Simon Kasif, and Steven Salzberg. Induction of oblique decision trees. In *International Joint Conference on AI (IJCAI)*, pages 1002–1007, 1993.
8. Manish Mehta, Jorma Rissanen, and Rakesh Agrawal. MDL-based Decision Tree Pruning. In *The First International Conference on Knowledge Discovery & Data Mining*, pages 216–221. AAAI Press, 1995.
9. Sreerama K. Murthy. *On Growing Better Decision Trees from Data*. PhD thesis, The John Hopkins University, 1997.
10. S.L. Needham and D.L. Dowe. Message length as an effective Ockham's razor in decision tree induction. In *Proc. 8th International Workshop on Artificial Intelligence and Statistics*, pages 253–260, Key West, Florida, U.S.A., Jan. 2001.
11. J.J. Oliver and C.S. Wallace. Inferring Decision Graphs. In *Workshop 8 International Joint Conference on AI (IJCAI)*, Sydney, Australia, August 1991.
12. C. Perlich, F. Provost, and J.S. Simonoff. Tree induction versus logistic regression: a learning-curve analysis. *Journal of Machine Learning Research*, 4:211–255, 2003.
13. Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52:199–215, Sept. 2003.
14. J.R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann,San Mateo,CA, 1992. The latest version of C5 is available from http://www.rulequest.com.
15. J.R. Quinlan and R. Rivest. Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation*, 80:227–248, 1989.
16. J.J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
17. R. Schack, G. M. D. Ariano, and C. M. Caves. Hypersensitivity to perturbation in the quantum kicked top. *Physical Review E.*, 50:972–987, 1994.
18. P.J. Tan and D.L. Dowe. MML inference of decision graphs with multi-way joins. In *Proc. 15th Australian Joint Conf. on AI, LNAI 2557 (Springer)*, pages 131–142, Canberra, Australia, 2-6 Dec. 2002.
19. P.J. Tan and D.L. Dowe. MML inference of decision graphs with multi-way joins and dynamic attributes. In *Proc. 16th Australian Joint Conf. on AI, LNAI 2903 (Springer)*, pages 269–281, Perth, Australia, Dec. 2003. http://www.csse.monash.edu.au/∼dld/Publications/2003/Tan+Dowe2003.ref .
20. P.J. Tan and D.L. Dowe. MML inference of oblique decision trees. In *Proc. 17th Australian Joint Conf. on AI, LNAI 3339 (Springer)*, pages 1082–1088, Cairns, Australia, Dec. 2004.
21. Vladimir N. Vapnik. *Statistical Learning Theory*, chapter 6.2, pages 224–229. JOHNSON WILEY & SONS, INC, 1998.
22. C.S. Wallace and D.M. Boulton. An Information Measure for Classification. *Computer Journal*, 11:185–194, 1968.
23. C.S. Wallace and D.L. Dowe. Minimum Message Length and Kolmogorov Complexity. *Computer Journal*, 42(4):270–283, 1999.
24. C.S. Wallace and P.R. Freeman. Estimation and Inference by Compact Coding. *Journal of the Royal Statistical Society. Series B*, 49(3):240–265, 1987.
25. C.S Wallace and J.D. Patrick. Coding Decision Trees. *Machine Learning*, 11:7–22, 1993.