BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages i93–i102
DOI: 10.1093/bioinformatics/btg1066

Discovery of significant rules for classifying cancer diagnosis data

Jinyan Li*, Huiqing Liu, See-Kiong Ng and Limsoon Wong

Institute for Infocomm Research, 21, Heng Mui Keng Terrace, 119613, Singapore

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Methods and Results: We introduce a new method to discover many diversified and significant rules from high dimensional profiling data. We also propose to aggregate the discriminating power of these rules for reliable predictions. The discovered rules are found to contain low-ranked features; these features are found to be sometimes necessary for classifiers to achieve perfect accuracy. The use of low-ranked but essential features in our method is in constrast to the prevailing use of an ad-hoc number of only top-ranked features. On a wide range of data sets, our method displayed highly competitive accuracy compared to the best performance of other kinds of classification models. In addition to accuracy, our method also provides comprehensible rules to help elucidate the translation between raw data and useful knowledge.

Supplementary information: http://sdmc.lit.org.sg/GEDatasets/supplementaldata/eccb2003/ECCB2003.html.

Contact: jinyan@i2r.a-star.edu.sg

1 INTRODUCTION

Microarray gene expression profiling is a technology that has been widely used in post-genome cancer research studies Golub *et al.* (1999); Yeoh *et al.* (2002); Singh *et al.* (2002); Gordon *et al.* (2002), while mass spectrometry is also increasingly being used in the cancer research field for measuring the mass-charge ratios of molecular proteins in tumor tissues Petricoin *et al.* (2002). Both microarray and mass spectrometry generate high-dimensional data from large-scale measurements of genes and proteins. The complexity of the resulting data then naturally requires computational analysis tools to extract significant and reliable *rules* from the data. The discovery of such rules can then ease the difficulty for translating the complex raw data into relevant and clinically useful diagnostic or prognostic knowledge.

We define a *rule* as a set of conjunctive conditions with a predictive term. The general form of our rules is represented as follows:

> **If** $cond_1$ **and** $cond_2$ **and** $\cdots$ $cond_m$,
> **then** *a predictive term*

The predictive term in a rule often refers to a single class (e.g. a particular subtype of a cancer). All conditions in a rule are required to be true in some samples of the predictive class, but not all true in any samples of any classes other than the one in the predictive term.

In cancer and other disease diagnosis, the number m of conditions is preferred to be no more than 5 for easy understanding. Ideally, rules with $m = 1$, 2, or 3 are best for clinical diagnosis. The following rule Li *et al.* (2003) is an example containing two conditions on the gene expression profiles of childhood leukemia cells:

> **If** *the expression of 40454_at is* $\geq$ 8280.25
> **and** *the expression of 41425_at is* $\geq$ 6821.75,
> **then** *this sample is subtype E2A-PBX1.*

This rule is not satisfied by any cells of any leukemia subtypes other than *E2A-PBX1*, while 100% of the samples in the *E2A-PBX1* class each satisfy both of the two conditions on gene expression profiling. It is therefore useful for clinical diagnosis purposes.

This paper aims to study two problems:

1. How to discover many significant rules from high-dimensional data, and
2. How to aggregate the discriminating power of the many rules to make reliable predictions.

The first problem addresses issues in understanding the mechanism of a disease or identification of new pathways: the discovery of valid and understandable rules from expression data will provide important insights on the underlying biological processes. The second problem, seeking to improve the discriminating power of the rules, will address the need for accuracy in clinical diagnosis and prognosis, or subtype classification of diseases.

The rest of the paper is organized as follows: Section 2 provides more background information and describes our

*To whom correspondence should be addressed.

contribution made in this paper. Section 3 explains with examples the intuition that ‘the second could be best’, an observation that has inspired our new approach. Section 4 describes formally our method for the discovery of rules based on the concept of tree committees. Section 5 describes state-of-the-art committee classifiers such as bagging and boosting which are the most relevant work to ours, and explains the differences between them and our approach. Section 6 describes two gene expression data sets and one proteomic data set, all having large numbers of samples ranging from nearly 200 to 327. We also report rigorous results for accurate classification. Section 7 concludes this paper with a discussion of other possible ways for the discovery of trees and rules.

2 BACKGROUND AND CONTRIBUTION

Traditionally, classification rules are discovered from training data (the known samples) by decision-tree based methods such as CART and C4.5 Breiman *et al.* (1984); Quinlan (1993). However, such rules have a limitation: They are mutually exclusive, covering the entire of training samples exactly only once. We call this the single coverage constraint. Due to this constraint, decision-tree methods are not encouraged to derive *many* significant rules; such possible omission of significant rules in the resulting system may bias unjustified predictions. Decision-tree methods have also the so-called fragmentation problem Pagallo and Haussler (1990); Friedman *et al.* (1996): as less and less training data are used to search for root nodes of subtrees, a series of many locally important but globally un-important rules are generated. These minor rules may in turn mis-guide the resulting system, decreasing the accuracy of the decision trees.

In our approach, we still use decision trees to discover rules. But we use committees of trees instead of single trees. As a tree is a collection of rules where every leaf of the tree corresponds to a rule, multiple trees can contain many significant rules. The use of multiple trees breaks the single coverage constraint, and allows the same training samples to be explained by many either significant or minor rules. This is a good idea because the mutually exclusive rules in one decision tree cut off many interactions among features. However, multiple trees contain significant rules that can capture many interactions from different aspects. The multiple cross-supportive rules therefore much strengthen the power of prediction.

As another contribution, our method solves the fragmentation problem by weighting rules with their coverage to prevent the minor rules from playing equal role as the more significant rules in making decisions.

Our approach differs fundamentally from the state-of-the-art committee methods such as bagging Breiman (1996) and boosting Freund and Schapire (1996). Unlike them, our method always uses the original training data instead of *bootstrapped*, or pseudo, training data to construct a sequence of different decision trees. Our rules reflect precisely the nature of the original training data, while the rules produced by the bagging or boosting methods may not be correct when applied to the original data, as they sometimes only approximate the true rules. The bagging or boosting rules should therefore be employed very cautiously, especially in the applications of bio-medicine where such concerns could be critical.

Specifically, we discover a committee of multiple trees using a cascading approach. First, we rank all features into a list according to their gain ratio Quinlan (1993). Then we build the first tree using the first top-ranked feature as the root node, the second tree using the second top-ranked feature as root node, and so on. In general, we build the *k*th tree using the *k*th top-ranked feature as root node.

Given a test sample for classification, our method makes the final decision by voting, in a weighted manner, the rules in the *k* trees of the committee that the test sample satisfies. We assign weights to the rules according to their *coverage* in the original training data; that is, each rule is weighted by the maximal percentage of training samples in a class that satisfy this rule. This weighting method distinguishes between significant and minor rules, so that those rules all contribute in accordance to their proportional roles to the final voting.

Note that all original features are open for our selection to form rules, so our method avoids the difficult classical problem of how many top-ranked features to be used for a classification model. We found that our significant rules often contain low-ranked features, and that these features are sometimes necessary for classifiers to achieve perfect accuracy. If ad-hoc numbers of only top-ranked features are used as traditionally, we would miss many significant rules and sometime lose perfect accuracy. We will show an example to explain this later in Section 6.1. We also note that the number of features used in our rules is as small as 3 or 5, so they are easily understandable.

On a wide range of data sets, our method also displays highly competitive accuracy compared to the best performance of other kinds of classification models. So, our method can provide both highly accurate and comprehensible rules to elucidate the translation between the raw data and useful knowledge for the scientific understanding and clinical diagnosis of many common diseases such as cancer.

3 MOTIVATING EXAMPLES

In this section, we present real examples discovered from high-dimensional profiling data to explain the following three facts:

- Significant rules often contain globally low-ranked features;
- If the construction of a tree is confined to a set of globally top-ranked features, the rules in the resulting tree may be less significant than those rules derived by using the whole feature space; and
- Alternative trees can often outperform or compete with the performance of the 'optimal' tree when the same set of test data are applied.

These facts and observations suggest us: (1) not only rely on top-ranked features; (2) not only use one single tree, namely only one set of mutually exclusive rules. These facts let us realize that not only top-ranked features are important in building significant rules, and that decision trees rooted by second-best features are also reliable and useful. We sometimes use the term 'second could be best' to outline and refer to these facts and observations.

3.1 Significant rules often contain globally low-ranked features

A rule has a *coverage*, namely the percentage of the samples in a class satisfying the rule. Suppose a class consists of 100 positive samples and a rule is satisfied by 75 of them, then this rule's coverage is 75%. A *significant* rule is one with a large coverage—e.g. of at least 50%. Otherwise, we define it as a *minor* rule.

Given a data set pair having two classes—positive and negative—of samples, a feature's discriminating power to differentiate the two classes can be roughly measured by its gain ratio Quinlan (1993), or by entropy Fayyad and Irani (1992). The entropy method measures the class distribution under a feature of the whole collection of samples. If the distribution—e.g. expression levels of a gene for x tumor and normal samples—shows clear boundary between the tumor and normal classes, this feature is then assigned a small entropy value. A small entropy value indicates a low or zero uncertainty for differentiating the two classes by this single feature, and such features are thus ranked at top positions.

To demonstrate the first observation that significant rules often contain low-ranked features, our example is a significant rule discovered from a prostate disease data set that comprises expression profiles from 52 tumor cells and 50 normal cells Singh *et al.* (2002):

> **If** *32598_at* $\leq$ 29 **and** *33886_at* $\leq$ 10 **and** *34950_at* $\leq$ 5, **then** *this is a tumor cell.*

This rule is a significant rule with a coverage of 94% (49/52) in the tumor class. Let us look at the three features' ranking positions. While gene *32598_at* sits at the first position, the other two of the three genes in this significant rule are globally lower-ranked: gene *33886_at* sits at the 210th position and *34950_at* sits at the 266th position in the entire set of 12 600 genes.

Table 1. Various ranking positions of the three features used in a significant rule discovered from a prostate disease gene expression profiling data. Here S-to-N stands for the signal-to-noise measurement Golub *et al.* (1999)

Features	*32598_at*	*34950_at*	*33886_at*
	ranking positions		
S-to-N	6	8109	9775
t-statistics	13	8302	9790
entropy	1	869	47
gain ratio	1	210	266
$\mathcal{X}^2$	2	42	1095

The rank order here is based on the gain ratio method. To verify that this is not an artifact of the ranking method used, we have also explored alternative ranking in terms of other metrics such as signal-to-noise measurement Golub *et al.* (1999), t-statistics, entropy Fayyad and Irani (1992), and $\mathcal{X}^2$ measurement Liu and Motoda (1998). Table 1 shows the ranking positions of the three genes using various ranking methods. We found that in general, the ranking of the genes agrees even when different methods are used. Therefore, this example illustrates that even very low-ranked genes can be included in significant rules.

As a second example, we present another significant rule, discovered from the same prostate cancer data set above, which is dominant in the normal class:

> **If** *32598_at*$>$ 29 **and** *40707_at*$>$ -6, **then** *this is a normal cell.*

This rule is significant with an 82% (41/50) coverage in the normal class. The ranking positions of the two genes are as follows: gene *32598_at* sits at the first position, while its component gene *40707_at* is globally lower-ranked at a position below 1000th.

Differentially expressed genes in a microarray experiment can be up-stream causal genes or can be merely down-stream surrogates. We note that a surrogate gene's expression should be strongly correlated to a causal gene's and hence they should have similar discrimination power and should have similar ranking. Thus, if a significant rule contains both high-ranked and low-ranked genes, we can suspect that these genes have independent paths of activation and thus there are at least two genes that are causal.

We have observed this interesting phenomenon in many other data sets such as a childhood leukemia data set Yeoh *et al.* (2002), a lung cancer data set Gordon *et al.* (2002), an ovarian disease data set Petricoin *et al.* (2002). In total, we examined over 50 significant rules which contain around 120 features. 56% of these features sit at position range between 10 and 500; 26% features have

Table 2. Five rules in a C4.5 tree derived from a prostate disease gene expression profiling data

Rules	1	2	3	4	5
Coverage	94%	6%	12%	6%	82%
Class	Tmr	Nml	Nml	Tmr	Nml
# of features	3	3	2	2	2

Table 3. Four rules in a C4.5 tree built on only three top-ranked features

Rules	1	2	3	4
Coverage	88%	14%	10%	88%
Class	Tmr	Nml	Tmr	Nml
# of features	2	3	3	2

positioned below 500. Overall, we can say that the low-ranked features are important components in building significant rules. The use of only top-ranked features for data analysis, which is a common pre-filtering strategy deployed by many methods in this field, is not impartial indeed. In the next subsection, we further illustrate this point from a different angle: there can indeed be important changes in the derived rules' significance level if a decision tree is constructed using only top-ranked features.

3.2 Down change of rules' significance if discovery is based on a small number of top-ranked features

Here, we use C4.5 Quinlan (1993) to build up two trees, namely two groups of rules, and then compare the rules to see if there are any changes. First, we construct a tree based on the original whole feature space. The selection of tree nodes is freely open to any features, including globally low-ranked features. Figure 1a shows the tree discovered from the prostate disease data set Singh *et al.* (2002). Each path of the tree, from root to a leaf, represents a single rule. So, this tree has five rules, obtained by the depth-first traversal to the five leaves. We name the rules 1, 2, 3, 4 and 5 from the left side to right. Their respective coverage and number of features contained are listed in Table 2. Rule 1 is the most significant rule: it has a 94% coverage over the tumor class. Recall that this rules contains two extremely lower-ranked features as mentioned earlier.

Next, we limit our second tree to be constructed with only 3 globally top-ranked features, namely *32598_at*, *38406_at*, and *37639_at*. The number 3 is chosen to be equal to the number of features in the most significant rule (Rule 1) in the first tree. Figure 1b shows the structure of the second tree; the rules' respective coverage and the number of features they contained are reported in Table 3.

An important observation is the unexpected decrease of the significance of top rule in the second tree constructed with only pre-filtered top-ranked features. This observation supports our belief that the best could be the second: best top-ranked feature groups do not necessarily produce the most important rules.

In fact, we can prove that if the lowest feature position in the most significant rule is p, then at least p number of top-ranked features are necessary for deriving a decision tree which can contain a rule with the same significance. It is hard to know the number p if the whole feature space is not considered. So, to pre-set a threthold to select top-ranked features is a heuristic that has a risk of losing useful low-ranked features.

3.3 Alternative trees can perform equally well in prediction

The aim of this subsection is to see if it is possible to generate, from the same training data set, two trees (or two groups of rules) that are diversified but perform equally well in prediction.

Given a data set, we use C4.5 to generate the 'optimal' tree using the most discriminatory feature as the root node. Next, to generate an alternative tree, we use an approach that is slightly different from C4.5: we force the second-best feature to become the root node for this tree. The remaining nodes are then built by the standard C4.5 method. We found that such pairs of trees often have almost the same prediction power, and sometimes, the second tree even outperforms the first one.

For illustration, we report an example of a pair of trees where the so-called second-best tree actually greatly outperformed the first. Figure 2 shows the 'optimal' C4.5 tree constructed on a layered data set to differentiate the subtype Hyperdip$>$50 against other subtypes of childhood leukemia Yeoh *et al.* (2002). Although this C4.5 tree made no mistakes on the training data, it made 13 errors out of 49 test samples. In this case, our second-best tree managed to independently improve the dismal accuracy of the first tree by making only 9 mistakes on the testing set. Interestingly, when the pair of trees are combined by our method (shown in next section), the resulting hybrid made even fewer mistakes of only 6.

On closer inspection of this pair of trees, we found that the set of features used in the first tree is disjoint from the set used in the second tree. The former has the following four features at its tree nodes: *36620_at*, *39806_at*, *32845_at* and *34365_at*; but the latter has a different set of features at its four tree nodes: *38518_at*, *32139_at*, *35214_at* and *40307_at*. Therefore, the two trees are really diversified. The two trees each contain two significant rules each for one of the two classes. Again,

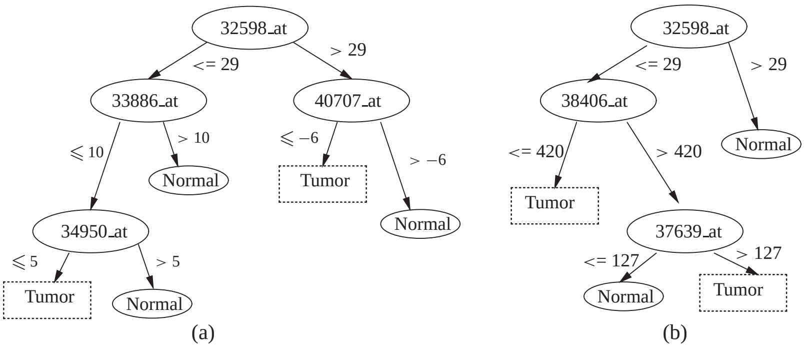

Fig. 1. Two trees induced from the prostate disease data set of gene expression profiles of 102 cells: (a) the standard C4.5 tree constructed by using whole feature set; (b) a tree constructed by using only three top-ranked features.

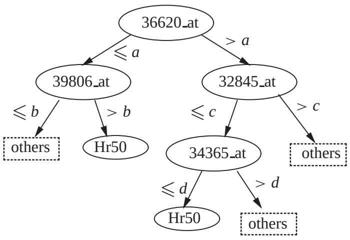

Fig. 2. A decision tree induced by C4.5 from a layered data set to differentiate the subtype Hyperdip>50 against other subtypes of childhood leukemia Yeoh *et al.* (2002). Here Hr50 = Hyperdip>50, $a = 16115.4$, $b = 4477.9$, $c = 3453.4$, $d = 2400.9$.

these significant rules contain very low-ranked features such as *34365_at* that sits at the 1878th position. Another particularly interesting point here is that the coverage of the top rules in the second tree has increased as compared to the rules in the first tree. This could explain why the second tree outperformed the first.

Yet another example can be found in trees constructed from the layered data set Yeoh *et al.* (2002) to differentiate the subtype MLL against other subtypes of childhood leukemia. Here, the first standard C4.5 tree made 1 mistake out of 55 test samples, while our second tree made 2 mistakes. However, by combining the two trees, the hybrid made no mistakes with the test set. Randomly, we examined ten such pairs of trees and found 4 pairs where the first tree won, 3 pairs where the second tree won, and 3 pairs where the two trees got a tie in performance.

As our tree pairs have generally similar prediction power, we can treat them as ‘experts’ who understood the inherent inter-relationship of the features in the data with their own diversified experience. This suggests a committee of trees approach: we can increase the diversity of the trees’ ‘expertise’ by generating a third tree, a fourth tree, and so on. The wide range of diversities provided by such a committee of trees or rules, together with the high quality of the individual trees in the committee, will provide a good basis for scientists to study bio-medical data and to conduct cancer diagnosis reliably.

4 METHODS

This section describes our new methods to discover significant rules using the concept of tree committees, and presents methods to use the rules as an ensemble for reliable predictions. We first discuss the two-class case, then generalize our methods to handle multi-class applications. Important variants of these methods will be presented later in Section 7 .

4.1 Rule discovery

Given a training data set $\mathcal{D}$ having two classes of samples, *positive* and *negative*, we use the following steps to iteratively derive k trees from $\mathcal{D}$, where k is significantly less than the number of features used in $\mathcal{D}$, and usually we set k as 20:

Step 1: Use gain ratios to rank all the features into an ordered list with the best feature at the first position.

Step 2: $i = 1$.

Step 3: Use the ith feature as root node to construct the ith tree.

Step 4: Increase i by 1 and goto Step 3, until $i = k$.

Then rules can be directly generated from these trees by the depth-first traversals. To identify significant rules, we just rank all the rules according to each rule’s coverage, the top-ranked ones are significant. The significant rules may then be used for understanding possible interactions between the features (e.g. genes or proteins) involved in these rules. To use the rules for class prediction, our method is described in the next subsection.

4.2 Class prediction

Given a test sample T, each of the k trees in the committee will have a specific rule to tell us a predicted class label for this test sample.

Denote the k rules from the tree committee as:

$$rule_1^{pos}, rule_2^{pos}, \cdots, rule_{k_1}^{pos},$$

$$rule_1^{neg}, rule_2^{neg}, \cdots, rule_{k_2}^{neg}.$$

Here $k_1 + k_2 = k$. Each of $rule_i^{pos}$ $(1 \le i \le k_1)$ predicts T to be in the positive class, while each of $rule_i^{neg}$ $(1 \le i \le k_2)$ predicts T to be in the negative class. Sometimes, the k predictions can be unanimous—i.e. either $k_1 = 0$ or $k_2 = 0$. In these situations, the predictions from all the k rules agree with one another, and the final decision is obvious and seemed reliable. Oftentimes, the k decisions are mixed with either a majority of positive classes or a majority of negative classes. In these situations, we use the following formulas to calculate two classification scores based on the coverages of these rules:

$$Score^{pos}(T) = \sum_{i=1}^{k_1} coverage(rule_i^{pos}),$$

$$Score^{neg}(T) = \sum_{i=1}^{k_2} coverage(rule_i^{neg}).$$

If $Score^{pos}(T)$ is larger than $Score^{neg}(T)$, we assign the positive class to the test sample T. Otherwise, T is predicted as negative.

By using the rules' coverage as weights, we avoid the pitfalls of simple equal voting adopted by bagging Breiman (1996). Our weighting policy allows the tree committee to automatically distinguish the contributions from the minor rules and from the significant rules in the prediction process.

For multi-class problems, the classification score for a specific class, say class $\mathcal{C}$, is calculated as:

$$Score^{\mathcal{C}}(T) = \sum_{i=1}^{k_{\mathcal{C}}} coverage(rule_i^{\mathcal{C}}).$$

The class that receives the highest score is then predicted as the test sample's class.

5 COMPARISON WITH THE STATE-OF-THE-ART CLASSIFIERS

In this section, we review C4.5 Quinlan (1993) for discovering a set of mutually exclusive rules, namely, a decision tree, and then highlight the difference of our committee trees from standard C4.5 trees. Then, we review two state-of-the-art committee classifiers—bagging Breiman (1996) and boosting Freund and Schapire (1996)—to compare their working platforms with ours. We will present our superior performance over the traditional committee classifiers in the next section.

C4.5 Quinlan (1993) is a heuristic algorithm for inducing decision trees. C4.5 uses an entropy-based selection measure to determine which feature is most discriminatory. This measure is also called *gain ratio*, or *maximum information gain*. Most decision trees in the literature are constructed by C4.5. In our committees of trees, however, most are not standard C4.5 trees. This is because instead of using only the most discriminatory feature, we have employed a wide range of feature choices for building the root nodes of the trees. Our trees also have strong prediction power, and sometimes even better performance than the standard C4.5 trees, as discussed in Section 3.

Committee decision techniques such as AdaBoost Freund and Schapire (1996) and Bagging Breiman (1996) have also been proposed to reduce the errors of single trees by voting the member decisions of the committee Bauer and Kohavi (1999); Quinlan (1996). Unlike our approach, AdaBoost and Bagging both apply a base classifier (e.g. C4.5) multiple times to generate a committee of classifiers using bootstrapped training data. Assume that a given set of training data has N samples, and a number R of repetitions or *trials* of the base classifier is to be applied. By the bagging idea, for each trial $t = 1, 2, \cdots, R$, a bootstrapped training set is generated from the original data. Although this new training set is the same size as the original data, some samples may no longer appear in the new set while others may appear more than once. Denote the R bootstrapped training sets as $B_1, B_2, \cdots, B_R$. For each B_t, a classifier C_t is built. A final, bagged classifier C^* is constructed by aggregating $C_1, C_2, \cdots,$ and C_R. The output of C^* is the class predicted most often by its sub-classifiers, with ties broken arbitrarily.

Similar to bagging, boosting also uses a committee of classifiers for classification by voting. Here, the construction of the committee of classifiers is different: while bagging builds the individual classifiers separately, boosting builds them sequentially such that each new classifier is influenced by the performance of those built previously. In this way, those samples incorrectly classified by previous models can be emphasized in the new model, with an aim to mold the new model become an expert for classifying those hard instances. A further difference between the two committee techniques is that boosting weights the individual classifiers' output depending on their performance, while bagging gives equal weights to all the committee members. AdaBoost.M1 Freund and Schapire (1996) is a very good example of the boosting idea.

Our method differs from these traditional committee classifiers in the management of the original training data. Bagging and boosting generate bootstrapped training data for every iteration's construction of trees. Our method keeps both the size of the original data *and* the features' values unchanged throughout the whole process. As a result, our rules will always reflect precisely the nature of the original data, whereas because of the use of bootstrapped training data, some bagging or boosting rules may not be true when applied to the original training data.

In addition to being different from bagging and boosting, our method also differs from another voting method called the randomized decision trees Dietterich (2000). This algorithm is a modified version of the C4.5 learning algorithm in which the decision about which split to introduce at each internal node of the tree is randomized. With a different random choice, a new tree is then constructed. Twenty of the best splits (in terms of gain ratio) for a feature were considered to be the pool of random choices Dietterich (2000). Every member of a committee of randomized trees constructed by this method always shares the same root node feature. The only difference between the members is at their internal nodes. In contrast, our trees in a committee differs from one another not only at root node but also at internal features. Our committees of trees have much larger potential for diversity than the randomized trees.

6 DATA AND EVALUATION

We report here the performance of our method and compare it with bagging and boosting methods, as well as support vector machines (SVM) Burges (1998) and k-nearest neighbours on a wide array of expression data, including a childhood leukemia gene expression data Yeoh *et al.* (2002), an ovarian tumor proteomic data Petricoin *et al.* (2002), a lung cancer gene expression data Gordon *et al.* (2002), as well as other data Armstrong *et al.* (2002); Golub *et al.* (1999). All these data have been grouped in our supplementary website http://sdmc.lit.org.sg/GEdatasets.

We report our results based on two measures: test error numbers—the number of misclassifications on independent *test* samples, and the error numbers of 10-fold cross validation. When the error numbers are represented in the format $x : y$, it means that x number of samples from the first class and y number of samples from the second class are misclassified. The number of iterations used in bagging and boosting was set as 20—equal to the number of trees used in our method. The main software package used in the experiments is *Weka* version 3.2, its Java-written open source are available at http://www.cs.waikato.ac.nz/~ml/weka/ under the GNU General Public Licence.

6.1 Classification of ovarian tumor and normal patients by proteomics

Our first evaluation is on a recent ovarian data set Petricoin *et al.* (2002) which is about how to distinguish ovarian cancer from non-cancer using serum proteomic patterns (instead of DNA expression). This proteomic spectra data generated by mass spectroscopy can be found at http://clinicalproteomics.steem.com; there are several similar data sets in this site. For our evaluation study, we have chosen the biggest one, which is dated on 6-19-02. The data have a total of 253 samples: 91 controls (non-cancer) and 162 ovarian cancers. Each data sample is described by 15 154 features, namely, the relative amplitudes of the intensities at 15 154 molecular mass/charge (M/Z) identities.

Table 4. The error numbers (Cancer : Normal) of 10-fold cross validation by four classification models over 253 proteomic ovarian data samples

Methods	Ours	C4.5		
		Single	Bagging	Boosting
Errors	0 (0:0)	10 (4:6)	7 (3:4)	10 (4:6)

For each feature, we normalize all its values (intensities) of the 253 samples using the following formula: $NV = (V - Min)/(Max - Min)$, where NV is the normalized value, V the raw value, Min the minimum intensity and Max the maximum intensity of the given feature. The normalized data can be found at our supplementary website: http://sdmc.lit.org.sg/GEdatasets.

The original data set does not include a separate test data set. As such, we evaluate our method using 10-fold cross validation over the whole data set. The performance is summarized in Table 4. We can see that our method is remarkably better than all the C4.5 family algorithms, reducing their 10 or 7 mistakes to a error-free performance in the total 253 test samples, giving rise to truly excellent diagnosis accuracy for ovarian cancer based on serum proteomic data.

For further comparison, we also used SVM and 3-nearest neighbour to conduct the same 10-fold cross validation. SVM also achieved 100% accuracy. However, SVM used all the 15 154 input features together with 40 support vectors and 8308 kernel evaluations in its decisions. It is difficult to derive understandable explanations of any diagnostic decision made by this system. In contrast, our method used only 20 trees and less than 100 rules. The other non-linear classifier, 3-nearest neighbour, have made 15 mistakes (5 in Cancer and 10 in normal); it is therefore not of comparable performance to ours.

What are the results if ad-hoc numbers of only top-ranked features are used in the classification models? If only top 10, 20, 25, 30, 35, or 40 entropy-ranked features are used, support vector machines could not achieve the perfect 100% accuracy; our method could not achieve the perfect 100% accuracy either. All other classifiers such as k-nearest neighbour, C4.5 family algorithms, or naive bayes could not either. So, if the cut threshold were set as one of these ad-hoc numbers, the classification algorithms would miss the perfect accuracy on this data set, as our algorithm and support vector machines can reach the 100% accuracy when the whole feature space are considered. In fact, we used some low-ranked features whose rankings were below the 3000th position. Such a comparison results indicate that some low-ranked features are necessary for classifiers to get perfect performance. Openning all features for consideration (though most of them may be not in the final rules) as used in our method is an idea that is more flexible than the idea of using only top-ranked features.

6.2 Subtype classification of childhood leukemia by gene expression

Acute Lymphoblastic Leukemia (ALL) in children is a heterogeneous disease. The current technology to identify correct subtypes of leukemia is an imprecise and expensive process, requiring the combined expertise from many specialists who are not commonly available in a single medical center Yeoh *et al.* (2002). Using microarray gene expression technology and supervised classification algorithms, this problem can be solved such that the cost of diagnosis is reduced and at the same time the accuracy of both diagnosis and prognosis is increased.

Subtype classification of childhood leukemia has been comprehensively studied previously Yeoh *et al.* (2002); Li *et al.* (2003). The whole data consists of gene expression profiles of 327 ALL samples. These profiles were obtained by hybridization on the Affymetrix U95A GeneChip containing probes for 12558 genes. The data contain all the known acute lymphoblastic leukemia subtypes, including T-cell (T-ALL), E2A-PBX1, TEL-AML1, BCR-ABL, MLL, and hyperdiploid (Hyperdip>50). The data were divided by doctors Yeoh *et al.* (2002) into a training set of 215 instances and an independent test set of 112 samples. There are 28, 18, 52, 9, 14, and 42 training instances and 15, 9, 27, 6, 6, and 22 test samples respectively for T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL, and Hyperdip>50. There are also 52 training and 27 test samples of other miscellaneous subtypes.

The original training and test data were layered in a tree-structure Yeoh *et al.* (2002). We present the test error numbers of four classification models, using the 6-level tree-structured data in Yeoh *et al.* (2002), in Table 5. Our test accuracy was shown to be much better than C4.5 and Boosting, and it was also superior to Bagging. SVM made 23 mistakes on the same set of 112 test samples, while 3-nearest neighbour committed 22 mistakes. Their accuracy is therefore only around 80% ($1-\frac{23}{112}$), which is far below our accuracy of 94%. Additionally, the SVM model is very complex, consisting of hundreds of kernel vectors and tens of thousands of kernel evaluations. In contrast, our rules contained only 3 or 4 features, most of them with very high coverage; the rules are therefore highly understandable.

Table 5. Test error numbers of four models on the 112 independent test samples in the problem of 6-subtype classification of the ALL disease Yeoh *et al.* (2002)

Methods	Ours	C4.5		
		Single	Bagging	Boosting
Errors	7	23	10	22

We also report our results with 10-fold cross validations to see how well we distinguish each subtype from all other subtypes in the whole data set. The results are listed in Table 6. Again, our method outperformed the C4.5 algorithm family and 3-nearest neighbour (3-NN), and had a comparable performance with SVM.

6.3 Classification of lung cancer by gene expression

Gene expression method can also be used to classify lung cancer to potentially replace current cumbersome conventional methods to detect, for instance, the pathological distinction between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. In fact, a recent study Gordon *et al.* (2002) has used a ratio-based diagnosis to accurately differentiate between MPM and lung cancer in 181 tissue samples (31 MPM and 150 ADCA), suggesting that gene expression results can be useful in clinical diagnosis of lung cancer.

Note that in this case, the training set is fairly small, containing 32 samples (16 MPM and 16 ADCA), while the test set is relatively large, having 149 samples (15 MPM and 134 ADCA). Each sample is described by 12 533 features (genes). We show our results in comparison to those by the C4.5 family algorithms in Table 7. Once again, our results are better than C4.5 (single, bagging, and boosting). Our results are also comparable to SVM and 3-nearest neighbour in this data set. SVM made only one mistake over the 149 test samples, and 3-nearest neighbour made 3 (2:1) mistakes. However, as before, the complexity of the SVM and the distance model is much more complicated than our trees, again limiting their practical use in scientific discovery and clinical diagnostics. The translation of the complex data into useful clinical knowledge using our method is much more straightforward.

6.4 Results on other data sets

The data sets that we have studied so far are all more than one hundred samples. In this subsection, we report our results using two relatively smaller data sets Armstrong

Table 6. 10-fold cross validation results in the problem of subtype classification of the ALL disease

data sets (whole data size in each class)	CV-10 Error Numbers for the Whole Data					
	Ours ($k = 20$)	C4.5 Single	 Bagging	 Boosting	SVM	3-NN
BCR-ABL vs others (15:312)	9(9:0)	21(12:9)	13(13:0)	22(13:9)	12(12:0)	15(14:1)
E2A-PBX1 vs others (27:300)	1(1:0)	1(1:0)	1(1:0)	1(1:0)	1(1:0)	1(1:0)
HyperL50 vs others (64:263)	14(10:4)	28(18:10)	19(16:3)	23(14:9)	11(8:3)	21(16:5)
MLL vs others (20:307)	8(7:1)	13(7:6)	10(9:1)	13(7:6)	7(7:0)	9(9:0)
T-ALL vs others (43:284)	2(2:0)	1(1:0)	1(1:0)	1(1:0)	1(1:0)	8(8:0)
TEL-AML1 vs others (79:248)	6(2:4)	16(7:9)	12(5:7)	9(4:5)	4(1:3)	14(5:9)

Table 7. The test error numbers (MPM:ADCA) by four classification models over independent 149 MPM and ADCA tissue samples

Methods	Ours	C4.5		
		Single	Bagging	Boosting
Errors	3 (1:2)	27 (4:23)	4 (0:4)	27 (4:23)

Table 8. The test error numbers by four classification models on two small data sets

data sets	test error numbers			
	Ours	C4.5	Bagging	Boosting
Armstrong	0	4 (2:2:0)	2 (1:1:0)	0
Golub	4(0:4)	3(2:1)	3(0:3)	3(2:1)
Golub(10-f)	1(0:1)	18(9:9)	5(0:5)	13(6:7)

et al. (2002); Golub *et al.* (1999) to see how our method fare with small data sets.

The first small data set from Armstrong *et al.* (2002) is about the distinction between MLL and other conventional ALL subtypes. There are a total of only 57 3-class training samples (20, 17, and 20 respectively for ALL, MLL, and AML) and 15 test samples (4, 3, and 8 respectively for ALL, MLL, and AML). Table 8 (the second row) reports the respective classification performance. Once again, single C4.5 trees made several more mistakes than the other classifiers, while our classifier displays outstanding excellence. SVM has similar results as us, making no mistakes as well; but 3-nearest neighbour made 2 mistakes (1:1:0).

For the widely-used ALL vs AML data set Golub *et al.* (1999), the performance are also reported in Table 8. This time, our method made one more mistake than the C4.5 family algorithms on the *34 test samples*. However, our method was better than SVM (5 mistakes) and 3-NN (10 mistakes). On the other hand, for a comprehensive 10-fold cross-validation on the entire 72 samples, our method was much better than the C4.5 family algorithms by making only 1 mistake (see the last row of Table 8). In this experiment, SVM made the same mistake as our method. But k-nearest neighbour made 10 mistakes. If ad-hoc numbers (50, 100, or 200) of top-ranked features are pre-set and then used, no classifers could achieve better performance than when all the original features are considered. In fact, the performance were most often worse than. These full results can be found at our supplementary website. Once again, this indicates that openning all original features for the selection of forming our rules is a good idea indeed.

7 VARIANTS AND DISCUSSION

We have shown that our method provides highly competitive accuracy compared to C4.5, bagging, boosting, SVM, and k-NN. Our new method also provides highly comprehensible rules that help in translating raw data into knowledge. As described, our committees of trees are constructed by forcing some top-ranked features iteratively as the root node of a new tree. There are also alternative ways to construct other types of tree committees that are in accordance with our idea that the second could be the best.

One way is to extend this idea to the selection of best features for the nodes at the second level of trees instead of only applying it to root node level. Suppose we allow k number of feature choices (usually top k features) for every node, then we can build a committee of n^k trees if the trees always have n nodes. If we allow k number of feature choices for nodes only at the first two levels (the root level and its immediate children level), we can get 27 trees when $k = 3$. This approach focuses our attention on top-ranked genes either globally at root node level or locally at children nodes' level.

Another possible alternative approach is to use reduced training data formed by deleting one feature after building one tree. The first tree is constructed using the whole original data. We then remove the feature from the original data which was understood as the most important feature by C4.5. We proceed to apply C4.5 to the reduced data to generate a second tree, and so on. We have tested this idea and found the performance by those trees are collectively good.

With pre-feature selections, for example using only top-ranked features, SVM and *k*-nearest neighbour can often (but not always) increase their accuracy. We have also done another comparison study of the classification models when a feature selection is processed on training data. In general, we can still win. The full results can be found at our supplementary website.

Emerging patterns Dong and Li (1999) have been shown to be an important concept for discovering significant rules from bio-medical data Li *et al.* (2003). However, due to the inherent complexity of the patterns, mining algorithms of emerging patterns may not be sufficiently efficient when applied to high-dimension data (e.g. data dimension of 100). Our method in this work can quickly discover significant rules using wide feature spaces. However, as C4.5 is a heuristic method, our answer to discover all significant rules is still incomplete. On the other hand, the emerging pattern approach can solve the incompleteness problem if the data dimension is not that high. Combining the emerging pattern approach and the C4.5 heuristics, we may find a good way to get a close approximate to the optimal answer, and this provides for a topic for further research.

ACKNOWLEDGMENT

We thank the three reviewers for their very helpful comments.

REFERENCES

Armstrong,S.A., Staunton,J.E., Silverman,L.B., Pieters,R., den Boer,M.L., Minden,M.D., Sallan,S.E., Lander,E.S., Golub,T.R. and Korsmeyer,S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.

Bauer,E. and Kohavi,R. (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, **36**, 105–139.

Breiman,L. (1996) Bagging predictors. *Machine Learning*, **24**, 123–140.

Breiman,L., Friedman,J., Olshen,R. and Stone,C. (1984) *Classification and regression trees*. Wadsworth International Group, Belmont, CA.

Burges,C. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167.

Dietterich,T.G. (2000) An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, **40**, 139–158.

Dong,G. and Li,J. (1999) Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, San Diego, CA, pp. 43–52.

Fayyad,U. and Irani,K. (1992) The attribute selection problem in decision tree generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI Press, pp. 104–110.

Freund,Y. and Schapire,R.E. (1996) Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*. Morgan Kaufmann, Bari, Italy, pp. 148–156.

Friedman,J.H., Kohavi,R. and Yun,Y. (1996) Lazy decision trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI 96*. AAAI Press, Portland, Oregon, pp. 717–724.

Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J. Caligiuri,M.A. *et al.* (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Gordon,G.J., Jensen,R.V., Hsiao,L.-L., Gullans,S.R., Blumenstock,J.E., Ramaswamy,S., Richards,W.G., Sugarbaker,D.J. and Bueno,R. (2002) Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res.*, **62**, 4963–4967.

Li,J., Liu,H., Downing,J.R., Yeoh,A.E.-J. and Wong,L. (2003) Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. *Bioinformatics*, **19**, 71–78.

Liu,H. and Motoda,H. (1998) *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers, Boston, MA.

Pagallo,G. and Haussler,D. (1990) Boolean feature discovery in empirical learning. *Machine Learning*, **5**, 71–99.

Petricoin,E.F., Ardekani,A.M., Hitt,B.A., Levine,P.J., Fusaro,V.A., Steinberg,S.M., Mills,G.B., Simone,C., Fishman,D.A. Kohn,E.C. *et al.* (2002) Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, **359**, 572–577.

Quinlan,J.R. (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA.

Quinlan,J.R. (1996) Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI 96*. AAAI Press, Portland, Oregon, pp. 725–730.

Singh,D., Febbo1,P.G., Ross,K., Jackson,D.G., Manola,J., Ladd,C., Tamayo,P., Renshaw,A.A., D'Amico,A.V. Richie,J.P. *et al.* (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**, 203–209.

Yeoh,E.-J., Ross,M.E., Shurtleff,S.A., Williams,W.K., Patel,D., Mahfouz,R., Behm,F.G., Raimondi,S.C., Relling,M.V. Patel,A. *et al.* (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, **1**, 133–143.