Using Rules to Analyse Bio-medical Data: A Comparison between C4.5 and PCL

Jinyan Li[1] and Limsoon Wong[1]

Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore, 119613
{jinyan, limsoon}@i2r.a-star.edu.sg

Abstract. For easy comprehensibility, rules are preferrable to non-linear kernel functions in the analysis of bio-medical data. In this paper, we describe two rule induction approaches—C4.5 and our PCL classifier—for discovering rules from both traditional clinical data and recent gene expression or proteomic profiling data. C4.5 is a widely used method, but it has two weaknesses, the single coverage constraint and the fragmentation problem, that affect its accuracy. PCL is a new rule-based classifier that overcomes these two weaknesses of decision trees by using many significant rules. We present a thorough comparison to show that our PCL method is much more accurate than C4.5, and it is also superior to Bagging and Boosting in general.

1 Introduction

In the analysis of bio-medical data, rule-based classifiers such as decision trees derived by C4.5 [16] or CART [2] have an advantage over non-linear kernel function based classifiers (e.g., support vector machines or neural networks). The advantage is that rules have easier comprehensibility than non-linear functions. Also, rules can capture interactions or causal relationships among features explicitly. However, the widely used C4.5 rule induction method is less accurate than other methods. Two factors are attributed to this inferior accuracy: the single coverage constraint and the fragmentation problem [14, 7, 13].

The single coverage constraint originates from the heuristic nature of the C4.5 method. The construction of C4.5 trees is a recursive process. The process first selects a feature which is most discriminatory with regard to the entire training data. Then the process goes to use this feature to split the data into non-overlapping sub-groups so that each sub-group contains as many samples of the same class as possible. Then the process is applied to each of these sub-groups of the training data, and iteratively applied to the resulting sub-groups until all sub-groups contain pure or almost pure class of samples.

Each sub-group of training data corresponds to a rule with which all samples in this sub-group satisfy. Because no training data is repeated in these sub-groups, no training samples can satisfy more than one rule's conditions. So, every training sample has one and only one rule in the tree to be satisfied. We call this the single coverage constraint or mutual exclusivity. Due to this constraint, decision tree methods do not encourage many significant rules to be generated, though there exist lots of significant rules in bio-medical data particularly for high-dimensional gene expression profiling data [8, 18, 17, 9]. The small number of significant rules can lead to biased predictions.

The fragmentation problem is that as less and less training data are used to search for root nodes of sub-trees (sub-groups), a series of many locally important but globally unimportant rules may be generated. These minor rules can in turn mis-guide the resulting system, decreasing the accuracy of the trees.

Recently, the Bagging [3] and Boosting [6] have been proposed to help improve C4.5 classifier. However the use of bootstrapped training data can cause other problems. For example, rules are not always true. As some samples are duplicated and some are removed during training, rules induced from bootstrapped training data may not be correct when applied to the original data. This is a critical concern in bio-medical applications such as the understanding and diagnosis of a disease. Although the approach may help to improve the resulting classifiers' accuracy, the rules should be interpreted cautiously.

How to discover many significant and true rules for a classifier so that the classifier can reach a high accuracy? This is the main motivation for us to propose the PCL classifier [10]. As shown later on a wide range of bio-medical data sets, PCL is more accurate than C4.5, Bagging, and Boosting. Its performance is sometimes better than SVM and k-nearest neighbour.

We make three main contributions in this paper:

- We found that the two weaknesses of decision trees mentioned earlier quite often exist in the analysis of high-dimensional bio-medical data.
- To solve the problem, we use multiple significant rules for classification, as adapted in our PCL classifier.
- To demonstrate the effectiveness of our idea, we conduct extensive experiments to show that our PCL classifier is more accurate than the C4.5 family of algorithms.

The core knowledge patterns used in PCL are called emerging patterns (or EPs for short) [4]. By definition, an emerging pattern is an itemset whose frequency changes significantly from one class to another class. If we interpret an item as a condition—the pair of a feature and its value, then an EP can be defined as a set of conditions. The following is an example of EP, as discovered from a gene expression profiling of the prostate disease [17],

$$\{\text{gene(37720\_at)} > 215, \text{gene(38028\_at)} <= 12\}$$

This EP can be interpreted as a conjunctive set of conditions: The expression of gene 37720_at is larger than 215 and the expression of gene 38028_at is not larger than 12. This condition is satisfied by about 73% of prostate disease cells, but is not satisfied by any normal cell. Thus, this EP jumps from its 0% frequency in the normal class to a high frequency of 73% in the tumor class. This paper focuses on this type of EPs. This EP also indicates a rule that if the expression of gene 37720_at is larger than 215 and the expression of gene 38028_at is not larger than 12, then this cell is a tumor cell. So an EP is also a rule; its significance is measured by its frequency change.

Every significant EP sharply differentiates two classes of data. Our PCL classifier can discover many significant EPs using an efficient global search method, and PCL can compactly integrate the discriminating power of multiple significant EPs. In contrast, C4.5 uses a minimum number of significant rules and also uses some minor rules for classification. This is the main reason for the high accuracy of our PCL classifier.

The rest of the paper is organized as follows. In Section 2, we present examples to show decision trees having thin structures—small number of features are contained in a tree, and the tree has almost perfect training accuracy. The thin structure is directly caused by the single coverage constraint. Such thin structures often occur in high dimensional bio-medical data sets, for example, gene expression profiling data or proteomic profiling data. Usually, the thin structure cannot provide high accuracy and reliability when real test data are encountered. To overcome this problem, we break the single coverage constraint, and allow every training data to be described by multiple rules. In Section 3, we describe steps for the discovery of emerging patterns from data. In Section 4, we briefly review our PCL classifier and give an example to show how to use this classifier. In Section 5, we describe two groups of data sets for comparing C4.5 and PCL and for evaluating our ideas. One group of data sets include the widely used bio-medical data sets stored at the UCI machine learning repository. The other include recently published high-dimensional medical profiling data. These high-dimensional data sets are usually described by more than 10000 features. In Section 6, we report our experimental results on the considered data sets, and compare the performance of C4.5 and PCL. Section 7 concludes this paper with a summary.

2 Thin structure of decision trees

An ideal decision-tree based classifier should be a structure that contains a small number of features and that has perfect accuracy on both training and test data. By a thin structure, we mean that a decision tree that contains a small number of features and has perfect training accuracy, but its performance does not well support independent unknown test data.

Decision trees having a thin structure are often found in trees induced from high-dimensional bio-medical data such as gene expression profiles. We show some examples later. With these examples, we aim to explain the following three facts:

- A very tiny portion of features can establish a decision tree, and the tree can explain the training data very well.
- There are a lot of decision trees that can explain the training data similarly.
- The use of combined trees can reduce errors in classifying test samples.

These facts strongly support our idea of the PCL classifier that uses multiple significant rules for excellent classification.

We first use C4.5 [16] to discover a decision tree from a gene expression profiling data. The data consists of 215 training samples (14 MLL plus 201 others) and 112 test samples (6 MLL plus 106 others) for differentiating the MLL subtype from the other subtypes of childhood leukemia disease [18]. The data are described by 12558 features. Here each feature is a gene, having continuous expression values.

This decision tree has a structure as depicted in Figure 1. We note that this tree has only 4 non-leave nodes. That is only 4 features are used to clearly differentiate the training samples. Compared to the total 12558 number of features describing the data, the 4 features used in the tree is a very tiny fraction. We also note that this tree has perfect 100% accuracy on the 215 training samples, but it made 4 mistakes on the 112 test samples; about one third of the MLL class were wrongly classified. So, the tree did not maintain its high performance on the unseen test data.

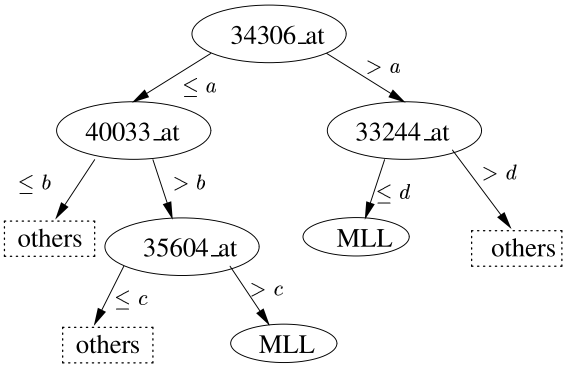

Fig. 1. A decision tree induced by C4.5 from the training data of the MLL-others data set for differentiating the subtype MLL against other subtypes of childhood leukemia [18]. Here $a = 13683.6$, $b = 3691.4$, $c = 986.9$, $d = 846.6$.

In fact, we discovered many different decision trees from this data set that have similar performance as the tree discovered by the C4.5 method. Our method for discovering decision trees is as follows: We force one of top-ranked features as root nodes sequentially to discover many decision trees. For example, to discover 10 decision trees, we use 10 of top-ranked features each as the root node of a tree. Table 1 summarizes the training and test performance, and the number of features used in our 10 trees that are discovered from the MLL-others data set. It can be seen that the 10 trees made similar numbers of errors on the training and test data in general. An interesting thing here is that the 5th, 8th, and 9th trees made smaller number of errors than the first tree made.

Table 1. The training and test errors of our 10 decision trees on the MLL-others data set that consist of 215 training and 112 test samples. The numbers of features in the 10 trees are also listed.

Tree No.	1	2	3	4	5	6	7	8	9	10
Training errors	0	0	0	0	0	0	1	0	0	0
Test errors	4	3	2	3	2	6	7	2	1	6
# of features	4	4	4	4	4	4	4	4	4	6

We also examined the performance of combined trees. If combining the first 4 trees, namely using 4 times more number of significant rules than the first tree, the committee did not made any errors on the 112 test samples, eliminating the 4 mistakes made by the first tree. Adding more trees till the 10th tree, the committee still maintains the perfect test accuracy. So, the use of multiple significant rules can reduce errors and therefore can overcome the thin structure problem of decision trees. This motivates us to discover many significant rules, for example using global search methods, from data and use them compactly for classification. In the next section, we use the concept of emerging patterns as a global search method to discover rules from data.

3 Real examples of emerging patterns

We use the following steps to discover emerging patterns [4].

1. Using the entropy method [5], rank individually the features in terms of their power to distinguish two classes.
2. Select and discretize the top-ranked features. Usually, we select the 20 top features in the analysis of high-dimensional bio-medical data.
3. Discover significant emerging patterns from the discretized data by a naive method [10] or border-based algorithms [12, 4, 11].

Next we present important emerging patterns discovered from the training data of the MLL-others data set (mentioned earlier). The importance of emerging patterns is measured by their frequency in their home class. Table 2 shows 10 important EPs, and their counts and their frequency change between the two classes.

Table 2. A partial list of top-ranked EPs discovered from the MLL-others data set. The reference numbers contained in the patterns each represent an item (a condition). For example, the reference number 3 in the first EP stands for the condition "the expression of gene $36777\_at < 5294.35$", and similarly for other numbers.

EPs	Counts (Frequency) in MLL class	Counts (Frequency) in others class
$\{1, 3\}$	0	193 (96.02%)
$\{1, 5, 7\}$	0	192 (95.52%)
$\{1, 7, 36\}$	0	191 (95.02%)
$\{1, 26\}$	0	190 (94.53%)
$\{11, 36\}$	0	187 (93.03%)
$\{13, 15, 21, 24, 39\}$	14(100.00%)	0
$\{13, 15, 21, 22, 24\}$	14(100.00%)	0
$\{13, 15, 24, 33, 39\}$	14(100.00%)	0
$\{13, 15, 22, 24, 33\}$	14(100.00%)	0
$\{9, 13, 15, 24, 33\}$	14(100.00%)	0

As already shown, the interpretation of EPs into rules is easy. Again for example, the first EP $\{1, 3\}$ of Table 2 is translated into the following rule. If the expression of gene

34306_at is less than 13630.3 and the expression of gene 36777_at is less than 5294.35, then this leukemia cell is from others class but not from MLL. The two conditions are represented by the reference number 1 and 3 respectively. This is a strong rule because about 96.02% samples from the others class satisfy this rule. Another example is the last EP—$\{9, 13, 15, 24, 33\}$—of Table 2. It is interpreted as the following rule. If the five conditions (represented by the five items $9, 13, 15, 24, 33$) are satisfied, then this cell is from the MLL class, otherwise it is from the others class.

Observe that some training samples in this MLL-others data set can be explained by different significant rules. In classifying a test sample, even when one or two significant rules are missed, the other significant rules can still help to make accurate decision. In contrast, in C4.5, if a test sample misses a significant rule, then those minor rules may sometime mis-guide the classification. Our PCL classifier [10] makes use of multiple significant rules so that some possibly biased predictions could be avoided. Next we briefly review this algorithm.

4 PCL: Prediction by collective likelihood from emerging patterns

Given two training datasets $\mathcal{D}^P$ and $\mathcal{D}^N$ and a testing sample T, the first phase of the PCL classifier is to discover EPs from $\mathcal{D}^P$ and $\mathcal{D}^N$. Denote the ranked EPs of $\mathcal{D}^P$ as, $EP_1^P, EP_2^P, \cdots, EP_i^P$, in descending order of their frequency. Similarly, denote the ranked EPs of $\mathcal{D}^N$ as $EP_1^N, EP_2^N, \cdots, EP_j^N$ also in descending order of their frequency.

Suppose the test sample T contains the following EPs of $\mathcal{D}^P$: $EP_{i_1}^P, EP_{i_2}^P, \cdots, EP_{i_x}^P$, where $i_1 < i_2 < \cdots < i_x \leq i$, and the following EPs of $\mathcal{D}^N$: $EP_{j_1}^N, EP_{j_2}^N, \cdots, EP_{j_y}^N$, where $j_1 < j_2 < \cdots < j_y \leq j$.

The next step is to calculate two scores for predicting the class label of T. Suppose we use k ($k \ll i$ and $k \ll j$) top-ranked EPs of $\mathcal{D}^P$ and $\mathcal{D}^N$. Then we define the score of T in the $\mathcal{D}^P$ class as $score(T, \mathcal{D}^P) = \sum_{m=1}^{k} \frac{frequency(EP_{i_m}^P)}{frequency(EP_m^P)}/k$, and similarly the score in the $\mathcal{D}^N$ class as $score(T, \mathcal{D}^N) = \sum_{m=1}^{k} \frac{frequency(EP_{j_m}^N)}{frequency(EP_m^N)}/k$. If $score(T, \mathcal{D}^P) > score(T, \mathcal{D}^N)$, then T is predicted as the class of $\mathcal{D}^P$. Otherwise, it is predicted as the class of $\mathcal{D}^N$.

Next we demonstrate how the classification scores are computed. Suppose $k = 5$, and the frequencies of the 5 top-ranked EPs of the positive class are sorted as 90% (EP_1^P), 85% (EP_2^P), 80% (EP_3^P), 75% (EP_4^P), and 70% (EP_5^P). Assume the test sample T contains EP_1^P (90%), EP_3^P (80%), EP_5^P (70%), EP_7^P (40%), and EP_9^P (35%). Then $score(T, \mathcal{D}^P) = (\frac{90}{90} + \frac{80}{85} + \frac{70}{80} + \frac{40}{75} + \frac{35}{70})/5 = 0.75$.

Note that $0 \leq score(T, \mathcal{D}^P), score(T, \mathcal{D}^N) \leq 1$. So, the scores appear to be like likelihood. Let's explain when $score(T, \mathcal{D}^P) = 1$ and when $score(T, \mathcal{D}^N)$ is 0. The score $score(T, \mathcal{D}^P) = 1$ if and only if the test sample T satisfies all k top-ranked positive EPs: $EP_1^P, EP_2^P, \cdots, EP_k^P$. The other score $score(T, \mathcal{D}^N) = 0$ if and only if the test sample does not satisfy anyone of the k top-ranked negative EPs. When such scores occur, the prediction is highly confident. If the two scores are close to each other, then the prediction should be taken carefully. But the tie-score cases rarely occurred in our analyses.

5 Data sets description

We use two groups of bio-medical data sets to compare the performance of C4.5 (single, Bagging and Boosting) and our PCL classifier. One group includes traditional clinical data sets stored at the widely used UCI machine learning repository [1]. The other group includes recently published high-dimensional profiling data sets such as gene expression profiles and proteomic mass/charge profiles.

We use Table 3 to summarize the background information of 10 bio-medical data sets from the UCI machine learning repository.

Table 3. Ten classical bio-medical data sets from the UCI machine learning repository. The total number of samples in a data set and the number of samples in each class are shown in the fourth column; while the third column can be used to match the data volume in a specific class.

Data sets	# of features	Class names	# of samples
Breast-w	9	2, 4	$699(=458+241)$
Cleve	13	1, 2	$303(=165+148)$
Heart	13	1, 2	$270(=150+120)$
Hepatitis	19	1, 2	$155(=32+123)$
HIV	8	0, 1	$362(=248+114)$
Hypothyroid	29	h, n	$3163(=151+3012)$
Lymph	18	2, 3	$142(=81+61)$
Promoter	60	+, -	$106(=53+53)$
Sick	29	sick, negative	$3772(=231+3541)$
Splice	60	EI, IE, N	$3175(=762+765+1648)$

The second group of data sets include 3 high-dimensional data sets for cancer diagnosis using gene expression or proteomic profiling data. Gene expression profiling has been widely used in post-genome cancer research studies [8, 18, 17, 9], 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 [15]. These technologies are breakthroughs in molecular biology because they can simultaneously measure values of thousands of features like genes' expression level or proteomic mass. The resulting data are very helpful in the subtype classification of a disease, in the classification of tumor or normal cells, in the prognosis of patients, and in the stage determination of a disease. Basically, all of these application are classical supervised learning problems. For example, in the childhood leukemia data set [18], the goal is to correctly classify subtypes of this heterogeneous disease; in the ovarian tumor data set [15], it is aimed to classify tumor and normal cells for diagnostic purpose; while in the lung cancer data set [9], it is aimed to differentiate two types of disease.

We use Table 4 to summarize the background information of 6 data sets for the subtype classification of the childhood leukemia disease. All these data are available at our Kent Ridge Bio-medical Data Sets Repository, whose URL is at `http://sdmc.lit.org.sg/GEDatasets/Datasets.html`.

Table 4. Data sets for the subtype classification of the childhood leukemia disease. The class names listed in the third column can be used to match the number of training or test samples in a specific class.

Data sets	# of features	Class names	Training size	Test size
BCR-ABL	12558	BCR-ABL, others	$9 + 206$	$6 + 106$
E2A-PBX1	12558	E2A-PBX1, others	$18 + 197$	$9 + 103$
HyperL50	12558	HyperL50, others	$42 + 173$	$22 + 90$
MLL	12558	MLL, others	$14 + 201$	$6 + 106$
T-ALL	12558	T-ALL, others	$28 + 187$	$15 + 97$
TEL-AML1	12558	TEL-AML1, others	$52 + 163$	$27 + 85$

The background information of the ovarian disease and the lung cancer disease data sets are summarized in Table 5. The two data sets are also available at our website mentioned above.

Table 5. Basic information of the ovarian disease data set and the lung cancer data set. The "—" sign represents no independent test data are available.

Data sets	# of features	Class names	Training size	Test size
Ovarian disease	15154	Cancer, Normal	$162 + 91$	—
Lung cancer	12533	MPM, ADCA	$16 + 16$	$15 + 134$

6 Comparison between C4.5 and PCL using experimental results

We report the accuracy and error rates of the learning algorithms on the considered bio-medical data sets. The accuracy of a classifier is defined as the *percentage* of samples in a data set that are correctly classified by a classifier in a stratified 10-fold cross validation or in a validation on independent test data. The error rate of a classifier is defined as the *number* of samples in a data set that are wrongly classified by a classifier in a stratified 10-fold cross validation or in a validation on independent test data. The latter is specially called *test* error rates, which is widely used in the bio-medical field. When the error rates are represented in the format $z(x : y)$, it means that x number of samples from the first class and y number of samples from the second class are misclassified, and that a total $z(= x + y)$ number samples are wrongly classified.

Our computer is a PC of DELL dimension 4100 running RedHat Linux 7.1 with a CPU speed of 886MHz and with a 512KB Ram. 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 License. Our in-house softwares like PCL are coded by C++. The C4.5 (single tree, Bagging and Boosting), SVM and k-NN programs were run under all default settings in the Weka package except that "the number of nearest neighbors to use in prediction" was reset as 3 of k-NN (the default is 1). For our PCL classifier, we set k as 5 when applied to the UCI data sets, and set k as 20 when applied to the high-dimensional bio-medical data sets.

Table 6 summarizes the performance of the classifiers on the 10 UCI bio-medical data sets. For a simple comparison, we give the following statistics numbers:

- Comparing PCL, C4.5, Bagging and Boosting, PCL won the best accuracy on 5 data sets (i.e., breast-w, cleve, heart, HIV, and promoter); Bagging won on 1 data set (hypothyroid); and Boosting won the best accuracy on 4 data sets (i.e., hepatitis, lymph, sick and splice).
- Comparing between PCL and C4.5, PCL won on 8 data sets, while C4.5 won on the rest 2 data sets.
- Comparing between PCL and Bagging, PCL won on 6 data sets, while Bagging won on 4 data sets.
- Comparing between PCL and Boosting, PCL won on 6 data sets, while Boosting won on 4 data sets.

Table 6. The performance of PCL and the C4.5 family algorithms on the 10 UCI bio-medical data sets. The accuracy is used for measuring the quality of both the learning algorithm and the data; while the error rate is used to show exact number of mistakes made in total and in each class on a data set by a learning algorithm.

Data sets	Accuracy (%)				Error rates			
	PCL	C4.5	Bagging	Boost	PCL	C4.5	Bagging	Boosting
Breast-w	96.6	94.8	96.3	96.0	24(15:9)	36(24:12)	26(14:12)	28(18:10)
Cleve	81.8	76.8	79.8	78.5	55(31:24)	70(32:38)	61(28:33)	65(33:32)
Heart	83.3	81.0	79.3	80.0	45(25:20)	50(21:29)	56(33:23)	54(25:29)
Hepatitis	80.0	78.7	80.6	83.8	31(7:24)	33(20:13)	30(19:11)	25(14:11)
HIV	91.1	85.9	85.1	89.5	32(14:18)	51(28:23)	54(30:24)	38(17:21)
Hypo	98.9	99.2	99.2	98.8	38(13:25)	25(14:11)	24(13:11)	39(17:22)
Lymph	83.0	77.5	81.7	85.2	24(13:11)	32(13:19)	26(8:18)	21(7:14)
Promoter	91.5	79.2	82.1	89.6	9(4:5)	22(10:12)	19(11:8)	11(1:10)
Sick	98.4	98.6	98.9	99.2	62(32:30)	52(30:22)	42(31:11)	30(20:10)
Splice	94.4	94.3	94.5	94.7	179 (44:42:93)	182 (23:61:98)	174 (26:54:94)	167 (27:55:85)

On a closer examination, we found that: (1) on the lymph and splice data sets where Boosting got the best accuracy, PCL had a very comparable accuracy with Boosting; (2) on the hypothyroid data set where Bagging got the best accuracy, PCL was better than Boosting. So, generally speaking, our PCL classifier is more accurate than the traditional decision-tree based single classifier like C4.5 or committee classifiers like Bagging and Boosting. We also found that the accuracy provided by the committee classifiers are all better than C4.5. So, once again, these results confirm that the use of multiple significant rules is an effective way to improve the accuracy of C4.5.

We next report experimental results on the 8 high-mensional profiling data sets. The results are summarized in Table 7 and Table 8. We can see that PCL was consistently (with only one exception on the TEL-AML1 data set) better than or equal to the performance of the C4.5 family of algorithms.

Table 7. The error rates of 7 classification algorithms on the data sets for the subtype classification of childhood leukemia.

Datasets	Error Rates for Test Data					
	PCL	C4.5	Bagging	Boosting	SVM	3-NN
BCR-ABL	1:0	1:4	2:0	1:4	1:1	1:0
E2A-PBX1	0:0	0:0	0:0	0:0	0:0	0:0
HyperL50	2:2	4:5	4:2	1:4	0:3	1:4
MLL	0:0	1:1	0:0	1:1	0:0	0:0
T-ALL	0:0	0:1	0:1	0:1	0:0	0:0
TEL-AML1	2:0	3:1	1:0	1:0	1:1	2:0
Total errors	7	21	10	14	7	8

Table 8. The error rates of 7 classification algorithms on the ovarian disease data set and the lung cancer data set.

Datasets	Error Rates for Test Data					
	PCL	C4.5	Bagging	Boosting	SVM	3-NN
Ovarian	4(3:1)	10(5:5)	8(4:4)	5(3:2)	5(2:3)	5(2:3)
Lung Cancer	3(1:2)	27(4:23)	18(2:16)	27(4:23)	1(1:0)	1(1:0)

Let's explain a bit more about the results on the lung cancer data set. The training part of this data set is small, having only 32 samples, but the test data consists about 4 times more samples than the training size. The C4.5 tree derived from this training data was very simple. It used only one feature to 100% accurately classify the 32 samples. However, this thin-structure tree made 27 mistakes on the test data. The Boosting algorithm made the same number of mistakes as C4.5 made. This is because the Boosting committee was a singleton— only one tree contained, unable to take a real committee power. This indicates that C4.5 has well learned only one aspect of the training data, and ignored many other significant rules. So, the possibility of making mistakes increases.

However, PCL discovered a total of 39 significant rules, and it used them as a committee. So, it is not a surprise to see that PCL was much better than C4.5 on this data set.

Compared to the non-linear classifiers such as SVM and nearest-neighbour, the performance of PCL is comparable and sometime is better. (See Tables 7 and 8.) As discussed previously, our advantage over SVM and nearest neighbour is that PCL can provide easily understandable rules.

We also conducted experiments to see the speed differences between PCL and C4.5. C4.5 was faster than PCL. This is because C4.5 is a heuristic search method, while PCL is a global search method. In theoretical worst cases, the candidate patterns searched by PCL is exponential to the number of features in a data set. This is a reason why PCL needs to select top-ranked features when handling high-dimensional medical data. However, on all the data sets presented in this paper, PCL completed each of the experiments within a couple of minutes, spending seconds on small data sets and longer time on other data sets. Such time are not that long for diagnostic purposes.

7 Conclusion

This paper has presented a comparison study on the performance between PCL and C4.5. Both PCL and C4.5 are rule-based classifiers. In their learning phase, explicit rules or interactions among relevant features are induced. Such a learning method differs from non-linear classifiers such as support vector machines or neural networks where the learning phase is to determine the parameters of the non-linear kernel functions. This advantage of rule-based classifiers is increasingly used in the analysis of bio-medical data.

We have compared the performance using two groups of bio-medical data sets. One group are benchmark data sets from the widely used UCI machine learning repository; the other are recently published high-dimensional profiling data for cancer diagnosis. Our experimental results have shown that PCL is more accurate than the C4.5 family of algorithms. These results indicate that the use of multiple significant rules can solve the thin structure problem of decision trees successfully.

As future work, we are planning to expand our research on the selection of top-ranked features, on the efficient discovery of emerging patterns, and on the integration methods for summarizing multiple rules.

Acknowledgement

We thank Huiqing Liu for providing some data and results.

References

1. C.L. Blake and P.M. Murphy. The UCI machine learning repository. [`http://www.cs.uci.edu/~mlearn/MLRepository.html`]. In *Irvine, CA: University of California, Department of Information and Computer Science*, 1998.
2. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.
3. Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
4. Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In Surajit Chaudhuri and David Madigan, editors, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52, San Diego, CA, 1999. ACM Press.

5. U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1029. Morgan Kaufmann, 1993.
6. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy, July 1996. Morgan Kaufmann.
7. Jerome H. Friedman, Ron Kohavi, and Yeogirl Yun. Lazy decision trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI 96*, pages 717–724, Portland, Oregon, August 1996. AAAI Press.
8. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J.R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.
9. Gavin J. Gordon, Roderick V. Jensen, Li-Li Hsiao, Steven R. Gullans, Joshua E. Blumenstock, Sridhar Ramaswamy, William G. Richards, David J. Sugarbaker, and Raphael Bueno. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62:4963–4967, 2002.
10. Jinyan Li, Huiqing Liu, James R. Downing, Allen Eng-Juh Yeoh, and Limsoon Wong. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. *Bioinformatics*, 19:71 –78, 2003.
11. Jinyan Li, Kotagiri Ramamohanarao, and Guozhu Dong. The space of jumping emerging patterns and its incremental maintenance algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA*, pages 551–558, San Francisco, June 2000. Morgan Kaufmann.
12. Jinyan Li and Limsoon Wong. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, 18:725–734, 2002.
13. Jinyan Li and Limsoon Wong. Solving the fragmentation problem of decision trees by discovering boundary emerging patterns. In *Proceedings of 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 653 – 656, Maebashi City, Japan, 2002. IEEE Computer Society.
14. Giulia Pagallo and David Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5:71–99, 1990.
15. Emanuel F Petricoin, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, and Lance A Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, 359:572–577, 2002.
16. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
17. Dinesh Singh, Phillip G. Febbo1, Kenneth Ross, Donald G. Jackson, Judith Manola, Christine Ladd, Pablo Tamayo, Andrew A. Renshaw, Anthony V. D'Amico, Jerome P. Richie, Eric S. Lander, Massimo Loda, Philip W. Kantoff, Todd R. Golub, and William R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, March 2002.
18. Eng-Juh Yeoh, Mary E. Ross, Sheila A. Shurtleff, W. Kent Williams, Divyen Patel, Rami Mahfouz, Fred G. Behm, Susana C. Raimondi, Mary V. Relling, Anami Patel, Cheng Cheng, Dario Campana, Dawn Wilkins, Xiaodong Zhou, Jinyan Li, Huiqing Liu, Ching-Hon Pui, William E. Evans, Clayton Naeve, Limsoon Wong, and James R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143, 2002.