Induction of a Novel Hybrid Decision Forest Model based on Information Theory

Limin Wang
College of Computer Science and Technology, JiLin University, ChangChun 130012, People's Republic of China.
Email: wanglim@jlu.edu.cn

Xuebai Zang and Peijuan Xu
College of Computer Science and Technology, JiLin University, ChangChun 130012, People's Republic of China.
Email: {zhangxb, xupj} @jlu.edu.cn

***Abstract*—For the task of classification, the quality of rule set is usually evaluated as a whole rather than evaluating the quality of a single rule. The present investigation proposes a hybrid classifier named FDF. By redefining information gain from the general sense of Information theory, rule sets are built and combined to be decision forest by down-top learning strategy. The finial decision tree nodes contain univariate splits as regular decision trees, but the leaves contain Naive Bayes. Empirical studies on a set of natural domains show that FDF has clear advantages with respect to the probabilistic performance.**

***Index Terms*—rule set, decision forest, Information theory**

I. INTRODUCTION

Classification is a fundamental issue in machine learning and data mining. Research in the rule induction field has been carried out for more than 30 years and has certainly produced a large number of algorithms. One of the most attractive ways to describe classification process is to use logical rules, which can be easily extracted from decision trees. However, these are usually obtained from the combination of a basic rule induction algorithm with a new evaluation function. In the present study, decision tree as rule induction extracts an initial rule set for any classification problem. One of the biggest constraints in using decision tree for data mining is the problem of scaling up the methods to handle the huge size of the data sets and their high dimensionality [1].

On the other hand, a fully trained tree is often pruned to improve the generalization accuracy and help to overcome the problem of overfitting [2]. Researchers tried to integrate all possible situations into a single tree. But decision tree structures can vary substantially when a small number of training samples are added or deleted from the training set. This instability affects the classification decisions made by the trees [3].

The hybrid approach for classification involves specific levels of knowledge where the hierarchy is defined in terms of concept granularity and specific interfaces [4]. It has been shown to be fundamental for efficient and intelligent behavior and it should be applied to learning and classification tasks as well. The concept of reductionism is a common practice in the development of intelligent systems, to design solutions to complex problems through a stepwise decomposition of the task into successive modules. In the context of classification, hybrid architectures, consisting of connectionist networks and symbolic methods, would thus combine the merits of "holistic template matching" with those of "discrete" methods using numerical and symbolic values, respectively [5].

Naive Bayes is one of the most widely used classifier in interactive applications due to its computational efficiency, competitive accuracy, direct theoretical base, and its ability to integrate the prior information with data sample information [6]. Although its conditional independence assumption is rarely valid in practical learning problems, experiments on real world data have repeatedly shown it to be competitive with much more sophisticated induction algorithms [7, 8]. Since the leaves of decision tree consist of very few samples, we suppose that the distribution of those samples approximately satisfies the conditional independence assumption. If the leaves are replaced by Naive Bayes, the advantages of both decision tree (i.e., segmentation) and Naive Bayes (evidence accumulation from multiple attributes) can be utilized simultaneously.

In this paper, we introduce an innovative hybrid model named Flexible Decision Forest (FDF) to explore this problem. The general information gain, which is defined as a scoring metric, can be used to construct logical rules and that will be finally combined to be several tree structures called decision forest. Thus the learning strategy is down-top rather than top-down.

The rest of this paper is organized as follows: Section II starts by giving the necessary background concerning the classification technique, and then the definition of information gain from the general sense of Information theory. Section III gives examples to illustrate the down-top building strategy and Naive Bayes leaf node. Section IV presents and analyzes experimental results carried out on UCI machine learning repository. Section V wraps up the discussion.

II. CLASSIFIER AND INFORMATION THEORY

Classification represents an important task in machine learning and data mining applications. Researchers

doi:10.4304/jsw.5.11.1195-1199

commonly induce a classifier from a set of historical examples (training set) with known class values and then use the induced classifier to predict the class value (the category) of new objects given the values of their attributes (features). Given the training set S consisting of n predictive attributes $\{X_1, \ldots, X_n\}$ and class label C. The classifier learned from set S should describe the relationship between $X_1, \ldots, X_n$ and C:

$$\text{Classifier: } X_1, \ldots, X_n \rightarrow C$$

Most classification algorithms tried to represent all information that training set S contains, but neglect the relationship between attribute values of test samples since they have no class label and thus considered to be incomplete. The class label of test sample $t=\{x_1, \ldots, x_n\}$ (where lower-case letters denote specific values taken by corresponding attributes. for sample, x_i represents the event that $X_i=x_i$.) is determined by the classifier induced. If the information of t is implicated in set S, the classification result may be right at best; but if not, the result may be wrong very likely.

Classical Information theory based decision tree algorithms can roughly describe the correlation that training set implicates [2]. It requires each classification process should start from the root node. But is it appropriate for all test samples? Most classification algorithms (including decision tree) learn from training set and build just one model, which is believed to match all possible situations. We think otherwise. The nature of classification problem should be: mine fully the relationship between attribute values of t based on the information provided by training set S and create a specific subclassifier to determine the corresponding class label. That is:

$$\text{SubClassifier: } x_1, \ldots, x_n \rightarrow C$$

The impact caused by other attribute values that not appear in t should be minimized. In other words, one should, first, determine the source(s) of uncertainty ingrained in our mathematical model, and then use the suitable measure of uncertainty relatively to each source. In this paper, we build one submodel for each test sample. And for this, we first redefine one basic concept of Information theory, Shannon Information gain.

Researchers are accustomed to applying Information theory to create classifier. Information theory, sometimes referred to as classical Information theory as opposed to Algorithmic Information theory, provides a mathematical model for communication. It is the theoretical foundation of modern digital communication and was invented in the 1940's by Claude E. Shannon. Though Shannon was principally concerned with the problem of electronic communications, the theory has much broader applicability. Entropy of Information theory characterizes the (im)purity of an arbitrary collection of samples.

$$Entropy(S) = -\sum_{c \in C} P(c) \log_2 P(c)$$

Definition 1. Information gain $Gain(S, X)$ of an attribute X, relative to a collection of samples S, is defined as

$$Gain(S, X) = Entropy(S) - \sum_{x \in Values(X)} \frac{|S_x|}{|S|} Entropy(S_x) \quad (1)$$

where $Values(X)$ is the set of all possible values for attribute X, and S_x is the subset of S for which attribute X has value x. The first term in (1) is just the entropy of the original collection S and the second term is the expected value of the entropy after S is partitioned using attribute X.

Note the expected entropy described by this second term is simply the sum of the entropies of each subset S_x, weighted by the fraction of samples $|S_x|/|S|$ that belong to S_x. $Gain(S, X)$ measures the expected reduction in entropy caused by knowing all the values of attribute X. That is why classical Information theory based classifier has only one model. $Gain(S, X)$ is the expected reduction in entropy caused by partitioning the samples according to X. It can be represented as:

$$Gain(S, X) = \sum_{x \in Values(X)} \frac{|S_x|}{|S|} (Entropy(S) - Entropy(S_x)) \quad (2)$$

According to the discussion described above, we only care about the attribute values that appear in sample t. From the general sense of Information theory we defined general information gain as follows.

Definition 2. General information gain $GenGain(S, x)$ of an attribute value x, relative to a collection of samples S, is defined as

$$GenGain(S, x) = \frac{|S_x|}{|S|} (Entropy(S) - Entropy(S_x)) \quad (3)$$

Obviously, when attribute X takes different values, $GenGain(S, x)$ will correspondingly take different values. Thus a dynamic subclassifier that can better match current sample will be created. Each sample corresponds to one subclassifier, but each subclassifier may correspond to several samples.

III. Flexible Decision Forest

A. Down-top Building Strategy

A standard tree induced by applying top-down strategy consists of a number of branches, one root, a number of nodes and a number of leaves. Each branch corresponds to one classification rule, which is a chain of nodes from root to a leaf; and each node involves one attribute.

In contrast, FDF proposed here applies a rather contrary building strategy: down-top. The induction process is illustrated as follows: in the building phase, the training set is recursively partitioned by discrete attribute values which maximize general information gain. Then for every partition, a new node is added to the branch. For a sample $t=\{x_1, \ldots, x_n\}$ in training set S, suppose attribute value x_1 is selected for further partitioning the set into subset T_1 which satisfies $X_1= x_1$. New node for T_1 is created and added to the branch as children of the node for S. And partition T_1 is then recursively partitioned. If

in partition T_1 all the records have identical class label then T_1 will not partitioned, and the leaf corresponding to it is labelled with the corresponding class. At last, each training sample corresponds to a specific branch or classification rule. If several rules are the same, they can express the same part of the structural information present in the data. Then combine different branches with the same root node to build a complete decision tree. As Table I shows, let S be the data set composed of eight samples $\{S_1, S_2, \ldots, S_8\}$ which are characterized by two attributes $\{A, B\}$. The first seven samples constitute training set and S_8 the test set.

The decision tree and FDF algorithm will get the structures as Fig.1 and Fig.2 show, respectively.

Take S_6 for an example, since $GenGain(S,a_1) > GenGain(S,b_2)$, attribute value a_1 is selected to further partition S into subset T_1 which satisfies $A=a_1$. Then node $A=a_1$ is created and added to the branch as one root node for S. Consequently attribute value b_2 is selected to partition T_1 into subset T_2 which satisfies $B=b_2$. Since all the records have identical class label c_2, partitioning stops and the classification rule for S_6 is created.

TABLE I.
EXAMPLE OF DATA SET S

Sample	*A*	*B*	*C*
S_1	a_0	b_0	c_1
S_2	a_0	b_1	c_1
S_3	a_0	b_2	c_1
S_4	a_1	b_0	c_2
S_5	a_1	b_1	c_1
S_6	a_1	b_2	c_2
S_7	a_2	b_0	c_2
S_8	a_2	b_1	?

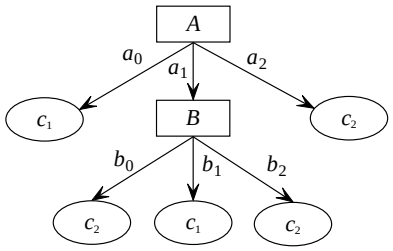

Figure 1. Decision tree structure corresponding to S.

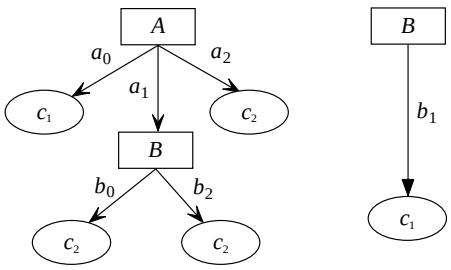

Figure 2. Decision forest structure corresponding to S.

B. Naive Bayes Leaf Node

Naive Bayes provides a simple and effective approach to classifier learning. It comes originally from work in pattern recognition and is based on one assumption that given class label C predictive attributes $\{X_1, \ldots, X_n\}$ are conditionally independent. Its structure can be depicted as Fig.3 shows.

However its attribute independence assumption rarely holds in real world problems. Many researchers try to adjust data distribution to approximate the independence assumption and then improve upon the prediction accuracy [9]. A straightforward approach to overcome the limitation of Naive Bayes is to combine its structure with other data mining models to represent explicitly the dependencies among attributes.

To classify a new sample, having only values of all its attributes, decision tree algorithms start with the root of the constructed tree and follow the path corresponding to the observed value of the attribute in the interior node of the tree. This process is continued until a leaf is encountered. Finally, we use the associated label to obtain the predicted class value of the sample at hand. But decision tree constructed from a training set usually does not retain its accuracy over the whole sample space due to over-training or over-fitting. Therefore, a fully grown decision tree needs to be pruned by removing the less reliable branches to obtain better classification performance over the whole sample space even though it may have a higher error over the training set. By contrast, the over-training problem does not exist for FDF algorithm. If an unlabeled sample does not match any decision tree, by calculating $GenGain(S, x)$ a new branch for a certain decision tree can be created. This branch is different from existing branches but can best match current sample.

The Information theory based technique improves the flexibility and scalability of decision tree algorithm greatly. As to test sample S_8, it match both subtrees. But by computing general information gain, the class label should be c_1 since $GenGain(S,b_1) > GenGain(S, a_2)$. And from Table I we can get the same result since there are two samples that satisfy $B=b_1$ and have class label c_1, but only one sample satisfies $A=a_2$ and has class label c_2. So c_1 seems much more reasonable. But from Fig.1, classical decision tree algorithm will get another result, c_2. The learning procedure of the FDF algorithm is described as follows:

Step 1: Calculate $Entropy(C)$ to identify the class in the training set S.

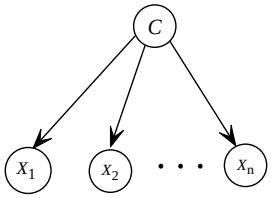

Figure 3. Example of Naïve Bayes structure.

Step 2: Given a training sample s_i with attribute values $\{x_1,\ldots, x_n\}$, calculate general information gain $GenGain(S, x_j)$ $(1\le j \le n)$ in turn.

Step 3: Generate the root node attribute X_j which maximize $GenGain(S, x_j)$ $(1\le j\le n)$. Verify if the generated node satisfies the stopping criteria.

(a) If yes, exit and the learning procedure stops;

(b) Else, node X_j is created and added to the branch as one root node for S.

Step 4: Compute for each attribute value, among those that have not been used so far, its general information gain, corresponding to the training subset which satisfies $X_j = x_j$. Children nodes are then created and added to the branch in turn until subset has the same class label or stopping criteria is satisfied. At last, the classification rule of s_i is created.

Step 5: Repeat the same process for each training sample from Step 3. Stop when all classification rules are created for training set.

Step 6: Combine classification rules with the same root node into one decision tree.

Decision forest is the most general form of classifiers, since it allows both serial and parallel combinations of arbitrary discriminators. In such methods a set of decision trees are constructed and new samples are classified by taking a vote on the results of these trees. The intuitive explanation for the success of ensemble learning is that mistakes made by individual classifiers are corrected by complementary results submitted by other classifiers in the committee. According to Occam's razor rule, a shorter assumption may be believable and a longer one is more likely to be a coincidence. If the descendant node satisfies specific stopping criterions, create a Naive Bayes as the leaf node and return.

IV. EXPERIMENTS

In order to test the soundness of the method proposed, this section describes preliminary experiments designed to compare FDF with Random Forest. The experiments were run on 10 data sets from the UCI machine learning repository. The classification performance was evaluated by 10 independent 10-fold cross-validations tests.

To construct discretizations for learning, we used a variant of the method of Fayyad and Irani [10], using only the training data, in the manner described in [11]. These preprocessing stages were carried out by the MLC++ system. And the experiments with the various learning procedures were carried out on exactly the same training sets and evaluated on exactly the same test sets. At each test, the results for Random Forests are returned by forests with 1000 trees.

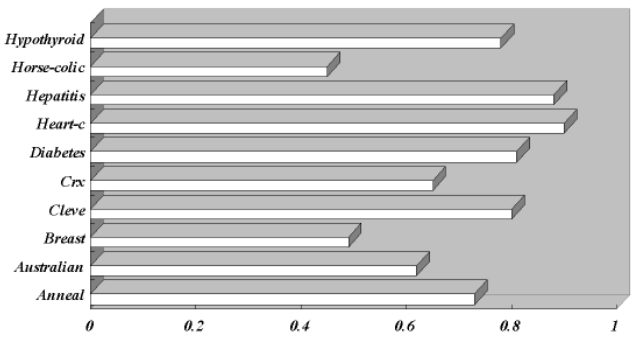

Figure 4. The P_{cost} ratio of FDF and Random Forest

In this paper, we are keen to compare the ensemble algorithms from the view of probabilistic performance. In many domains, like oncological and other medical data, not only the class predictions but also the probability associated with each class is essential. To compare probabilistic prediction performances, we use a metric called probabilistic costing (or log loss) [12], defined as

$$P_{\cos t} = -\sum_{i=1}^{n} \log(P_i) \tag{4}$$

where n is the total number of test data and P_i is the probability assigned by the model to the true (correct) class associated with test sample. The probabilistic costing is equivalent to the accumulated code length of the total test data encoded by an inferred model. As the optimal code length can only be achieved by encoding with the real probability distribution of the test data, it is obvious that a smaller probabilistic costing indicates a better performance on overall probabilistic prediction.

To obtain P_{cost} for Random Forest, the probabilistic prediction for each test sample is calculated by arithmetically averaging the probabilistic predictions submitted by each iteration. Experimental study showed that promising results can be achieved using not only single classifiers but ensembles of multiple classifiers. From Fig.4 we can see that, in terms of logarithm of probability (P_{cost}) bit costing, FDF has achieved better (lower) probabilistic costing on average compared to Random Forest. The superior performance on probabilistic prediction of the FDF can be attributed to the fact that a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is that the classifiers are accurate and diverse. To satisfy this condition, each decision tree should learn on different parts of the training set by varying the training sets and by varying the attributes (or metric) used.

The distributions of different attribute values are uneven, some may distribute closely, some may distribute sparsely. And when some attribute values are set, the conditional distribution of other attribute values will be even more complex. Just as we introduced in Section I, FDF satisfies the necessary and sufficient condition for an ensemble of classifiers to be more accurate, i.e., each of its individual subtrees is accurate and diverse. Different tree structures learned from different training set subspaces, which are incompatible and can represent multi-level semantic knowledge. But Random Forest algorithm (or other decision tree ensemble learning algorithms) tried to build different tree structures by artificially selecting different data subset and determine the tree numbers. All these human factors may affect the classification performance negatively.

V. CONCLUSIONS

The top-down induction of decision tree is one of the most popular approaches that have been used on a variety of real-world data mining tasks. In this paper, by combining Naive Bayes as leaf nodes, our novel down-top decision tree generating scheme is capable of constructing a decision forest with a large number of distinct highly performing decision trees. The proposed scheme exploits the potential of using different classification rules to improve the predictive accuracy of ensemble classifiers, especially its performance will not be affected by any human intervention. It is reasonable to believe that replacing the preliminary model with well developed and more elaborate models to approximate the posterior probabilities of the inferred trees can further enhance the results.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China (Project number: 60803055) and Education Ministry Research Project for Humanities and Social Sciences (Project number: 08JC630041).

REFERENCES

[1] Huimin Zhao, "A multi-objective genetic programming approach to developing Pareto optimal decision trees," Decision Support Systems, vol. 43, Issue 3, pp. 809-826, April 2007.
[2] B. Chandra, and P. Paul Varghese, "Fuzzifying Gini Index based decision trees," Expert Systems with Applications, vol. 36, Issue 4, pp. 8549-8559, May 2009.
[3] Hakan Altinay, "Decision trees using model ensemble-based nodes," Pattern Recognition, vol. 40, Issue 12, pp. 3540-3551, December 2007.
[4] Srinivas Gutta, and Harry Wechsler, "Face recognition using hybrid classifiers," Pattern Recognition, vol 30, Issue 4, pp. 539-553, April 1997.
[5] Bikash Kanti Sarkar, and Shib Sankar Sana, "A hybrid approach to design efficient learning classifiers," Computers & Mathematics with Applications, vol 58, Issue 1, pp. 65-73, July 2009.
[6] Eunseog Youn, and Myong K. Jeong, "Class dependent feature scaling method using naive Bayes classifier for text datamining," Pattern Recognition Letters, vol. 30, Issue 5, pp. 477-4851, April 2009.
[7] Burak Turhan, and Ayse Bener, "Analysis of Naive Bayes' assumptions on software fault data: An empirical study," Data & Knowledge Engineering, vol 68, Issue 2, pp. 278-290, February 2009,
[8] Ludmila I. Kuncheva, "On the optimality of Naïve Bayes with dependent binary features," Pattern Recognition Letters, vol 27, Issue 7, pp. 830-837, May 2006.
[9] Li-Min Wang, Xiao-Lin Li, Chun-Hong Cao, and Sen-Miao Yuan, "Combining decision tree and Naive Bayes for classification," Knowledge-Based Systems, vol 19, Issue 7, pp. 511-515, November 2006.
[10] Usama M. Fayyad, and Keki B. Irani, "Multi-interval discretization of continuous valued attributes for classification learning," in Proc. 13th International Conference on Artificial Intelligence, pp.1022-1027, September 1993.
[11] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in Proc. 12th International Conference on Mathematical Linguistics, pp.653-669, May 1995.
[12] Peter J. Tan, and David L. Dowe, "Decision Forests with Oblique Decision Trees," *LNAI*, vol. 4293, pp.593-603, Springer Berlin, November 2006.

Limin Wang received the Ph.D. degree from JiLin University, Changchun, China, in 2005.

He is currently a associate professor of college of computer science and technology, JiLin University, ChangChun, China. His research activity mainly focuses on data mining, pattern recognition, and bayesian network.

Xuebai Zang received the Ph.D. degree from JiLin University, Changchun, China, in 2002.

She is currently a professor of college of computer science and technology, JiLin University, ChangChun, China. Her research activity mainly focuses on data mining, database security and prallel computing.

Peijuan Xu received the B.S. degree from JiLin University, Changchun, China, in 1986.

She is currently a associate professor of college of computer science and technology, JiLin University, ChangChun, China. Her research activity mainly focuses on information security, formal methods.