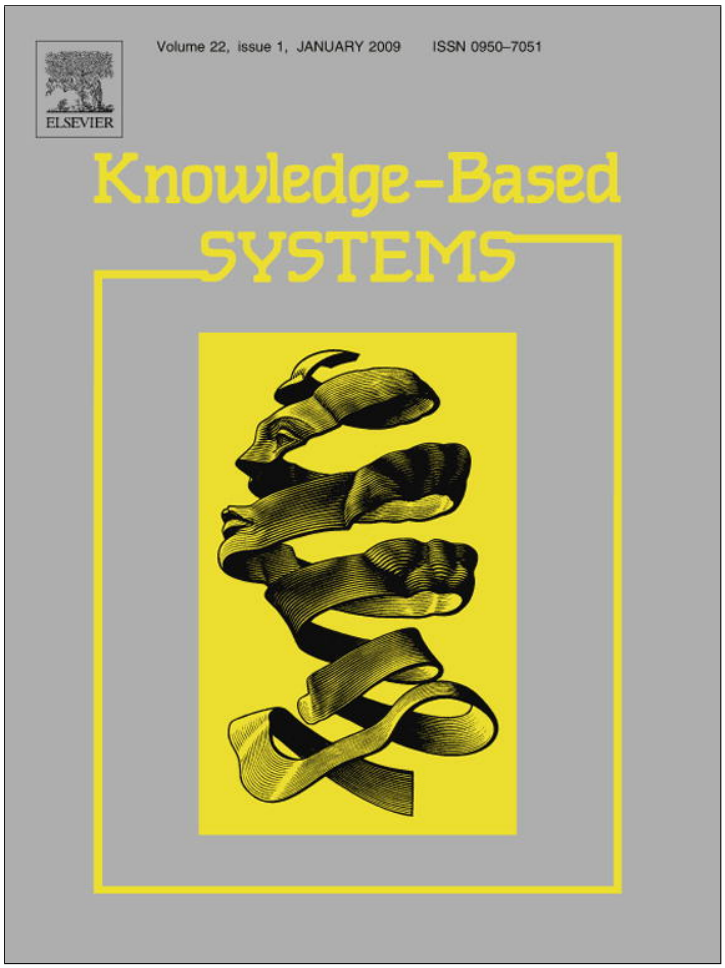

Knowledge-Based Systems 22 (2009) 1–7

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Knowledge-based system for text classification using ID6NB algorithm

Subramanian Appavu [a,*], Ramasamy Rajaram [b]

[a] Faculty, Department of Information Technology, Thiagarajar College of Engineering, Madurai, India
[b] Department of Computer Science and Information Technology, Thiagarajar College of Engineering, India

ARTICLE INFO

Article history:
Received 17 November 2007
Received in revised form 12 April 2008
Accepted 21 April 2008
Available online 1 May 2008

Keywords:
Data mining
Dimensionality reduction
Classification
Decision tree
Majority voting
Naive Bayes

ABSTRACT

This paper presents a novel algorithm named ID6NB for extending decision tree induced by Quinlan's non-incremental ID3 algorithm. The presented approach is aimed at suggesting the solutions for few unhandled exceptions of the Decision tree induction algorithms such as (i) the situation in which the majority voting makes incorrect decision (generating two different types of rules for same data), and (ii) in case of dimensionality reduction by decision tree induction algorithms, the determination of appropriate attribute at a node where two or more attributes have equal highest information gain. Exception due to majority voting is handled with the help of Naive Bayes algorithm and also novel solutions are given for dimensionality reduction. As a result, the classification accuracy has drastically improved. An extensive experimental evaluation on a number of real and synthetic databases shows that ID6NB is a state-of-the-art classification algorithm that outperforms well than other methods of decision tree learning.

1. Introduction

The process of Knowledge Discovery in Databases (KDD) is defined by Fayyad et al. [3] as "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." Data mining is the core step of the KDD process, which is concerned with a computationally efficient enumeration of patterns presenting in a database. Classification is a primary data mining task aimed at learning a function that classifies a database record into one of several predefined classes based on the values of the record attributes. Common classification methods, like Backpropagation, Naive Bayes, SVM, ID3, and C4.5, are designed to optimize the predictive performance of the induced model [4]. Other aspects of knowledge discovery, such as two different rules concerning the same data, and identification of relevant features, are given only secondary consideration by most existing algorithms. Consequently, classification models induced from real-world data are not efficiently deal with inconsistent data and are statistically insignificant. The ID6NB algorithm presented in this paper is aimed at solving these problems.

1.1. Information theory and classification

The data classification process is aimed at reducing the amount of uncertainty or gaining information about the target (classification) attribute. In Shannon's information theory (see [2]), information is defined as that which removes or reduces uncertainty. For a classification task, more information means higher accuracy of a classification model since the predicted class of new instances is more likely to be identical to their actual class. A model that does not increase the amount of information is useless and its predictive accuracy is not expected to be better than just a random guess. We also realize that more information is needed to accurately predict a multivalued outcome than to predict a binary outcome. Information theory (see [2]) suggests a general modeling of conditional dependency between random variables. If nothing is known on the causes of a variables X, its degree of uncertainty can be measured by the unconditional entropy $H(x) = \sum p(x)\log_2 p(x)$. Entropy is different from statistical variance by its metric-free nature: It depends only on the probability distribution of a random variable rather than on its concrete values. Thus, in classification tasks, where the metric of class labels is unimportant, minimizing the entropy of the target attribute can be a criterion for choosing the best hypothesis. Examples of this include the use of information gain in ID3 [9] and C4.5 [11] algorithms for finding the best feature to split a node of a decision tree. In this paper, we present, for the first time, a detailed example of ID6NB algorithm, a new way of extracting rules from the Benchmark datasets and a comprehensive comparison of our method to other decision tree induction algorithms.

1.2. Dimensionality reduction and feature selection

Minimizing the number of relevant attributes or features in a classification model is important for several reasons, from increasing the learning speed of a classification algorithm to dealing with the curse of dimensionality problem in parameter estimation. John et al. [7] distinguish between two models of selecting a "good" set

* Corresponding author.
E-mail address: sbit@tce.edu (S. Appavu).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.04.006

of features under some objective function. The feature filter model assumes selecting the features before applying an induction algorithm (by using some evaluation measures), while the wrapper model uses the prediction accuracy of the induction algorithm itself to evaluate the features. An over view of existing filter and wrapper methods for feature selection can be found in [8].The wrapper approach is usually associated with a considerable computational effort since it requires the rerunning of an induction algorithm multiple times. The filter methods, on the other hand, are computationally cheaper, but, as indicated by [8], there is a danger that the features selected by a filter method will not allow a classification algorithm to fully exploit its potential. Unlike the filter and the wrapper approaches, the ID6NB algorithm presented in this paper implements automated feature selection "on the fly" as an integral part of the learning process. Thus, a minimal subset of features is found in a single run of the induction algorithm.

1.3. Paper organization

Related works are stated in Section 2. Problem Statements are illustrated in Section 3. Section 4 describes the proposed algorithms. In Section 5, we compare the ID6NB algorithm to the most common algorithm of decision tree construction and evaluate the algorithm performance on a variety of Standard Benchmark datasets. Section 6 concludes the paper with representing a number of issues for future research.

2. Related work

The ID3 algorithm [9] is a useful concept learning algorithm because it can efficiently construct a decision tree that generalizes well. For non-incremental learning tasks, this algorithm is often a good choice for building a classification rule. However, for incremental learning tasks, it would be far preferable to accept instances incrementally, without needing to build a new decision tree each time.

There exist several techniques to construct incremental decision tree based models. Some of the earlier efforts include ID4 [12], ID5 [16], ID5R [18], and ITI [19]. All these systems work using the ID3 style "information gain" measure to select attributes. They are all designed to incrementally build a decision tree using one training instance at a time by keeping the necessary statistics (measure for information gain) at each decision node.

The ID4 algorithm [13] builds decision trees incrementally. Many learning tasks are incremental as new instances or details become available overtime. The ID4 algorithm [14] works by building a tree and updating it as new instances become available. The ID3 algorithm can be used to learn incrementally by adding each new instance to the training set as it becomes available and re-running ID3 against the enlarged training set. This is, however, computationally inefficient.

The ID5 [16] and ID5R [18] are both incremental decision tree builders that overcome the deficiencies of ID4. The essential difference is that when tree restructuring is required, because the attribute at a node does not have the lowest entropy score, any sub trees are not discarded, rather the attribute that is to be placed at the node is pulled up to the node and the tree structure below the node is retained. In the case of ID5 [16] the sub trees are not recursively updated while in ID5R [18] they are. Not restructuring the sub trees is computationally more efficient. However, the resulting sub tree is not guaranteed to be the same as the one that would be produced by ID3 [10] on the same training instances. ID5R [17] does guarantee this to be the case.

The ITI (Incremental Tree Inducer) [19] is a program that constructs decision trees automatically from labeled examples. The most useful aspect of the ITI algorithm [20] is that is provides a mechanism for incremental tree induction. If one has already constructed a tree, and then obtains a new labeled example, it is possible to present it the algorithm, and have the algorithm revise the tree as necessary. The alternative would be to build a new tree from scratch, based on the now augmented set of labeled examples, which is typically much more expensive. ITI handles symbolic and numeric variables, and missing data values. It includes a virtual pruning mechanism too.

3. Problem statement

(i) The Decision tree induction algorithm works iteratively until the end condition to decide the correct class label. But the algorithm tends to choose the class label arbitrarily when the majority voting fails.
(ii) When the Decision tree induction algorithm itself is used to determine the attribute subset, then it is called wrapper approach. In this approach, if two or more attributes have equal highest values for information gain, then the algorithm does not handle the problem efficiently.

This paper aims to suggest possible solutions to the above mentioned problems.

4. Proposed work

4.1. Exception in dimensionality reduction

Decision tree induction algorithms handle dimensionality reduction along with classification. During dimensionality reduction, the attribute with highest information gain is selected. The other possible situation i.e. "when two or more attributes have equal information gain" will create an exception. The possible and optimal solutions are given for the identification of the best attribute. The problem generated due to above mentioned exception, is the selection of worst attribute beside the optimal attribute for dimensionality reduction. The advantage of handling this exception is the optimal attribute reduction when compared to other decision tree induction algorithm under this same condition.

4.2. Resolving the exception in dimensionality reduction

Consider the depth of the decision tree drawn as'd' and in which there exists two attributes (A_i and A_j) having the same highest information gain. Some of the solutions that would be effective are as follows

1. If this situation happens in the depth 0 that is choosing root node then we temporarily draw two decision trees by having each as root node. Apply this for the given test data. Select the attribute to be chosen by the high accuracy it possesses.
2. If this situation happens in between the decision tree that is from the depth 1 to d-1(that is the level before leaves).
   a. Then Traverse through other branches of that particular node's parent and remove the attribute that never occurs at least in anyone of its branches.
   b. Else Traverse through other branches of that particular node's parent and keep the attribute that occurs as the deciding attribute of that depth.
   c. Or else if both A_i and A_j occurs while traversing the other branches. Choose the attribute that has highest information gain in its parent's depth.

The ID6NB algorithm update procedure for effective dimensionality reduction is given in Fig. 1. Here the first condition to be

Algorithm:

The ID6NB algorithm update procedure for effective Dimensionality Reduction

Input: Attribute Ai, Aj and depth parameter {0, 1… (Leaf-1)}.

Output: A decision tree with optimal dimension

Method:

1. If depth D=0 then

   Form separate decision tree for both a_i and a_j, Compare the accuracy of decision tree and assign appropriate attribute.

2. Else // i.e. the depth between D=1 to D=leaf-1

   Traverse to the parent of this node and check for occurrence

   - If (A_i occurred)

     Then A_i
   - Else if (A_j occurred)

     Then Aj
   - Else

     Both A_i and A_j has occurred or not occurred

     Choose the attribute with highest information gain among A_i, A_j from the parent of this node.

Fig. 1. The ID6NB algorithm update procedure for effective dimensionality reduction.

checked is depth of the node i.e. root node or the node lies between the root and leaf. The else statement checks the occurrence of the attributes, if both attributes hold then the information gain of those attribute in the previous level is taken in to account.

The key steps 2a, 2b, and 2c are clearly demonstrated in Fig. 2.

4.3. Exception due to the failure of majority voting

In decision tree induction algorithm, one of the terminating condition for the recursive call function is majority voting. According to majority voting, "When a table has only two attributes (class attribute and one among other attributes), the attribute with maximum number of occurrence records, is selected". We have considered the case, "What would happen if two or more attribute have equal and maximum number of occurrence of records?". The traditional algorithms fail to provide any specific solution for the above mentioned problem. In the Proposed method, the optimal solution is given with the help of probability based Naive Bayesian algorithm for selecting the class label during above mentioned situation. The problem generated due to majority voting is incorrect identification of the class label which directly leads to wrong identification of instance from test dataset. The advantage of solving majority voting condition is efficient removal of noise data from training dataset which helps in increasing the accuracy towards identification of inconsistent data from test dataset.

4.4. Resolving the problem of arbitrary selection of class label due to the failure of majority voting

In order to avoid the arbitrary selection of class label due to failure of majority voting, the assigning of class label with Naive bayes algorithm is proposed. The Naive bayes algorithm decision to give the class value would be the best solution and hence ID6NB would be the extension of the Decision Tree induction algorithm by giving best solution and not giving contradictory rules (see Fig. 3).

In the traditional decision tree induction algorithm any dataset would fall under any one of the categories, (i) Consistent data, and (ii) Inconsistent data. But with ID6NB algorithm, any dataset considered for classification would fall under any one of the categories, (i) Consistent data identified by alpha rules, (ii) Inconsistent data identified by alpha rules, (iii) Consistent data identified by beta rules, and (iv) Inconsistent data identified by beta rules (see Fig. 4).

4.4.1. Alpha and beta rules

Inconsistent data are those which are not classified in the given training dataset. Hence efficiency of the algorithm decreases due to the inability to classify the given data. The records which are considered as inconsistent are identified by alpha rules. The noisy data may occur in the training data due to human error that is the user records irrelevant value in the dataset. Hence the algorithm should be developed to handle these noisy data too.

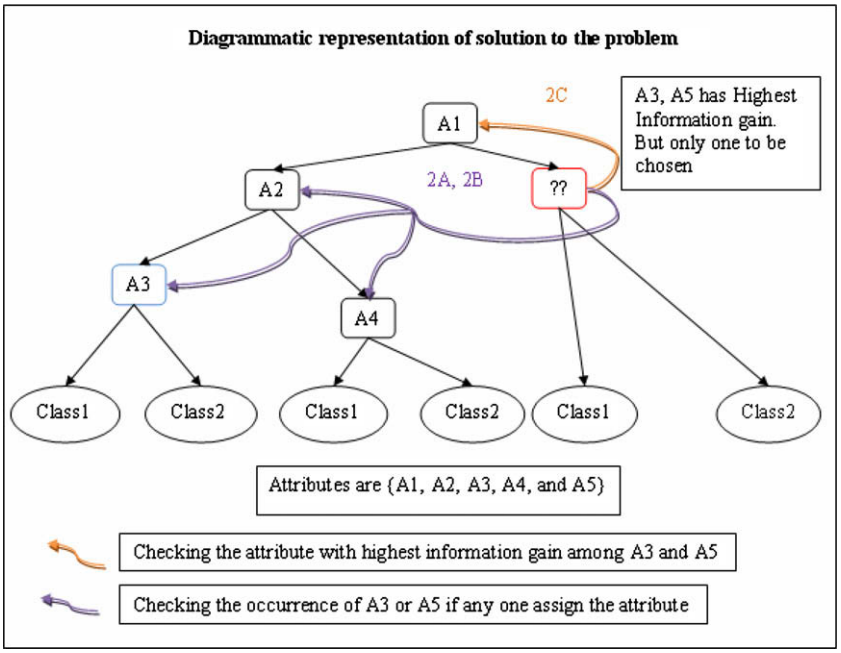

Fig. 2. A new approach to dimensionality reduction using ID6NB algorithm.

1) if attribute-list is empty then
2) if training data rules is null
3) return N as a leaf node labeled with the most common class in samples // Majority voting
   - This rule traversal from the root to the leaf is alpha rule.
   - If (record satisfy alpha rule)
     Correctly classified as (as normal DECISION TREE INDUCTION).
   - Else
     Incorrectly classified (as normal DECISION TREE INDUCTION).
4) else
5) **return N as a leaf node labeled with the attribute corresponding to class label value from probability based algorithm (Naive Bayesian algorithm)**
   - This rule traversal from the root to the leaf is beta rule.// An unique rule
   - If (record satisfy beta rule)
     Correctly classified.(This type of records cannot be handled by DECISION TREE INDUCTION algorithm)
   - Else
     Incorrectly classified.

Fig. 3. The ID6NB algorithm-decision tree induction algorithm along with updated terminating condition.

Algorithm:
ID6NB. Generate a decision tree from the given training data.
Input: The training samples, samples, represented by discrete-valued attributes; the set of candidate attributes, attribute-list.
Output: A decision tree and set of rules.

Method:

1) Create a node N;
2) if samples are all of the same class, C then
3) return N as a leaf node labeled with the class C;
4) if attribute-list is empty then
5) if training data rules is null
6) return N as a leaf node labeled with the most common class in samples // Majority voting
7) else
8) **return N as a leaf node labeled with the attribute corresponding to class label value from probability based algorithm (Naive Bayesian algorithm)**
9) select test-attribute, the attribute among attribute-list with the highest information gain;
10) label node N with test-attribute;
11) for each known value a_i of test-attribute //partition the samples
12) grow a branch from node N for the condition test-attribute= a_i ;
13) let s_i be the set of samples in samples for which test-attribute= a_i;// a partition
14) if s_i is empty then
15) attach a leaf labeled with the most common class in samples;
16) else attach the node returned by Generate_decision_tree (s_i ,attribute-list-test-attribute);

Fig. 4. The proposed ID6NB algorithm.

When a rule is formed by a set of records in which the value of the class label counts more than 50% than its counterpart, then that rule is acceptable and those instances which contradicts this rule will be recorded as the noisy data. The inconsistent records are those which are identified as noise by beta rules.

Table 1 presents a training data tuples taken from the All Electronics customer database original. (The data are adapted from [Qui86].) The class label attribute, Buys_Computer, has two distinct values (namely, {yes, no}); therefore, there are two distinct classes ($m = 2$). Let class C1 correspond to yes and class C2 correspond to no. There are 10 samples of class yes and 6 samples of class no. To compute the information gain of each attribute, the expected information needed to classify a given sample is first derived.

$$I(S1,S2) = I(10,6) = -10/16\log_2(10/16) - 6/16\log_2(6/16) = 0.954$$

Table 1
Training data tuples from the All Electronics customer database original

RID	Age	Income	Student	Credit_rating	Class: Buys_computer
1	>40	Medium	No	Fair	Yes
2	>40	Low	Yes	Fair	Yes
3	>40	Low	Yes	Excellent	No
4	>40	Medium	Yes	Fair	Yes
5	>40	Medium	No	Excellent	No
6	31…40	High	No	Fair	Yes
7	31…40	Low	Yes	Excellent	Yes
8	31…40	Medium	No	excellent	Yes
9	31…40	High	Yes	Fair	Yes
10	<=30	High	No	Excellent	No
11	<=30	Medium	No	Fair	No
12	<=30	Low	No	Fair	No
13	<=30	Low	Yes	Fair	Yes
14	<=30	Medium	Yes	Excellent	Yes
15	<=30	High	No	Fair	Yes
16	<=30	High	No	Fair	No

For age = "<=30": $s_{11} = 3$, $s_{12} = 4$ then I (s_{11}, s_{21}) = 0.
For age = "31…40": $s_{11} = 4$, $s_{12} = 0$ then I (s_{12}, s_{22}) = 0.985.
For age = ">40": $s_{11} = 3$, $s_{12} = 2$ then I (s_{13}, s_{23}) = 0.971.

Next, we need to compute the entropy of each attribute. Let us start with the attribute age. We need to look at the distribution of yes and no samples for each value of age. We compute the expected information for each of these distributions.

The expected information needed to classify a given sample if the samples are partitioned according to age is

$$E(age) = 7/16I(s_{11},s_{21}) + 4/16I(s_{12},s_{22}) + 5/16I(s_{13},s_{23}) = 0.734$$

Hence, the gain in information from such a partitioning would be

$$Gain(age) = I_{(s1,s2)} - E(age) = 0.220.$$

Similarly, we computed the Gain (income) = 0.003, Gain (student) = 0.138, and Gain (Credit_rating) = 0.029.Since age has the highest information gain among the attributes, it is selected as the test attribute. A node is created and labeled with age, and branches are grown for each of the attribute's values.

The samples are then partitioned accordingly, as shown in Fig. 5. Table A is terminated by the first termination condition of the Decision tree induction algorithm. Table B is terminated because of the third termination condition of the Decision Tree Induction algorithm. Table C is further subdivided on the basis of the information gain of the student attribute, in which Table E is further subdivided. Notice that the samples falling into the partition for age = "31…40" all belong to the same class. Since they all belong to the class yes, a leaf should therefore be created at the end of this branch and labeled with yes.

Consider Table E in Fig. 5, which cannot decide the class label's value, for this problem the algorithm, considers the path of the tree as the test data and all the remaining dataset as the training data. Hence the instances 15th and 16th in Table 1 are considered as the test data and all other records as the training data. Now this training dataset is given to the probability based algorithm (Simple Naive Bayesian) and the path of tree is considered as the test data i.e.

Age ="<=30" and student = "no" and income = "high" and credit-rating = "fair" class =?

Bayesian classifiers are statistical classifiers [5]. They can predict class membership probabilities such as the probability that a given sample belongs to a particular class. Bayesian classification is based on bayes theorem. According to bayes theorem, let X be a data sample whose label is unknown. Let H be some hypothesis, such as that the data sample X belongs to a specified class C. For classification problems, we want to determine P(H|X), the probability that the hypothesis H holds given the observed data sample

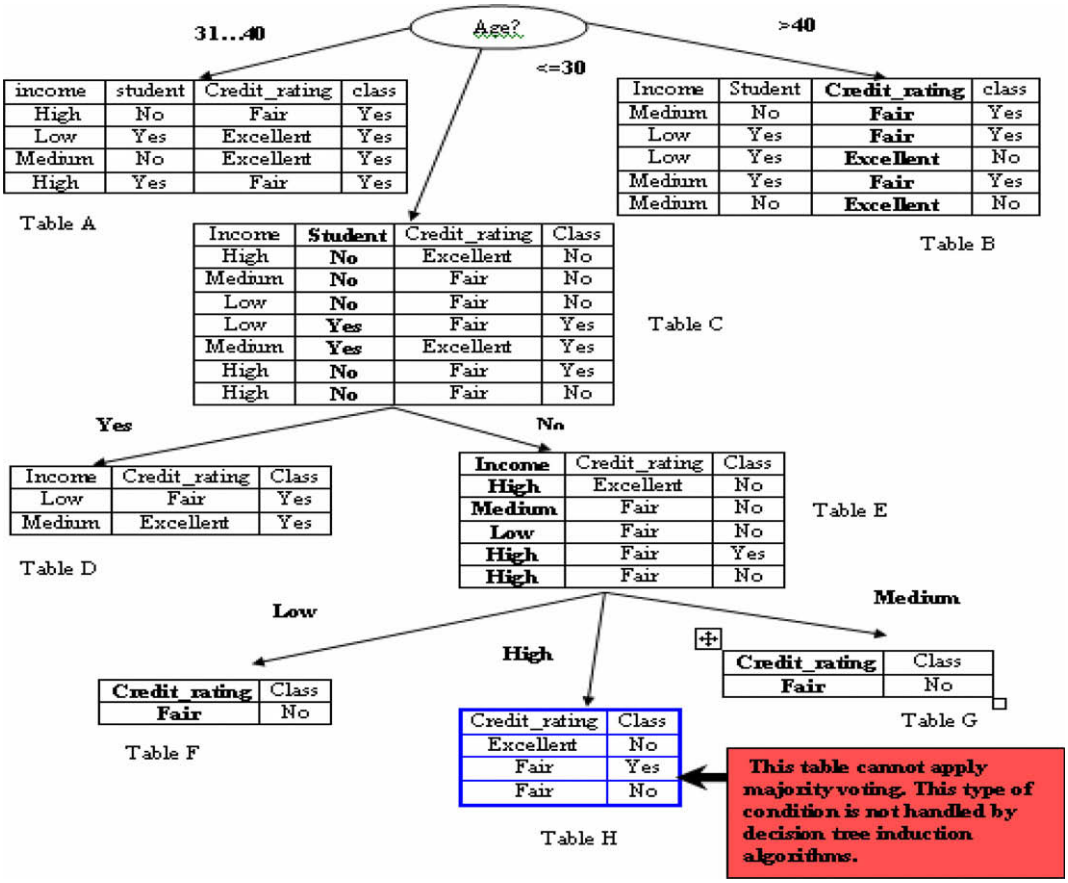

Fig. 5. The attribute age has the highest information gain and therefore becomes a test attribute at the root node of the decision tree. Branches are grown for each value of age. The samples are shown partitioned according to each branch.

X. P(X), P(H), and P(X|H) may be estimated from the given data[6]. Bayes theorem is useful in that it provides a way of calculating the posterior probability,

P(H|X), from P(H), P(X), and P(X|H). Bayes theorem is

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Let X=(age="<=30" and student="no" and income="high" and credit-rating="fair").

We need to maximize P ($X|C_i$) P (C_i), for i = 1, 2. P (C_i), the prior probability of each class, can be computed based on the training samples:

$$P(\text{class} = \text{yes}'') = 9/14 = 0.643$$
$$P(\text{class} = \text{no}'') = 5/14 = 0.357$$

To compute P ($X|C_i$), for i = 1, 2, we compute the following conditional probabilities:

$$P(\text{age} = <= 30''|\text{class} = \text{yes}'') = 2/9 = 0.222$$
$$P(\text{age} = <= 30''|\text{class} = \text{no}'') = 3/5 = 0.600$$
$$P(\text{student} = \text{no}''|\text{class} = \text{yes}'') = 3/9 = 0.333$$
$$P(\text{student} = \text{no}''|\text{class} = \text{no}'') = 4/5 = 0.800$$
$$P(\text{income} = \text{high}''|\text{class} = \text{yes}'') = 2/9 = 0.222$$
$$P(\text{income} = \text{high}''|\text{class} = \text{no}'') = 2/5 = 0.400$$
$$P(\text{credit} - \text{rating} = \text{fair}''|\text{class} = \text{yes}'') = 6/9 = 0.666$$
$$P(\text{credit} - \text{rating} = \text{fair}''|\text{class} = \text{no}'') = 3/5 = 0.600$$

Using the above probabilities, we obtain

$$P(X|\text{class} = \text{yes}'') = 0.222 * 0.333 * 0.222 * 0.666 = 0.01093$$
$$P(X|\text{class} = \text{no}'') = 0.666 * 0.800 * 0.400 * 0.600 = 0.12787$$
$$P(X|\text{class} = \text{yes}'')P(\text{class} = \text{yes}'') = 0.01093 * 0.643 = 0.0070$$
$$P(X|\text{class} = \text{no}'')P(\text{class} = \text{no}'') = 0.12787 * 0.357 = 0.0456$$

Therefore, the Naive Bayesian classifier predicts class = "no" for sample X.

Hence the class value is assigned based on the given training dataset where majority voting is not possible. The rules which are developed in this manner are called as the "beta rules" because it has the unique feature of handling the exceptions due to majority voting.

5. Experimental results and performance evaluation

5.1. Overview

The performance of the ID6NB algorithm was evaluated on 12 publicly available datasets: All Electronics customer database original, All Electronics customer database extended, breast cancer, chess endgames, credit approval, diabetes, glass identification, heart disease, Iris plants, liver, lung cancer, and wine. All these datasets are posted at the UCI Machine Learning Repository [1] and widely used by the data mining community for evaluating learning algorithms [21]. The data sets selected by us here com-

prise a diverse in nature. The datasets reported in the literature to possess majority voting problem are purportedly chosen for experimentation to prove the efficacy of our algorithm i.e. two records having the same attribute values but different class value. Another problem is the attributes having equal highest information gain. These datasets are efficiently classified and the attribute reduction done to optimal level.

5.2. Performance evaluation on monk dataset

The experiments measuring, the performance of the proposed algorithm was conducted. The proposed algorithm performance is compared with various existing incremental algorithms. The classification accuracy of other algorithms is based on the results published in the literature by [15].

Another experiment measuring the performance of proposed algorithm was conducted based on the results published in the literature by [15].

Tables 2 and 3 shows, for each dataset, the estimated predictive accuracy of the ID6NB versus other decision tree methods. As one can see from Tables 2 and 3, the predictive accuracy of the ID6NB tends to be better than the accuracy of other decision tree induction algorithms.

5.3. Dimensionality reduction

Dimensionality reduction is an important objective of the knowledge discovery process. Most real-world datasets contain some portion of completely irrelevant attributes. Unlike the Naive Bayes classifier, which uses all attributes in a dataset, the decision tree algorithm tense to remove the irrelevant attributes from the final tree (see [11]). The ID6NB algorithm, presented above, is also aimed at minimizing the set of input attributes required for classification. Table 4 shows the initial no of input attributes in each dataset, the no of input attributes selected by the evaluated algorithms(ID3,C4.5, and ID6NB), and the reduction in data dimensionality(the portion of input attributes that were excluded from the model).The training datasets included all records of each dataset. Table 4 also compares the complexity of the resulting models in terms of the run times of the algorithms on a Pentium IV computer.

The results show that the models produced by the ID6NB algorithm are significantly smaller than the decision trees built by the ID3 and C4.5. Thus, C4.5 failed to remove more than 50 percent of the attributes in seven datasets out of 12. On the other hand, the ID6NB algorithm never included more than 50 percent of available attributes. The average difference between the two methods is 28.75 percent of the no of available attributes. This means that the ID6NB algorithm is a much more "aggressive" dimensionality reducer than C4.5. ID3 tends to use fewer attributes than C4.5, but its average no of selected attribute (5.08) is still higher than the ID6NB average (3.58) (see Fig. 6).

5.4. Predictive accuracy

There are commonly four approaches for estimating the accuracy such as using training data, using test data, cross-validation, and percentage splitting. Table 5 shows, for each dataset, the estimated predictive accuracy of the ID6NB verses other decision methods. As one can see from Table 5, the predictive accuracy of

Table 2
Predictive accuracy – comparison to other methods

Dataset	ID5R (%)	IDL (%)	ID5R-hat (%)	TDIDT (%)	ID6NB (%)
Monk-1	81.7	97.2	90.3	75.7	97.2
Monk-2	61.8	66.2	65.7	66.7	74.53

Table 3
Predictive accuracy – comparison to other methods

Dataset	ID3 (%)	ID3, no windowing (%)	ID5R (%)	ID6NB (%)
Monk-1	98.6	83.2	79.7	98.6
Monk-2	67.9	69.1	69.2	73.33
Monk-3	94.4	95.6	95.2	98.6

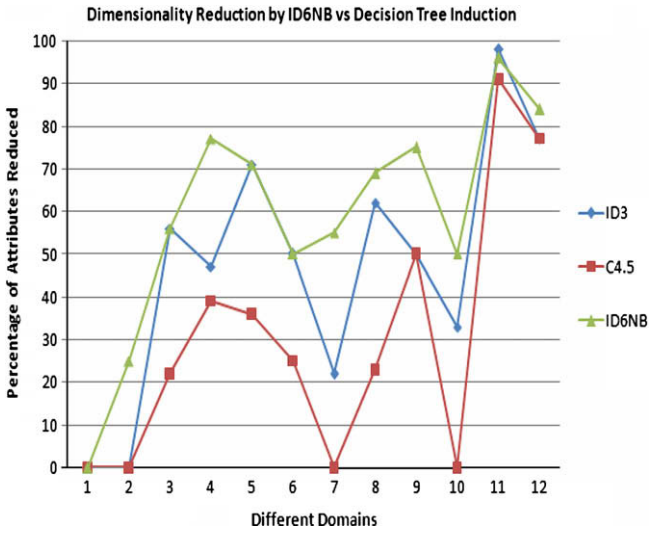

Fig. 6. Dimensionality reduction.

Table 4
Dimensionality reduction – summary table

Dataset	Available input attributes	Selected input attributes			Dim. reduction (%)			Run time (s)		
		ID3	C4.5	ID6NB	ID3	C4.5	ID6NB	ID3	C4.5	ID6NB
All Electronics original	4	4	4	4	0	0	0	0.01	0.05	0.1
All Electronics extended	4	4	4	3	0	0	25	0.01	0.05	0.1
Breast	9	4	7	4	56	22	56	0.06	0.05	0.12
Chess	36	19	22	8	47	39	77	0.55	0.33	0.8
Credit	14	4	9	4	71	36	71	0.33	0.11	0.3
Diabetes	8	4	6	4	50	25	50	0.61	0.11	0.35
Glass	9	7	9	4	22	0	55	0.27	0.11	0.3
Heart	13	5	10	4	62	23	69	0.11	0.05	0.16
Iris	4	2	2	1	50	50	75	0.05	0.00	0.1
Liver	6	4	6	3	33	0	50	0.06	0.06	0.12
Lung cancer	57	1	5	2	98	91	96	0.05	0.00	0.1
Wine	13	3	3	2	77	77	84	0.16	0.06	0.18
Mean	14.75	5.08	7.25	3.58	47.17	30.25	59	0.19	0.08	0.23

Table 5
Predictive accuracy – comparison to other methods

Dataset	ID3	C4.5	ID6NB	ID6NB-Min	ID6NB-Max
All Electronics original	93.75	95	98	97.2	98.9
All Electronics extended	90.625	93.5	93.75	92.5	94.75
Breast	93.6	94.4	93.6	92.6	94.6
Chess	99.1	99.2	99.1	98.1	99.5
Credit	83.1	85.9	84.1	83.1	85.1
Diabetes	73.3	73.5	73.3	72.3	74.3
Glass	63.8	67.9	64.6	63.1	65.2
Heart	74.3	77.5	75.7	74.7	76.6
Iris	94.9	92.6	95.6	94.5	96.6
Liver	63.5	65.9	63.1	62.1	64
Lung cancer	33.4	40.9	35.5	34.5	36.8
Wine	91.3	92.4	91.3	90.3	92.5
Mean	79.55	81.55	80.63	79.58	81.57

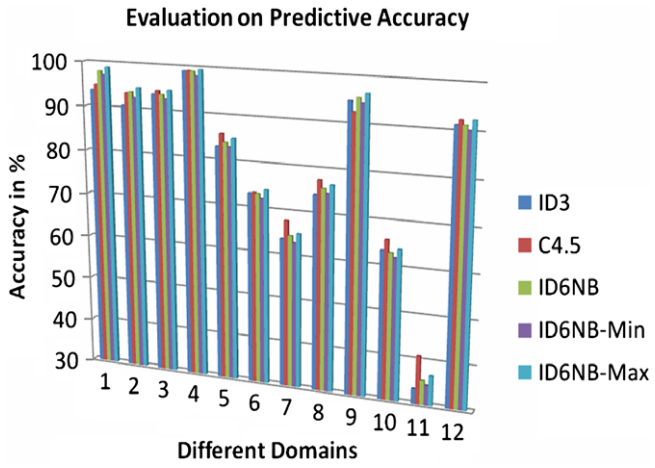

Fig. 7. Evaluation on predictive accuracy.

the ID6NB tends to be only slightly worse than the accuracy of C4.5. One exception is the All Electronics original, All Electronics extended, and Iris dataset, where the ID6NB has provided better results than C4.5 along with reducing dimensionality and solving the exceptions caused by majority voting. In all other datasets, a small lose of accuracy (the mean difference of less than 1 percent) is compensated by considerable reduction in the number of input attributes. ID3 does not show any advantage at all, since it has the lowest average accuracy while using more input attributes than ID6NB. Though the choice of the best model (either the most accurate or the simplest) depends on a specific application, in many cases a small amount of accuracy can be sacrificed for the sake of obtaining a compact and interoperable model, like the one produced by the ID6NB algorithm (see Fig. 7).

6. Conclusion

In this paper, we have presented a novel algorithm for building simple and reasonably accurate classification model, termed ID6NB.In the earlier version of ID3, the tree is constructed based upon the information gain of the attributes, the concept of majority voting and other terminating conditions. The later versions of ID3 algorithm such as ID4, ID5, and ID5R focus on optimizing the trees. We studied the unhandled exceptions of the Decision Tree induction algorithm and improved its performance by fusing data cleaning, dimensionality reduction, and data smoothening with the algorithm ID6NB. In the proposed algorithm, the exception due to majority voting is resolved with the help of probability based Naive Bayesian algorithm. Since majority voting problem is corrected, it paves way to handle noisy data, thereby helpful in data smoothing. The highlight of the algorithm is the dimensionality reduction and Classification.

References

[1] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[2] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, 1991.
[3] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, in: U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996, pp. 1–36.
[4] Ian H. Witten, Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufman Publishers, 2000.
[5] James Joyce, Bayes theorem, Standford encyclopedia of philosophy, 2003.
[6] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, 2005.
[7] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Proceeding of the 11th Int'l Conf.Machine Learning, 1994, 121–129.
[8] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data mining, Kluwer Academic, Boston, 1998.
[9] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
[10] J.R. Quinlan, Simplifying decision trees, International Journal of Man-Machine Studies 27 (1987) 221–234.
[11] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman, 1993.
[12] J.C. Schlimmer, R. Granger Jr., Beyond incremental processing: tracking concept drift, Proceedings of AAAI 1 (1986) 502–507.
[13] J.C. Schlimmer and D. Fisher, A case study of incremental concept induction, in: Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelpha, PA, Morgan Kaufmann, 1986, 496–501.
[14] J.C. Schlimmer, R.H. Granger Jr., Incremental learning from noisy data, Machine Learning 1 (1986) 317–334.
[15] S. Thrun, J. Kreuziger, R. Hamann, W. Wenzel, et.al., The MONK's Problems: A Performance Comparison of Different Learning Algorithms, Tech. Report CMU-CS-91-197, Computer Science Department, Carnegie Mellon University,1991.
[16] P.E. Utgoff, ID5: An incremental ID3, in: Proceedings of the Fifth International Conference on Machine Learning,Morgan Kaufmann Publishers, San Mateo, California, 1988, 107–120.
[17] P.E. Utgoff, Improved training via incremental learning, in: Proceedings of the Sixth International workshop on Machine Learning. Ithaca, Ithaca, New York, United States, 1989.
[18] P.E. Utgoff, Incremental induction of decision trees, Machine Learning 4 (1989) 161–186.
[19] P.E. Utgoff, An improved algorithm for incremental induction of decision trees, in: Proceeding of the 11th Int'l Conf. Machine Learning, 1994, 318–325.
[20] P.E. Utgoff, Decision tree induction based on efficient tree restructuring, in: Journal of Machine Learning, Springer, 2004, pp. 5–44.
[21] WEKA-Open Source Collection of Machine Learning Algorithm.