Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

Ensemble with Neighbor Rules Voting

Itt Romneeyangkurn, Sukree Sinthupinyo *

Abstract—**Ensembles of classifiers have been employed to improve accuracy over single classifier. Various methods sequentially bootstrap data set and invoke a base classifier on these different bootstraps. In this paper, we propose an idea based on the use of "similar rules" or "neighbor rules" in voting for the given test example, instead of using only the rule that matches with the test example. From our experimental results, we can conclude that our method achieves comparable accuracy and is significantly better than regular majority vote. We also empirically derive the least of value of a similarity between rules that gives more accurate result.**

Keywords: ensemble of classifiers, neighbor rules, majority vote, decision trees, bootstrapping

1 Introduction

Ensemble is one of methods which have been investigated to improve accuracy of classification over the use of single learner. Ensembles are groups of classifiers in which the individual classifiers have their own predictions combined to classify new examples. Ensembles have been found to be more accurate than individual classifiers when the ensemble consists of classifiers that make errors on different examples.

A simple method for constructing ensembles is Breiman's Bagging technique [2]. Bagging has been shown to work well with unstable algorithms for constructing classifiers. Unstable algorithms are defined as algorithms that significantly change the induced classifier with minor perturbations in the training examples. Breiman found decision tree induction (e.g., Quinlan's ID3) to be unstable [3] and found up to 47% improvement when using an ensemble of bagged decision tree classifiers compared to a single tree classifier[2]. Furthermore, Bagging can be used to create an ensemble by training individual classifiers using a bootstrap replicate of the training examples. A bootstrap replicate is a set of examples of size m that is drawn with replacement from m training examples. Consequently, the rules that are established from different bootstraps must have similarity between each other and seem to be the same, as the value of similarity is high enough. This is because the set of examples is collected from the same original data set. Therefore, the final prediction from majority vote of neighbor rules may give us more accuracy. Due to this idea, we present a technique to construct Majority Rule$^+$ and Majority Class$^+$ ensemble of decision tree by adding similar rule in voting along with the exactly matching rules and use all of these rules to make the final decision. In our experiments, our methods improve the accuracy and comprehensibility of both Bagging and Simple Majority Class.

Section 2 briefly introduce the bootstrapping method, Simple Majority Vote and Simple Majority Class, and explain the formula to find similarity between rules both of discrete and continuous attributes. Section 3 presents a brief overview of the algorithm, which we applied from regular majority vote to use with neighbor rules. Section 4 describes experimental results comparing these new methods to the original Bagging and Sample Majority Class procedure using many well-known data sets, and shows the appropriate value of similarity between rules that gives us more accuracy. Section 5 provide further improvements with similarity between rules. Finally, Section 6 concludes this paper.

2 Preliminaries

2.1 Bootstrapping

The bootstrap (Efron, 1979; Efron & Tibshirani, 1993) is a computer-based method to estimate the standard error of a parameter. Bootstrap samples, also called replications, are created by uniformly sampling m times with replacement from a dataset of size m. Some instances in the original data set may not appear while others may appear multiple times. The bootstrap samples are used to train the multiple classifiers or rule induction algorithm in ensemble style.

Bootstrap sampling underlies the machine learning method of Bagging classifiers (Breiman, 1996), which is an acronym for "Bootstrap AGGregatING". Breiman applied this technique to CART classification trees and nearest neighbor classifiers. Kohavi (1995) provides another example of applying bootstrap sampling to accuracy estimation for C4.5 decision trees and Naïve Bayes classifiers. Breiman conducted trials with between 10 and 100 bootstrap replications and Kohavi's experiments varied from 1 to 100 bootstrap replications. Breiman found that most of the improvement in bagging was gained with only 10 bootstrap replications.

*Thammasat University Department of Computer Science, Pathumthani Thailand Email: sukree@cs.tu.ac.th

2.2 Simple Majority Vote and Simple Majority Class

Simple Majority Vote and Simple Majority Class method are based on Bagging method that is implemented as follows. Given a test instance and the training set, Bagging generates T bootstrap samples of the original training set, where T is the ensemble size. Each bootstrap sample is generated by uniformly sampling m instances into each bootstrap from the training set with replacement, where m is the size of the original training set. Each bootstrap sample is then used as the training set to build a decision tree for classifying the given test instance. At this point, a majority vote amongst the resulting T decision trees is then used as the final output of the Simple Majority Vote. Whereas, Simple Majority Class, the final decision is derived by using majority vote among the class of training set that match the rule in T decision trees.

2.3 Similarity Between Rules

Gower proposed a method for calculating similarity measures for variables of mixed type, both quantitative (continuous) and discrete:

$$S_{ij} = \frac{\sum_{k=1}^{p} W_{ijk} S_{ijk}}{\sum_{k=1}^{p} W_{ijk}} \tag{1}$$

S_{ijk} is the similarity between the i^{th} and j^{th} individuals as measured k^{th} variable.

W_{ijk} is typically 1 or 0 depending on whether the comparison is considered valid for the k^{th} variable.

2.3.1 Similarity Calculation for Continuous Attributes

For two rules X_i and X_j, let us examine one continuous attribute, k.

Let $X_{ik\ min}$ be the minimum value of k specified by the first rule.

Let $X_{jk\ min}$ be the minimum value of k specified by the second rule.

Let $X_{ik\ max}$ be the maximum value of k specified by the first rule.

Let $X_{jk\ max}$ be the maximum value of k specified by the second rule.

$R_{ik} = X_{ik\ max} - X_{ik\ min}$ is the range of k for the first rule.

$R_{jk} = X_{jk\ max} - X_{jk\ min}$ is the range of k for the second rule.

R_k is the range of k across all the examples.

Let $X_{jk\ max} > X_{ik\ max}$, so we can refer to the X_j rule as having a higher upper boundary for variable k when compared to X_i. Using these definitions, continuous attribute similarity falls into three categories as follows:

1. Dissimilar rules, with no overlap, $X_{ik\ max} < X_{jk\ min}$

$$S_{ijk} = \frac{X_{ik\ max} - X_{jk\ min}}{R_k - R_{ik} - R_{jk}} \tag{2}$$

2. Similar rules, with overlap, $X_{ik\ max} > X_{jk\ min}$

$$S_{ijk} = \frac{X_{ik\ max} - X_{jk\ min}}{(R_{ik} - R_{jk})/2} \tag{3}$$

3. Implicitly similar rules

$$S_{ijk} = \frac{R_{ik}}{(R_{ik} + R_k)/2} \tag{4}$$

Where the attribute value is specified for rule i but not rule j.

2.3.2 Similarity Calculation for Discrete Attributes

Discrete attribute similarity falls into seven categories as follows:

1. If only one of the rules specifies an attribute and the operator is "equal to" then similarity is

$$S_{ijk} = \frac{2}{1+N} \tag{5}$$

Where N is the number of possible values for the discrete attribute.

2. If only one of the rules specifies an attribute and the operator is "not equal to" then the similarity is

$$S_{ijk} = \frac{2N-2}{2N-1} \tag{6}$$

This generalizes when the one rule specifies an attribute as "not equal to" multiple values. If there are M "not equal to" expressions for the attribute, then the similarity is

$$S_{ijk} = \frac{2N-2M}{2N-M} \tag{7}$$

3. If $X_{ik} = X_{jk}$, the attribute values are identical, and the operators for each attribute are the same, then

$$S_{ijk} = 1 \tag{8}$$

4. If $X_{ik} = X_{jk}$, the attribute values are identical, but the operators for each attribute not equal, then

$$S_{ijk} = -1 \tag{9}$$

5. If $X_{ik} \neq X_{jk}$, the attribute values are different, but the operators are "equal to", then

$$S_{ijk} = -1 \tag{10}$$

6. If one of the rules specifies "equal to" an attribute value and the other rule specifies "not equal to" a different attribute value then the similarity is

$$S_{ijk} = \frac{2}{N} \tag{11}$$

This is expanded when the "not equal to" operator is used more than once in one of the rules. If specified M times, then the similarity is

$$Sijk = \frac{2}{1 + N - M} \tag{12}$$

Note that (5) is a special case of (12), where $M = 0$.

7. Finally, when both rules specify $\neq$ to the same attribute but they specify multiple attribute values, the similarity is based on the portion of the ranges shared by the rules minus the portion of the ranges which are not shared.

$$S_{ijk} = \frac{2(N - 2(M_1 \cup M_2) + (M_1 \cap M_2))}{2N - M_1 - M_2} \tag{13}$$

This measure results to a minimum score of -2 instead of -1. A score of -2 occurs when all of an attribute's values appear in a $\neq$ expression of one rule, but no value appears in $\neq$ expressions of both rules (i.e., $M_1 + M_2 = N$). Essentially, this measure double counts the number of unshared ranges. Rather than correcting for this exactly, an adequate approximation is to divide by 2 when the score is negative, thus yielding -1 as a minimum. Note that (6) is a special case of (13), where either M_1 or M_2 is 0.

For more information about the similarity between rules, see [8].

3 Majority Rule$^+$ and Majority Class$^+$

The basis of Majority Rule$^+$ and Majority Class$^+$ are similar to Simple Majority Rule and Majority Simple Class, respectively, but they are different number of rules involving in voting procedure. After we build a decision tree, we collect every rule from all trees and calculate similarity between these rules. All rules are then made into groups, and these groups are added with rules which have similarity values according to our experiment, 0.6, 0.7, 0.8 and 0.9. A majority vote among the class of rule-member is then used as the final output of the Majority Rule$^+$ ensemble. Whereas, the final decision of Majority Class$^+$ is derived by majority vote among the class of training set that matches the rule-member.

4 Experiments

In this section we compare Majority Rule$^+$ with its base learner Bagging, and Majority Class$^+$ with Simple Majority Class. Fifteen well-known benchmark data sets from the UCI data collection (Blake & Merz, 1998) are used in these comparisons. The characteristics of these data are summarized in columns 1-4 of Table 1. For each data set, ten runs of a stratified ten-fold cross-validation are conducted and the reported results are averaged over the ten runs. An ensemble size of ten is used for all ensemble methods. Note that we implemented all method algorithms based on the C4.5 source code and C4.5 was run in its default mode with pruning enabled.

4.1 Comparison of Majority Rule$^+$ and Bagging (Simple Majority Rule)

Table 2 presents the accuracies of Bagging and Majority Simple Rule$^+$, by using of similarity value between rules 0.6, 0.7, 0.8 and 0.9 in columns 2, 3, 4, 5 and 6, respectively. A paired t-test with 0.1-level is used to compare the new method to their base method. In Table 2, "$\oplus$" is used to indicate a significantly better performance than Bagging and "$\ominus$" is used to indicate a significantly worse performance than Bagging.

Amongst the fifteen data sets, similarity value at 0.8 and 0.9 yield the better result than similarity value at 0.6 and 0.7. Using Majority Rule$^+$ decision trees, at these similarity values, both of similarity value at 0.8 and 0.9 perform significantly better than the base learner for three and two data sets, respectively, while perform significantly worse for only one data set. Whereas, similarity value 0.6 and 0.7 show nearly the same performance deteriorations over bagging for three and four data sets, respectively. Besides this, they are more accurate for two and one data set, respectively.

Table 1: The Characteristics of the Data

Data Set	Instances	Attributes	Classes
AUDIOLOGY	180	69	24
AUSTRALIAN	621	14	2
BALANCE-SCALE	625	4	4
BRIDGES	105	10	6
CAR	1,728	6	4
DERMATOLOGY	366	34	6
HAYES-ROTH	132	4	3
HEART	243	13	2
HEPATITIS	140	13	2
HORSE-COLIC	270	23	2
LABOR-NEG	40	16	2
LIVER-DISORDERS	311	6	2
SOYBEAN	307	35	19
TAE	151	3	3
ZOO	101	16	7

4.2 Comparison of Majority Class$^+$ and Majority Simple Class

Table 3, in the same way as Table 2, presents the accuracies of Simple Majority Class and Majority Class$^+$, by using of similarity between rules value 0.6, 0.7, 0.8 and 0.9 in columns 2, 3, 4, 5 and 6, respectively. Also, a paired t-test with 0.1-level is used to compare the new method to their base method. In Table 3, "$\oplus$" is used to indicate a significantly better performance than Simple Majority Class and "$\ominus$" is used to indicate a significantly worse performance than Simple Majority Class.

Similarity values 0.8 and 0.9 perform significantly better than Simple Majority Class for two data set. On the other hand, similarity value 0.8 gives the worse result for one data set; meanwhile, there is no worse result in similarity value 0.9. Next, similarity value 0.7 is worse than 0.8 and 0.9, by performing six worse data sets. Finally, similarity value 0.6 gives us the worst results by yielding nine worse data sets, with no better data set.

4.3 The Appropriate Similarity Value

From Table 2 and Table 3, we conclude that similarity values 0.6 and 0.7 are too low to use as similarity between rules in making a neighbor rule. Intuitively speaking, using the similarity value of 0.6 and 0.7 is too loose. Too many rules, which are not actually similar with the exactly matching rules, involve in voting. This can reduce the accuracy of the classifications. On the other hand, similarity values 0.8 and 0.9 perform well and yield appropriate results. Therefore, the least similarity between rules that gives us improved accuracy is 0.8.

5 Further Improvement with Similarity Between Rules

We can see that the improvement of Majority Rule$^+$ and Majority Class$^+$ over the traditional Simple Majority Vote and Simple Majority Class is caused by applying weight based on the number of neighbor rules. Therefore, the number of classifiers is still unchanged. Thus, in the next step of developing decision trees, we can cluster the rules by using similarity value and derive to one classifier. This shortens the ensemble process, and also reduces the time and resource required. Finally, the similarity value 0.8 could be used to apply with other decision trees methods, such as AdaBoost etc., to develop better results.

6 Conclusions

Bootstrapping is based on creating random instance from the same original training set. In this paper, we have proposed an idea of using similar rules from each bootstrapping to involve in voting, it should derive better accuracy. We conducted experiments by implementing a new method called Majority Rule$^+$ and Majority Class$^+$, which use neighbor rules in voting. By using the similarity values of 0.6, 0.7, 0.8 and 0.9, and compare the result with regular majority vote. The results of our comparisons show that our method yields the better percent accuracy, of which the similarity value 0.8 is the most

Table 2: Comparing Bagging and Majority Rule$^+$

Data Set	Bagging	Majority Rule$^+$			
		0.6	**0.7**	**0.8**	**0.9**
AUDIOLOGY	78.64±6.56	77.60±7.54	77.14±5.40	76.62±5.92	79.60±4.74
AUSTRALIAN	83.91±4.74	84.20±6.50	84.78±5.28	84.93±4.55	84.49±4.45
BALANCE-SCALE	78.58±4.09	78.09±3.43	79.21±3.66	79.70±4.79 ⊕	80.01±4.54 ⊕
BRIDGES	59.91±14.86	61.82±15.68 ⊕	59.00±18.20	61.73±16.57	61.73±16.57
CAR	93.81±1.37	89.76±2.00 ⊖	90.34±2.33 ⊖	93.17±1.87	94.33±1.97
DERMATOLOGY	95.36±4.04	95.90±3.29 ⊕	95.91±3.90 ⊕	96.19±3.88 ⊕	95.63±3.69
HAYES-ROTH	74.89±8.65	74.89±13.02	75.66±11.94	74.95±11.96	73.41±11.60
HEART	53.33±12.74	48.52±8.02 ⊖	45.56±5.25 ⊖	46.30±6.68 ⊖	49.63±9.83 ⊖
HEPATITS	72.96±15.07	75.00±14.08	65.38±16.94 ⊖	69.75±18.55	69.08±11.15
HORSE-COLIC	81.00±7.61	78.00±7.03 ⊖	76.00±11.91 ⊖	77.33±12.89	80.33±9.83
LABOR-NEG	67.50±22.50	67.50±22.50	67.50±22.50	67.50±22.50	67.50±22.50
LIVER-DISORDERS	45.55±8.46	48.40±9.42	48.40±9.42	46.40±8.85	48.98±9.50
SOYBEAN	85.67±6.39	86.62±5.62	85.02±5.45	86.62±5.75	86.96±4.85
TAE	43.00±15.95	45.00±14.08	45.00±14.08	43.00±15.95	43.00±15.95
ZOO	92.00±7.48	92.00±7.48	92.00±7.48	94.00±6.63 ⊕	94.00±6.63 ⊕

appropriate.

References

[1] Bay and S. D. (1999), *The UCI KDD Archive*, http://kdd.ics.uci.edu. University of California, Department of Information and Computer Science, Irvine, CA.

[2] Breiman and Leo, *Bagging predictors*, Technical Report. 421, Department of Statistics, University of California at Berkeley, 1994.

[3] Breiman and Leo, *Heuristics of Instability in Model Selection*, Technical Report. Department of Statistics, University of California at Berkeley, 1994.

[4] Dietterich and Thomas, *Machine Learning Research: Four Current Directions*, The AI Magazine. 18(4): 97-136, 1998.

[5] Gower, J. C. (1971), *A general coefficient of similarity and some of its properties*, Biometrics. 27: 857-872.

[6] H. Kargupta, B. H. Park and H. Dutta, *Orthogonal Decision Trees*, Technical Report. Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 2004.

[7] John Tobler, *Building Decision Tree Ensembles: A Comparative Analysis of Bagging, AdaBoost, and a Genetic Algorithm*, Technical Report. CS760 Shavlik.

[8] Lemuel R. Waitman, Douglas H. Fisher* and Paul H. King, *Bootstrapping Rule Induction to Achieve Rule Stability and Reduction*, Technical Report. Department of Biomedical Engineering, *Department of Electrical Engineering and Computer Science, Vanderbilt University, 2005.

[9] Quinlan and J. R., *Induction of Decision Trees*, Machine Learning. 1: 81-106, 1996.

[10] Turney and P., *Bias and the Quantification of Stability*, Machine Learning. 20: 23-33, 1995.

[11] X. Zhang Fern and Carla E. Brodley, *Boosting Lazy Decision Trees*, Technical Report. School of Electrical and Computer Engineering, Purdue University, 2003.

Table 3: Comparing Simple Majority Class and Majority Class$^+$

	Simple Majority	**Majority Class$^+$**			
Data Set	Class	**0.6**	**0.7**	**0.8**	**0.9**
AUDIOLOGY	73.64±5.43	58.69±4.60 ⊖	65.69±7.78 ⊖	73.67±5.72	76.12±6.24 ⊕
AUSTRALIAN	84.49±4.00	85.07±4.20	85.51±4.58	85.51±4.30	85.22±3.99
BALANCE-SCALE	81.45±5.47	80.81±4.81 ⊖	80.81±4.32	81.29±5.68	81.45±5.74
BRIDGES	63.64±19.13	59.82±17.00 ⊖	62.64±19.37	63.64±17.78	64.55±18.89
CAR	94.68±1.28	91.67±1.29 ⊖	93.40±1.19 ⊖	94.45±1.48	94.56±1.77
DERMATOLOGY	91.51±6.23	91.25±6.29	93.44±4.65 ⊕	93.71±5.37 ⊕	93.71±4.05 ⊕
HAYES-ROTH	70.38±10.69	73.30±12.73	74.12±11.15 ⊕	72.64±10.52 ⊕	71.15±9.71
HEART	78.15±6.92	68.52±14.37 ⊖	72.22±9.55 ⊖	73.33±10.18	74.82±8.73
HEPATITS	78.75±7.58	79.37±8.93	79.37±8.93	78.75±7.58	78.75±7.58
HORSE-COLIC	85.33±5.62	70.33±11.00 ⊖	80.33±12.60 ⊖	82.67±11.62	85.33±5.21
LABOR-NEG	67.50±22.50	67.50±22.50	67.50±22.50	67.50±22.50	67.50±22.50
LIVER-DISORDER	57.98±5.25	57.98±5.25	57.98±5.25	57.98±5.25	57.98±5.25
SOYBEAN	87.28±5.41	63.94±11.39 ⊖	78.84±7.00 ⊖	85.00±3.38 ⊖	87.60±3.87
TAE	43.67±14.49	39.67±14.64 ⊖	44.33±15.06	44.33±15.06	43.67±14.49
ZOO	91.00±11.36	40.64±15.20 ⊖	84.09±13.61 ⊖	92.00±6.00	92.00±6.00

ISBN: 978-988-98671-8-8