2007-10-25

Heuristic Weighted Voting

Kristine Perry Monteith
Brigham Young University - Provo

Heuristic Weighted Voting

by
Kristine Perry

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Masters of Science

Department of Computer Science
Brigham Young University
December 2007

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of the thesis submitted by
Kristine Perry

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

______________________________ ______________________________
Date Dr. Tony Martinez, Chair

______________________________ ______________________________
Date Dr. Christophe Giraud-Carrier

______________________________ ______________________________
Date Dr. Michael Jones

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Kristine Perry in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

______________________________ ______________________________
Date Dr. Tony Martinez
Chair, Graduate Committee

Accepted for the Department

____
Dr. Parris Egbert
Graduate Coordinator

Accepted for the College

____
Dr. Thomas Sederberg
Associate Dean – College of Physical
and Mathematical Sciences

ABSTRACT

Heuristic Weighted Voting

Kristine Perry
Department of Computer Science
Master of Science

Selecting an effective method for combining the votes of classifiers in an ensemble can have a significant impact on the overall classification accuracy an ensemble is able to achieve. With some methods, the ensemble cannot even achieve as high a classification accuracy as the most accurate individual classifying component. To address this issue, we present the strategy of Heuristic Weighted Voting, a technique that uses heuristics to determine the confidence that a classifier has in its predictions on an instance by instance basis. Using these heuristics to weight the votes in an ensemble results in an overall average increase in classification accuracy over when compared to the most accurate classifier in the ensemble. When considering performance over 18 data sets, Heuristic Weighted Voting compares favorably both in terms of average classification accuracy and algorithm-by-algorithm comparisons in accuracy when evaluated against three baseline ensemble creation strategies as well as the methods of stacking and arbitration.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Tony Martinez, for both his ideas and support in this research and his patience in helping me complete it. I am grateful to the other faculty members, and particularly the members of my committee, for their assistance and encouragement.

I am especially thankful to my parents and siblings for their love and support in the form of countless phone calls, emails, and prayers. They were of invaluable assistance in helping me finish this research. The completed project is a result of a number of miracles, so the prayers undoubtedly helped.

I would like to thank Heather Hogue and Emily Stout for their assistance in proofreading and editing, as well as their encouragement.

I would also like to thank Adam Monteith for his love, assistance and patience. I am grateful to him for taking me out to dinner when I was too distracted to eat, for advice in actually completing the project, and for extreme patience in listening to rehearsals of research presentations. I think he may be the only person who wanted this thesis to be finished more than I did.

In addition, I am grateful for the funding I received from the National Science Foundation in support of my graduate studies.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Much attention has been directed towards finding an optimal set of classifiers for use in an ensemble. One approach is to train classifiers on different portions of a data set in order to produce a diverse ensemble, the strategy employed in Boosting [FS96, SS98]. Another strategy is to generate diversity not from the training sets but from the classifiers themselves. A number of techniques have been proposed to measure the diversity between classifiers [KW96, PM05], and researchers have discussed correlations between classifier diversity in an ensemble and the accuracy that the ensemble is able to achieve [KW03, RG01]. Efforts have been directed towards developing search strategies to dynamically discover a set of classifiers for use with a given task [RG05, GR00, KS97, Lam00]. Some research has even focused on discovering which classifiers and sets of classifiers are most accurate over the set of all "interesting" classification tasks. In a large scale empirical study, Caruana [CN06] looks at the behavior of a number of individual classifiers and ensembles on tens of thousands of data sets in order to determine which classifiers or combinations demonstrate the overall best performance.

Even with an optimal set of classifiers, however, there remains the question of how to combine the information provided by these individual classifiers. Ideally, component classifiers specialize on different areas of a given data set, and the effectiveness of an ensemble can be enhanced if a way is found to combine votes in a way that allows the overall ensemble to leverage these areas of expertise. Rather than attempting to select an ideal set of classifiers, this work will focus on optimizing the method of combining the votes of these classifiers to increase the overall accuracy of the ensemble.

The simplest method of combining the information presented in an ensemble is to allow each learner to have one vote toward the overall classification of an instance. A number of ensemble techniques such as the traditional method of Bagging [Bre96] employ this strategy. With Boosting [FS96, SS98], votes are weighted by the accuracy a given learner can achieve on the data set. More complicated ensemble-combining strategies include Gating [JJN91], which allows only highly confident classifiers in the ensemble to vote, and Stacking [Wol92] which makes use of a meta-level learning algorithm that discovers the best way to combine outputs from the base level classifiers. Arbitration [OKA01] creates a "referee" to determine the confidence that a learner has in its classification of the various subdomains of a given problem. Information about the misclassification of points and information about the learners themselves are used in the development of the meta-learner referees.

These ensemble construction methods highlight the fact that individual learners perform better on some portions of a given data set than others, and higher ensemble accuracy can often be obtained by taking this into account. For example, Delegating [FFH04] is an approach where a learner assigns a class label to a given point only if it has high confidence in that particular class. If it is less confident, the point is delegated to another learner. With a technique called Dynamic Selection [Mer95], information is collected on how well learners perform on points in the training set. These learners are then used to classify test set examples, and their collective predictions are used to determine similarity to different points in the training set. The learner that achieves the best performance on that area of the training set is then used to classify the test set example.

Dzeroski and Zenko [DZ04] found that the accuracy of an ensemble over a data set is often less than the accuracy of one of the classifiers contributing to the ensemble. In order to justify the overhead of creating an ensemble, the ensemble should meet the criterion of having a higher overall classification accuracy than any of its component classifiers. In the algorithms Dzeroski and Zenko explore, only their modified stacking strategy was able to consistently achieve this level of accuracy. Arbitration is also shown to be superior to the strategy of selecting the best classifier.

This work presents the technique of *Heuristic Weighted Voting.* This strategy uses a number of different heuristics that estimate the confidence that a given classifier has in its classification of a given instance. The confidence metrics are then combined to produce an overall value with which to weight the classification. The algorithm functions in a similar manner to strategies such as Arbitration in that the weighting of the votes of an ensemble are determined on an instance-by-instance basis. However, it avoids the extra overhead of creating a meta-learner to combine the votes. It also has the advantage of more explicitly taking into account a wide number of factors that contribute to confidence in individual instance classification.

Heuristic Weighted Voting is shown to achieve higher average classification accuracy over 18 data sets than the standard combination strategies employed by Bagging and Boosting as well as the SelectBest strategy of allowing the most accurate classifier in the ensemble make all the classifications. It also achieves higher average classification accuracy than the stacking algorithms presented by Dzeroski and Zenko [DZ04]. Heuristic Weighted Voting outperforms these four strategies in an algorithm-by-algorithm comparison by wins and losses in accuracy on the individual 18 data sets.

Heuristic Weighted Voting can be used in conjunction with Arbitration to increase referee accuracy. It also functions on a competitive level with this strategy both in terms of average classification accuracy and win/loss ratios.

Section two of this work gives an overview of the Heuristic Weighted Voting algorithm. Section three presents options for heuristics that can be used with five common classification algorithms. Section four provides results comparing Heuristic Weighted Voting with standard voting, voting by accuracy, the SelectBest strategy, and a stacking algorithm. A discussion of Heuristic Weighted Voting and Arbitration is given in section five. Section six outlines conclusions and suggests options for further research.

2. HEURISTIC WEIGHTED VOTING

For these experiments, n component classifiers $C_1 \ldots C_n$ are constructed using elements from a data set D. Each member of the set is a vector of attributes $\boldsymbol{d_i}$ and its corresponding class label y_i, where y_i is an element of Y, a discrete set of possible labels. Then an unlabeled attribute vector $\boldsymbol{x_i}$ is classified by each classifier C_j, with $y_{i,j}$ being the class label assigned to $\boldsymbol{x_i}$ by C_j. A vector of heuristics $\boldsymbol{h_{i,j}}$ is calculated. Each element in $\boldsymbol{h_{i,j}}$ represents a different way to predict the confidence of classifier C_j in its assignment of the class label $y_{i,j}$ to $\boldsymbol{x_i}$. In this manner, the issue of confidence is addressed from a number of different perspectives, with the expectation that this will produce a more accurate representation of confidence. The elements in $\boldsymbol{h_{i,j}}$ are then combined to produce a single confidence value $w_{i,j}$ which is used to weight $y_{i,j}$. The class label assigned to $\boldsymbol{x_i}$ is calculated by summing the weights for each possible label and selecting the class label with the maximum total.

Averaging is used as the combination strategy in this work. More sophisticated strategies are certainly possible; this averaging strategy gives disproportionate weight to highly correlated heuristics and does not take full advantage of unique relationships between specific heuristics. However, the results show that even a simple averaging strategy allows the algorithm to achieve high classification accuracy.

```
1. Train each of $n$ classifiers $C_1 \ldots C_n$ using training set $D$
2. For an unlabeled attribute vector $\boldsymbol{x_i}$
   a. For each classifier $C_j$
      1. Determine $y_{i,j}$ for the class value of $x_i$ as predicted by $C_j$
      2. Create vector $\boldsymbol{h_{i,j}}$ of confidence measures using heuristics specific to $C_j$
      3. Calculate $w_{i,j}$ by combining the elements in $\boldsymbol{h_{i,j}}$
   b. Class label for $\boldsymbol{x_i}$ = $\arg\max_{y \in Y}\left(\sum_{1}^{j} y_{i,j} w_{i,j}\right)$
```

Figure 2.1 Heuristic Weighted Voting

The experiments in this work use five different types of classifiers. Each classifier is trained using the same training set data. Then each instance in the test set is assigned a class value and an overall confidence rating for that classification by each of the five classifiers. The confidence rating is calculated differently for each type of classifier. For example, six different heuristics are used to calculate confidence in the prediction of a decision tree classifier. A given instance would receive six confidence ratings, reflecting properties such as the purity of the leaf node in which it was classified and the number of instances classified at that leaf. These six numbers are then averaged together to produce an overall confidence rating for the decision tree's classification of this particular instance. A similar method is used to calculate an overall confidence rating for each of the classifiers. Class label predictions are then weighted by these overall confidence ratings, and the ensemble selects the maximum predicted class label as the one to assign to this particular instance.

.3. PRESENTATION OF HEURISTICS

This section contains the information about the heuristics used to predict confidence in predictions for each of five different algorithms. The five algorithms used in this work were selected because they are representative of standard classes of models in machine learning. Many of the heuristics presented here could be adapted for use with similar machine learning algorithms. While we have tried to select diverse models to represent the spectrum of machine learning algorithms, the technique of Heuristic Weighted Voting could be applied to ensembles with any number and type of base-level classifiers. The ensembles constructed using the five base-level classifiers discussed here are designed simply to present the concept.

The algorithms used in this work are implemented using Weka open source code [WF05]. Efficacy of the various heuristics is evaluated using eighteen data sets, taken from the UCI Repository [HBM98]. Table 3.1 provides information about these data sets.

Half of the data sets have real-valued attributes, and half have attributes with discrete values. Data sets were selected so as to achieve variety in number of instances, attributes, and output classes. For the vote and zoo data sets, the two discrete-valued data sets with unknown values, the majority value for a given attribute was used in place of any unknown values. With the real-valued cancer data set, unknown values were replaced with the average value for the attribute.

Each subsection contains information about the algorithm to be addressed and the heuristics specific to that algorithm. A brief discussion of how these heuristics operate on the data set used in this work is also included. The subsections also contain graphs providing information about the behavior of each heuristic on the data sets shown in Table 3.1. Each of the classifiers was evaluated over each data sets using ten-fold cross validation. Instances were then marked as correctly or incorrectly classified based on the classifier's ability to classify the instance when it appeared in the test set.

Data Set	Instances	Attributes	Output Classes
audiology	200	69	24
bupa	286	9	2
cancer	345	6	2
car evaluation	1728	6	4
cmc	1473	10	3
diabetes	768	8	2
ecoli	336	7	8
glass	214	9	7
haberman	306	3	2
hayes	132	4	3
heart statlog	270	13	2
iris	150	4	3
monks	432	6	2
postOp	90	8	3
sonar	209	60	2
tic-tac-toe	958	9	3
vote	435	16	2
zoo	101	16	7

Table 3.1 Information for Data Sets

As an example, Figure 3.0.1 shows a graph constructed for the heuristic measuring purity of classification, or the percentage of instances with the majority classification, at a leaf node of a decision tree. The graph shows the number of instances receiving a given confidence measure that were correctly and incorrectly classified. While the purity heuristic provides real values, for clarity in graphing, confidences provided in this section are shown by placing values in discrete bins. For example, confidence values from 0.5 to 0.59 are all graphed as 0.5. The more precise confidence measurements are used in the actual Heuristic Weighted Voting experiments. The lighter bar on the left for each bin represents the number of instances receiving this confidence value that were correctly classified. The darker bar on the right represents the number of instances that were incorrectly classified. For example, the far right-hand bin in Figure 3.0.1 shows that, out of all 18 data sets, that there were 3384 correctly classified instances that received a confidence value of 1.0 from this heuristic, and there were 304 instances receiving this confidence value that were incorrectly classified.

Figure 3.0.2 presents a different way of looking at the behavior of this heuristic on the instances in the data sets. It plots the percentage of instances receiving a given confidence measure that were correctly classified. For example, the column on the far right hand side indicates that the 3384 correctly classified instances receiving a confidence value of 1.0 represent 92% of all the instances given this value by this heuristic. In other words, 92% of test set instances classified in leaf nodes with complete purity of classification were assigned the correct classification by the decision tree. The column labeled 0.3 indicates that only 33% of the instances given a confidence value between 0.3 and 0.39 were correctly classified. These would be instances in data sets with a fairly high number of class values, since the 0.3 confidence value would indicate that only a third of the instances in the leaf node in which a given instance was classified had the majority classification for that particular leaf node. As Figure 3.0.1 shows, there were only a negligible number of instances that received these confidence values from this heuristic.

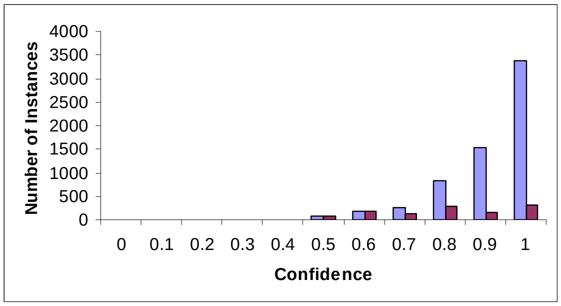

Figure 3.0.1 Number Correct/Incorrect

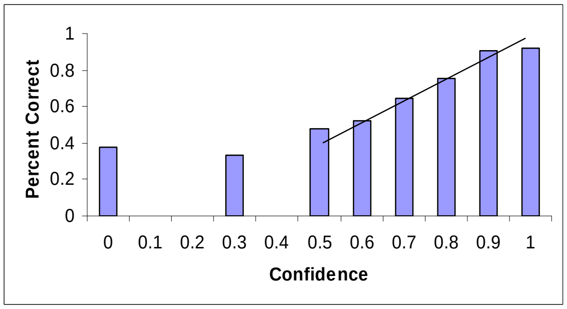

Figure 3.0.2 Percent Correctly Classified

The information from both graphs is then combined into one. Trend lines are calculated using graphs similar to Figure 3.0.2, and these trend lines are superimposed on graphs like the one shown in Figure 3.0.1. Trend lines are only shown over the portion of the graph with a substantial number of instances.

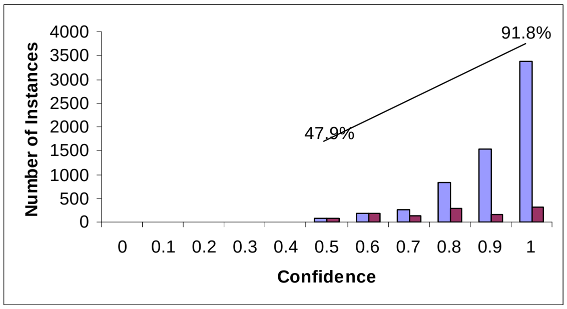

Figure 3.0.3 Number Correct/Incorrect with Trend Line

With an ideal heuristic, the confidence values would be identical to the percentage of instances receiving that value that were correctly classified. One way of estimating the effectiveness of a given heuristic is to observe how close its trend line comes to this ideal. With the purity heuristic here, over 91% of instances receiving a confidence rating of 1.0 are correctly classified, while less than half of those instances receiving a confidence rating between 0.5 and 0.59 are correctly classified. The heuristic is able to make a rough estimate of the likelihood of an instance to be correctly classified. While not all the heuristics presented here estimate likelihood of correct classification as effectively, they all demonstrate the tendency to assign higher confidence ratings to correctly classified instances.

3.1 Decision Tree – J48

The J48 algorithm is the Weka implementation of the C4.5 algorithm [Qui93], an extension of the ID3 decision tree [Qui86]. Six different heuristics are used to predict confidence in this algorithm's classification of a given instance:

1. The purity of classification at this node (the percentage of instances with the majority classification at the leaf node where the given instance was classified).
2. The percentage of instances in the training set that were classified at this leaf node.
3. The level of the tree at which the given instance was classified (calculated by subtracting the level of this node from the maximum number of levels in the tree and normalizing by the maximum number of levels).
4. The average of the information gain statistics along the classification path (normalized by the maximum possible information gain for a given data set).
5. The percentage of instances at this leaf node that were correctly classified in hold-one-out cross validation experiments.
6. The percentage of instances at this leaf node with the majority classification for the node that were correctly classified in hold-one-out cross validation.

The first heuristic is a standard method for predicting confidence in the classification of a decision tree [WF05]. The second and third provide an effective complement to the first by providing information about the amount of overfit and thus how much the first should be trusted. The fourth heuristic provides information about how effectively a given attribute is able to split the data at each level of the decision tree, assuming that strong attributes will lead to more confident classifications. The fifth identifies how effective the classifier is at classifying the instances in this particular section of the data. The sixth heuristic provides information about how effectively the classifier was able to classify the instances specifically contributing to the classification of the given instance.

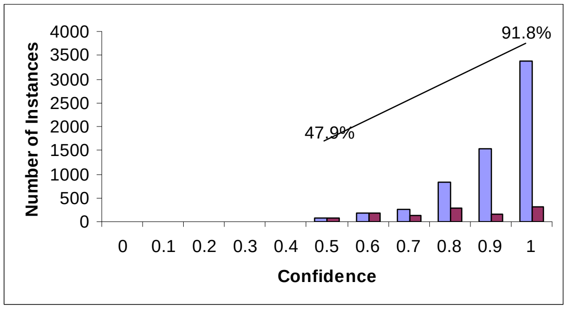

Figure 3.1.1 Purity of Classification

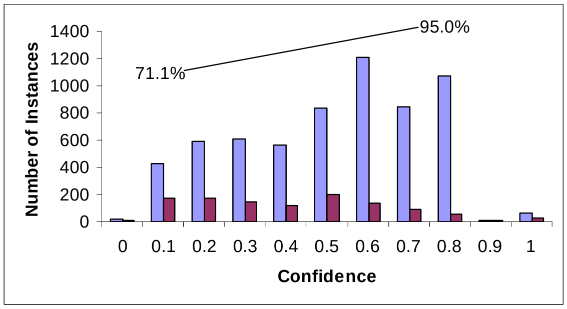

Figure 3.1.2 Instances at Leaf Node

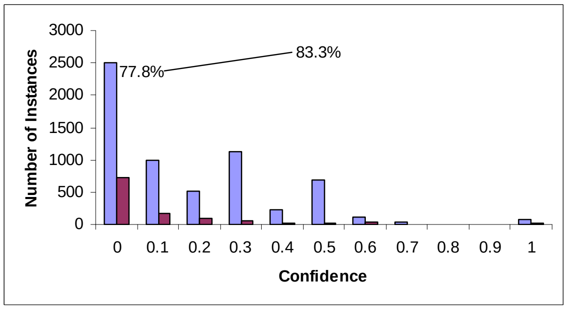

Figure 3.1.3 Level of Leaf Node

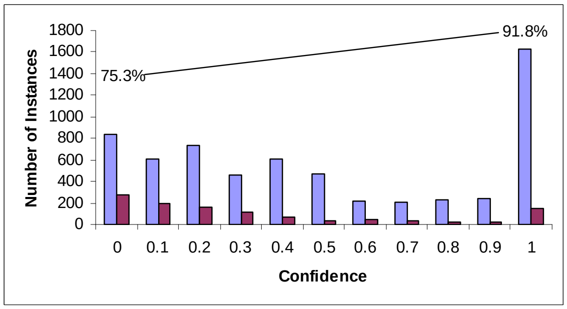

Figure 3.1.4 Information Gain along Path

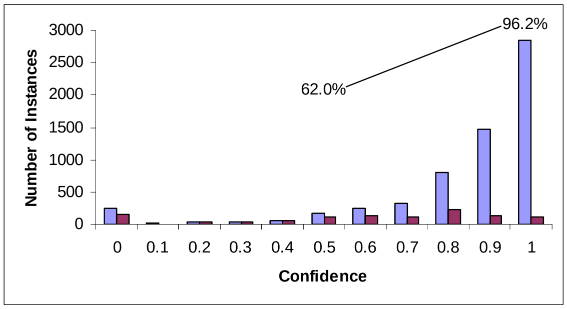

Figure 3.1.5 Correctly Classified Instances

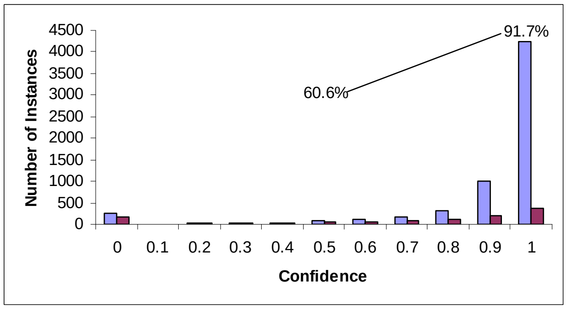

Figure 3.1.6 Correctly Classified Voters

One can infer from these graphs that the purity of classification at a leaf node and the number of instances correctly classified at that leaf node appear to be better predictors of correct classification for test instances classified at that leaf node than the other heuristics studied. However, the other heuristics do provide additional information that may be useful in determining confidence, particularly when taken into consideration with the more accurate confidence-predicting heuristics. For example, with most of the data sets, a majority of the correctly classified instances were given a confidence rating of 1.0 by the purity heuristic. But with the haberman data set, a majority of the correctly classified instances were assigned a confidence rating of 0.82 by this same heuristic. The few instances receiving higher confidence ratings were all misclassified. These misclassified instances were generally found in leaf nodes that contained only a few instances, so they received lower confidence ratings both from the heuristic that measured the percentage of instances at the leaf node and the one that measured the level of the tree.

.3.2 Rule-Based Classifier – Decision Table

These experiments use one of Weka's rule-based classifiers called a Decision Table [Koh95]. This algorithm selects a set of attributes to be used in determining classification, and produces a classification for each combination of observed values for these attributes. The following heuristics are used to predict confidence in this algorithm's classification of a given instance:

1. The number of instances with the majority classification covered by the rule that applies to the instance.

2. The number of antecedents in this rule (calculated by subtracting the number of antecedents by the maximum number of antecedents in a rule created for this data set and normalizing by the maximum number of antecedents).

3. The percentage of instances in the training set covered by this rule.

4. The percentage of instances covered by this rule that were correctly classified in hold-one-out cross validation experiments.

5. The percentage of instances covered by this rule with the majority classification for the rule that were correctly classified in hold-one-out cross validation experiments.

6. Whether or not the instance is covered by a rule.

The rationale for these heuristics is similar to the rationale given for the heuristics presented for the decision tree. The first heuristic is a standard measure of confidence.

The second and third assess the likelihood of overfit or underfit. The fourth and fifth measure the effectiveness and strength of classification. They indicate how effectively the decision tree was able to classify instances that would end up in this region and, more specifically, how effectively the most pertinent instances in this region can be classified. The sixth heuristic indicates whether or not a rule was found in the table that covered the given instance to be classified.

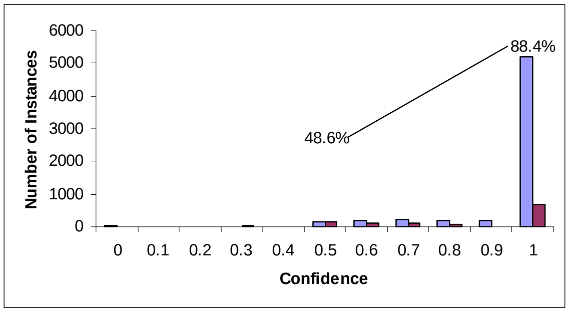

Figure 3.2.1 Purity of Classification

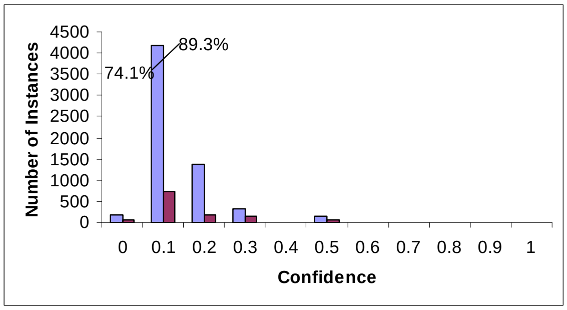

Figure 3.2.2 Number of Antecedents

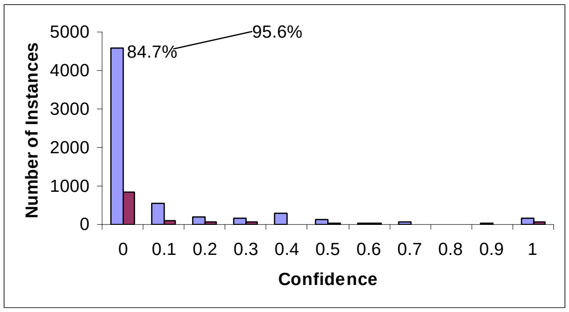

Figure 3.2.3 Number of Instances Covered

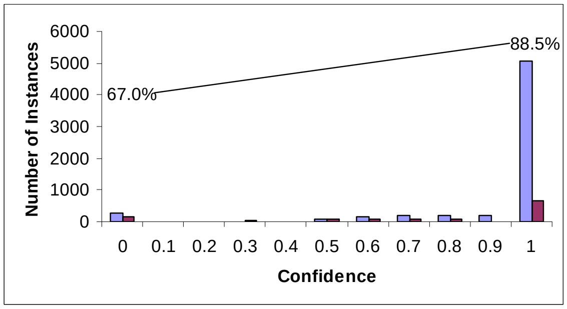

Figure 3.2.4 Correctly Classified Instances

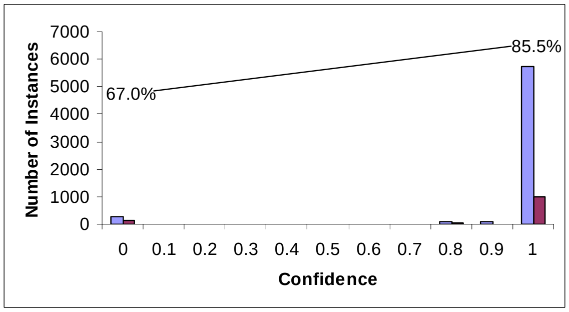

Figure 3.2.5 Correctly Classified Voters

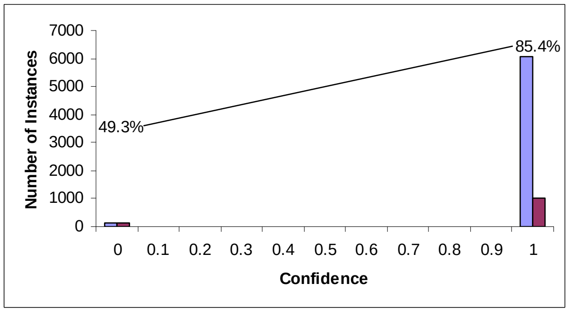

Figure 3.2.6 Instance is Covered by Rule

For most of the heuristics, a majority of the correctly classified instances received the highest classification rating. The heuristic reporting the number of antecedents appears to be the least effective at predicting confidence on its own, but like the second heuristic for the decision tree, it functions as a complement to the first. In many cases, when an incorrectly classified instance received too high a rating from the purity heuristic because it was classified by too specific a rule, the heuristic identifying the number of antecedents and the one identifying number of instances covered would balance it with a lower confidence rating. The sixth heuristic, representing whether or not an instance was covered by a rule, gives a clear example of how the heuristics are more likely to assign higher confidence ratings to correctly classified instances. If an instance received the higher confidence rating from this heuristic, there was an 85.4% probability that it was correctly classified. If it received the lower confidence rating, there was only a 49.3% chance that it was classified correctly.

3.3 Instance-Based Classifier

With the instance-based *k*-nearest-neighbor algorithm, an instance is classified based on the classifications of the *k* instances nearest that instance [CH67]. These experiments use the five-nearest-neighbor version of the algorithm. Six different options are used to predict confidence in this algorithm's classification of a given instance:

1. The percentage of the first five neighbors that had the same classification as the majority classification for those five neighbors.

2. The difference between the distance-weighted vote of the predicted class and the distance-weighted vote of the next highest class.

3. The average distance from this instance to its first five neighbors (normalized and subtracted from one).

4. The percentage of the first five neighbors that were correctly classified in hold-one-out cross validation.

5. The percentage of neighbors with the majority classification that were correctly classified in hold-one-out cross validation.

6. Consistency of classification with 3-NN, 5-NN and 7-NN variants of the instance-based algorithm.

The first and second heuristics indicate the general confidence in a classification, and how confident that classification is relative to other possible classifications. The third measures how close the neighbors are to the individual instance, using the assumption that a point closer to other points is more likely to be correctly classified. The fourth and fifth heuristics measure the accuracy of classification of instances in this region and the accuracy on instances contributing to the classification of the instance in question. The last heuristic indicates the effectiveness of using this particular number of neighbors to classify the given instance. The following figures demonstrate the overall effectiveness of these heuristics:

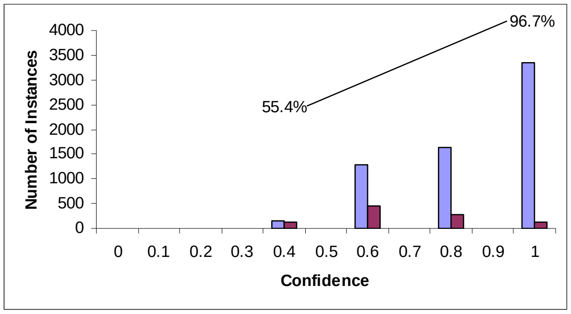

Figure 3.3.1 Neighbors in Agreement

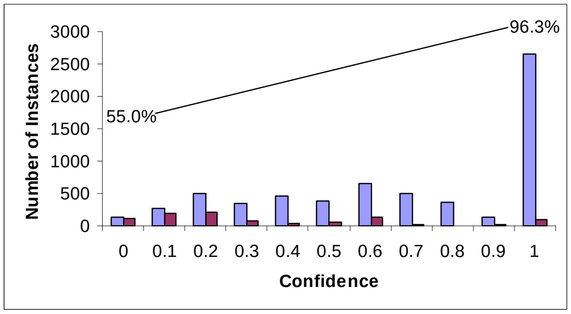

Figure 3.3.2 Highest Minus Second

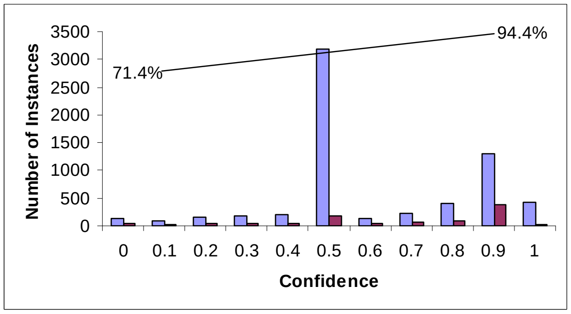

Figure 3.3.3 Average Distance to Neighbors

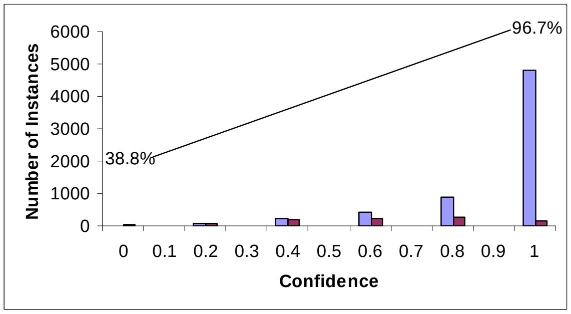

Figure 3.3.4 Correctly Classified Neighbors

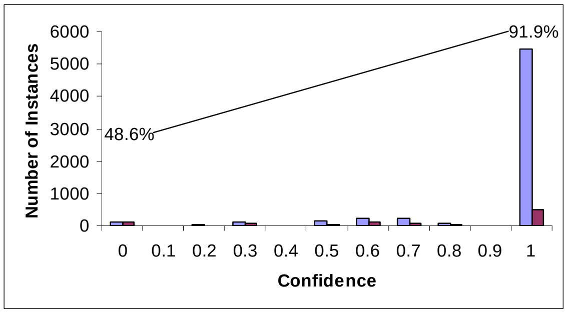

Figure 3.3.5 Correctly Classified Voters

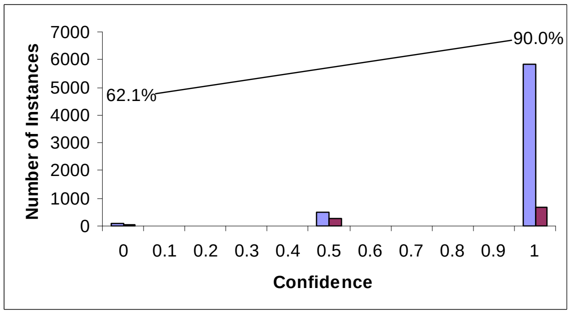

Figure 3.3.6 3-NN vs. 5-NN vs. 7-NN

All of the heuristics are effective in assigning a high confidence rating to a large percentage of the correctly classified instances and, in general, instances with higher confidence ratings were much more likely to be correctly classified than incorrectly classified. Due to the fact that only five neighbors were considered by this classifier, the heuristics identifying number of neighbors in agreement and number of correctly classified neighbors only assigned one of five possible values. The heuristic measuring consistency of classification only assigned three possible values. An instance received a confidence rating of 1.0 if both the 3-NN and 7-NN classifiers agreed with the classification of the 5-NN classifier. It received a confidence rating of 0.5 if only one of these classifiers agreed and a rating of 0.0 if neither agreed with the 5-NN classification.

.3.4 Naïve Bayes Classifier

The Naïve Bayes classifier uses Bayesian logic to predict class values for each instance based on the probabilities of the attribute values for that instance [Lan95, Mit97]. The following are used to predict confidence in classifications for the Naïve Bayes classifier:

1. Probability predicted by the Naïve Bayes classifier.
2. The difference between this probability and the next highest probability predictor.
3. The distance between the highest probability and the remaining probabilities when the first and second highest were excluded.

4. The average probability of each attribute value in the instance.

5. The percentage of the nearest five neighbors with regards to the probability spectrum that were correctly classified in hold-one-out cross validation.

The first heuristic was used because it is the standard way of predicting confidence of a Naïve Bayes classifier. However, McCallum and Niggam [MN98] found that the power of the Naïve Bayes classifier lies not in the actual probability predicted, but in the ordering of the probabilities. The second and third heuristics are attempts to gain more information about how confident the classifier is in its ordering. The fourth heuristic addresses the fact that attribute values with lower representation in a data set may be less effective at contributing to a correct classification. The last heuristic is aimed at determining how confident the classifier is in this region of the data set. With this heuristic, the output probabilities of all the instances in the training data are taken into consideration. The five instances with output probabilities closest to the output probabilities of the instance in question are then located, and the heuristic is calculated by observing what percentage of these five instances are correctly classified in hold-one-out cross validation on the training set.

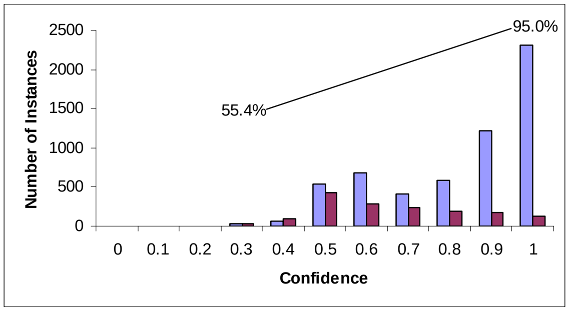

Figure 3.4.1 Probability of Class Value

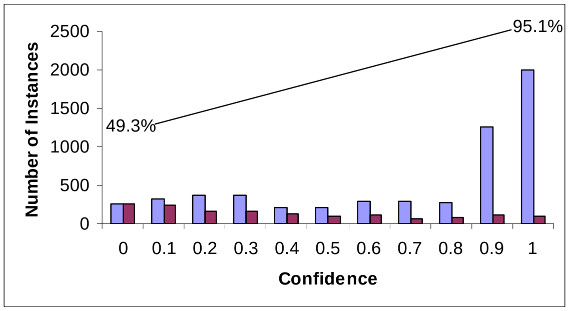

Figure 3.4.2 Highest Minus Second

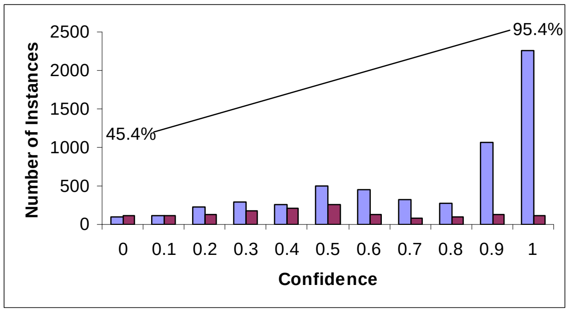

Figure 3.4.3 Highest Minus Remaining

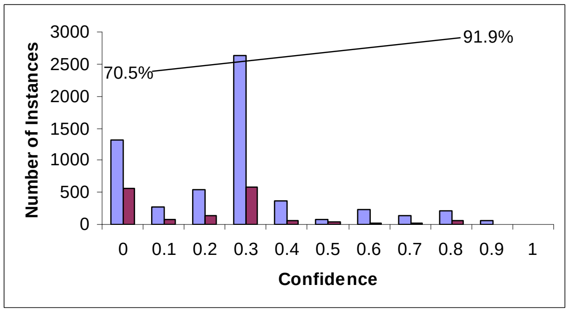

Figure 3.4.4 Value Probability Averages

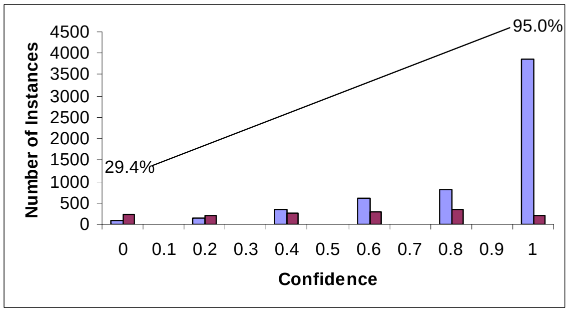

Figure 3.4.5 Correctly Classified Neighbors

Most of the heuristics behave as expected. The heuristic measuring the average probabilities of the attribute values does behave somewhat differently. On some data sets, it was able to function as a good predictor of confidence in its own right. For example, with the zoo data set, a majority of the correctly classified instances received high confidence ratings from this heuristic, and all the incorrectly classified instances received confidence ratings of less than 0.5. On other data sets, this heuristic provides a good complement to some of the other heuristics. With the haberman data set, most of the incorrectly classified instances receiving high confidence ratings from the first heuristic were given lower confidence ratings by this fourth heuristic.

3.5 Multilayer Perceptron trained with Backpropagation

As one of the most common methods of training a multilayer perceptron, backpropagation incrementally changes the weights between nodes when these weights are responsible for the misclassification of points during training [RHW86]. These experiments use a multilayer perceptron with a single hidden layer. The following heuristics are used to predict confidence in classification by the Multilayer Perceptron:

1. The activation output for the selected classification.
2. The difference between the highest activation output and the second highest activation output.
3. The percentage of the nearest five neighbors with regards to the activation output spectrum that were correctly classified in hold-one-out cross validation.
4. The percentage of the nearest five neighbors with regards to the activation output spectrum of the hidden layer that were correctly classified in hold-one-out cross validation.
5. The average distance to the five closest neighbors compared to the average of this statistic computed for all instances.
6. The average distance to the five closest neighbors based on hidden-layer activation values compared to the average of this statistic computed for all instances.

The first and second heuristics provide information about the confidence of a given classification, and confidence relative to other possible classifications. The third and fourth provide information about how confident the learner is on this section of the data set. These heuristics are calculated in a similar manner to the fifth heuristic used for the Naïve Bayes algorithm. All the instances in the training set are considered. The five with output spectra most similar to the instance in question are then used to calculate the heuristic. With the third heuristic, the outputs from the standard output nodes are used when calculating the nearest neighbors. With the fourth heuristic, the outputs from the hidden nodes are used. The fifth and sixth heuristics provide information about how close a given instance is to previously seen instances. The following figures demonstrate the overall effectiveness of these heuristics:

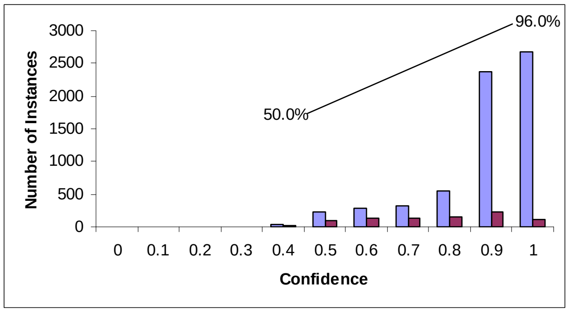

Figure 3.5.1 Activation Output

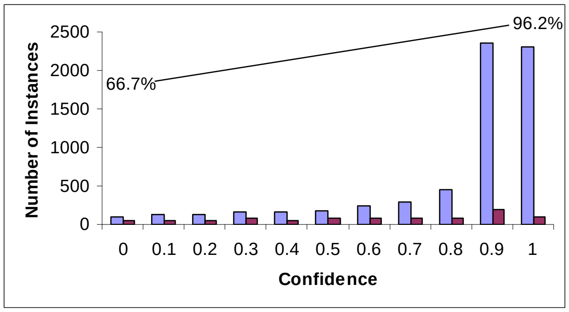

Figure 3.5.2 Highest Minus Second

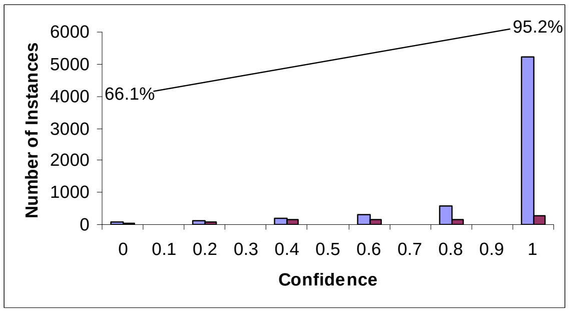

Figure 3.5.3 Correctly Classified Neighbors

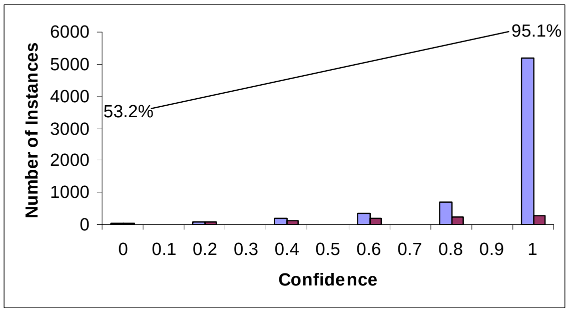

Figure 3.5.4 Correctly Classified Neighbors (Hidden Layer)

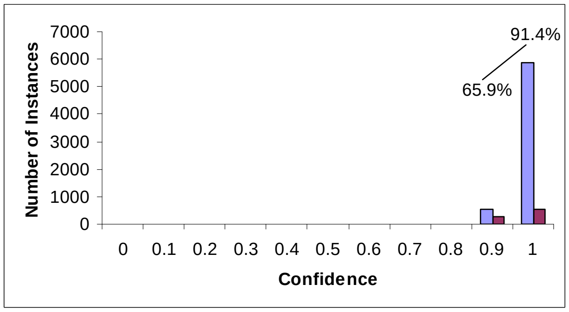

Figure 3.5.5 Average Distance to Neighbors

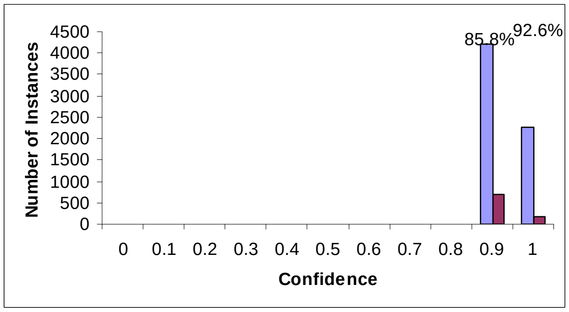

Figure 3.5.6 Average Distance to Neighbors (Hidden Layer)

As illustrated by the above figures, all of these heuristics tend to assign a 1.0 confidence rating to a large number of correctly classified instances. The number of correctly classified instances at each confidence rating tends to taper off as the ratings become lower. On average, the heuristics for this classifier were more highly correlated with each other than the heuristics for other classifiers. However, an examination of the confidence ratings assigned to individual instances in the data sets shows that there is enough variation that each heuristic does provide some extra information to a classifier.

4. RESULTS AND DISCUSSION

In this section, Heuristic Weighted Voting is compared with a number of different ensemble combining strategies. Using a combination of multiple heuristics is shown to be more effective than the strategy of weighting by single heuristics or pairs of heuristics. A comparison of Heuristic Weighted Voting to Arbitration [OKA01] is provided in section five.

4.1 Results

Heuristic Weighted Voting is compared with three different baseline methods. The first is a standard voting method where each classifier in an ensemble votes on the classification of an instance and the votes are weighted equally. The second baseline method weights the votes by the overall accuracy of the learner. The third baseline method, identified here as the SelectBest method, chooses the classifier in the ensemble that achieved the highest accuracy on the training data and uses that classifier alone on the test data. Heuristic Weighted Voting is also compared to the method of stacking found to be most effective by Dzeroski and Zenko [DZ04]. In this method, identified as Modified Stacking in the following analyses, the output probabilities of each of the component classifiers are given as input to a set of model trees. Each tree is designed to make a binary decision about a given possible output class, and the ensemble assigns a value to the instance according to which model tree has the highest positive confidence in its prediction. Table 4.1 shows the results of these comparisons:

Data Set	Standard Voting	Weight by Accuracy	SelectBest	Modified Stacking	Heuristic Weighted Voting
Audiology	78.761	78.761	76.991	75.221	78.319
bupa	68.986	69.275	67.246	57.101	71.304
cancer	96.567	96.567	96.996	97.425	96.567
car	96.296	96.296	98.727	99.132	96.644
cmc	53.7	52.885	52.614	50.441	53.021
diabetes	76.563	76.563	74.349	71.224	76.563
ecoli-c	86.31	87.202	86.607	84.821	87.798
glass	71.495	73.364	70.093	71.495	72.43
haberman	74.183	74.183	74.837	71.895	74.183
hayes	71.97	72.727	81.061	83.333	74.242
heart-statlog	83.704	83.704	81.481	79.259	83.333
iris	95.333	96	90	95.333	96
monks	99.769	99.769	100	100	99.769
postop	70	70	70	71.111	71.111
sonar	82.692	82.692	83.654	86.058	83.654
tic-tac-toe	93.319	93.319	98.956	99.791	98.434
vote	95.879	95.879	96.095	97.18	95.662
zoo	95.05	95.05	92.079	93.069	96.04
Average:	82.81	83.01	82.88	82.44	83.62

Table 4.1 Comparison of Heuristic Weighted Voting with Other Strategies using Average Accuracy

Heuristic Weighted Voting achieves the highest average accuracy. Statistical significance in the differences between the average accuracies of the various algorithms is calculated using the technique proposed by Menke and Martinez [MM04]. This technique was shown to provide more accurate p-values than the traditional Student's T-test. At an alpha level of 0.05, Heuristic Weighted Voting was the only algorithm to achieve significantly higher accuracy than the basic strategy of standard voting.

Heuristic Weighted Voting also performs well in an algorithm-by-algorithm comparison of accuracy on the eighteen individual data sets. Table 4.2 shows an algorithm-by-algorithm comparison of each of the five strategies. Each box shows the number of wins, losses, and ties in accuracy on each of the 18 data sets when comparing the algorithm in the given row with the algorithm in the given column. For example, when Modified Stacking was compared to a standard voting strategy, it achieved a higher classification accuracy on eight of the data sets, a lower classification accuracy on eight of the data sets, and the same classification accuracy on two data sets.

	Standard Voting	Weight by Accuracy	SelectBest	Modified Stacking
Weight by Accuracy	4/1/13			
SelectBest	9/8/1	8/9/1		
Modified Stacking	8/8/2	8/10/0	11/6/1	
HWV	9/5/4	8/5/5	10/7/1	10/7/1

Table 4.2 Comparison of Heuristic Weighted Voting with Other Strategies using Win/Loss/Tie Ratios

Table 4.2 demonstrates that Heuristic Weighted Voting achieves a higher classification accuracy on a majority of the data sets studied when compared to Standard Voting, Weighting by Accuracy, the SelectBest method, and Modified Stacking.

In order to further motivate the need for multiple heuristics, the accuracies of different Heuristic Weighted Voting ensembles created with single heuristics and subsets of heuristics are tested. Four different options are given for selecting single heuristics or subsets of heuristics for each of the different classifiers.

The first alternate ensemble is created by using heuristics traditionally used in predicting confidence in a classification. An intuitive heuristic is used for classifiers that do not traditionally output a confidence. The following are the traditional or intuitive heuristics used in this ensemble:

- Decision Tree: Purity of Classification
- Rule-Based Classifier: Purity of Classification
- Instance-Based Classifier: Percentage of Neighbors in Agreement
- Naïve Bayes Classifier: Probability of Class Value
- Multilayer Perceptron: Activation Output

The next ensemble is also constructed using single heuristics to predict confidence. But in this case, an attempt is made to select more effective heuristics. To this end, one ensemble for each heuristic was constructed using five of the same type of classifier. Each classifier was trained on different portions of the available training data to ensure variability, and the heuristic in question was used to weight the votes of the classifiers. For the decision tree and the multilayer perceptron, the selected heuristics are the same as in the first ensemble, but different heuristics are used for the other three types of classifiers. The following are the heuristics that resulted in the highest non-hybrid ensemble accuracy for each classifier:

- Decision Tree: Purity of Classification
- Rule-Based Classifier: Instances Covered by Rule
- Instance-Based Classifier: Highest Minus Second Highest
- Naïve Bayes Classifier: Percent of Correct Neighbors
- Multilayer Perceptron: Activation Output

As another method of selecting the most effective heuristics, decision trees were constructed to predict whether or not an instance was correctly classified using the confidence values produced for each classifier as inputs. The following were the most commonly occurring attributes in such decision trees:

- Decision Tree: Purity of Classification
- Rule-Based Classifier: Percent Voters Correctly Classified
- Instance-Based Classifier: Percent Correctly Classified
- Naïve Bayes Classifier: Highest Probability
- Multilayer Perceptron: Percent Correctly Classified (Hidden Layer)

Several heuristics were commonly paired in such trees, so ensembles were also created with these commonly paired attributes. The following are the heuristics used to predict confidence in this set of ensembles:

- Decision Tree: Purity and Percent Correctly Classified
- Rule-Based Classifier: Number of Nodes and Number of Antecedents

- Instance-Based Classifier: Percent Correct and First Minus Second
- Naïve Bayes Classifier: Significance and Correct Neighbors
- Multilayer Perceptron: Correct Neighbors (Hidden) and Average Distance

All these techniques are then compared to a strategy where all the heuristics for a given learner were averaged to produce an overall confidence heuristic for each learner on each data point. The resulting predictive accuracies, shown in Table 4.3, demonstrate the utility of using more heuristics.

4.2 Discussion

Heuristic Weighted Voting is able to achieve a higher average classification accuracy than any of three standard baseline strategies. A comparison between Table 4.1 and Table 4.3 shows that using single confidence heuristics in weighting the votes of an ensemble can allow the ensemble to achieve a fairly high average predictive accuracy. The traditional heuristics and the ensemble-selected heuristics allow the ensemble to achieve higher accuracy than the four comparison ensemble creation methods. However, using a single heuristic to predict confidence is not sufficient to create an ensemble that can produce a higher average predictive accuracy on a level that is statistically significant, so investigation into additional heuristics is warranted.

Using commonly paired heuristics allows an ensemble to achieve greater average predictive accuracy than Standard Voting, Weighting by Accuracy, SelectBest, and the Modified Stacking strategy, as well as the single heuristic ensembles. Even higher predictive accuracy can be achieved with the Heuristic Weighted Voting strategy of averaging all the heuristics to produce an overall confidence measure with which to weight ensemble votes.

This higher average accuracy does come with a higher cost of computation, but for two-thirds of the heuristics, the increase in computational complexity is only linear. The other one-third of the heuristics requires a cross validation strategy in the training set. The computational complexity for these heuristics could be reduced by reducing the number of folds used in the calculations.

Data Set	Traditional Heuristics	Non-hybrid Ensemble Selected Heuristics	Decision Tree Selected Heuristics	Commonly Paired Heuristics	Heuristic Weighted Voting
audiology	77.88	77.88	75.22	77.43	78.32
bupa	70.15	71.30	70.15	71.59	71.30
cancer	96.42	96.42	96.71	96.28	96.57
car evaluation	96.70	96.24	96.07	98.61	96.64
cmc	53.63	53.63	53.97	52.41	53.63
diabetes	76.56	75.39	76.43	76.56	76.56
ecoli	86.31	86.01	86.31	86.01	87.80
glass	71.96	72.43	70.09	69.16	72.43
haberman	74.18	74.84	73.86	75.16	74.18
hayes	75.76	77.27	74.24	77.27	74.24
heart statlog	83.33	84.82	83.70	81.48	83.33
iris	96.00	96.00	96.00	96.00	96.00
monks	99.77	99.77	99.77	98.61	99.77
postOp	70.00	67.78	71.11	70.00	71.11
sonar	82.69	82.69	83.17	83.17	83.65
tic-tac-toe	93.53	92.80	94.15	97.29	98.43
vote	95.88	96.53	96.10	96.53	95.66
zoo	96.04	95.05	95.05	97.03	96.04
Average:	83.16	83.16	82.89	83.37	83.65

Table 4.3 Comparison of Heuristic Combination Strategies

.5. ARBITRATION WITH HEURISTICS

The heuristics used in Heuristic Weighted Voting can also be used in conjunction with Ortega, Koppel, and Argamom’s ensemble strategy of Arbitration [OKA01]. With Arbitration, meta-learner decision trees called referees are used to determine how confident a classifier should be in its prediction. The authors state that in addition to the attributes found in the individual instances, “intermediate subconcepts” of the classifiers should also be used as features to train the referees in order to obtain a higher overall predictive accuracy. The heuristics presented in this work can be used as these intermediate subconcepts. Using the values calculated with these heuristics improves the accuracy of the referees, and thus increases the predictive accuracy of the overall ensemble.

5.1 Referees

	Decision Tree		Rule-Based		Instance-Based	
	Attributes	With Heuristics	Attributes	With Heuristics	Attributes	With Heuristics
audiology	81.42	78.32	73.45	71.24	72.12	76.99
bupa	55.36	55.07	60.29	62.32	44.06	58.84
cancer	83.69	89.84	81.12	93.28	83.98	95.71
car evaluation	90.22	88.60	92.65	90.86	93.46	86.69
cmc	58.79	60.29	59.88	56.55	55.06	56.48
diabetes	69.27	73.70	68.88	75.00	63.80	72.53
ecoli	77.68	80.06	68.75	79.76	84.52	84.82
glass	56.08	49.07	66.36	70.09	53.27	67.29
haberman	71.90	71.24	70.26	71.57	66.67	69.94
hayes	71.97	71.21	54.55	57.58	74.24	63.64
heart statlog	70.74	77.41	72.96	77.78	76.30	76.30
iris	94.67	84.00	97.33	92.67	94.67	96.67
monks	88.89	91.44	100.00	100.00	98.15	98.61
postOp	70.00	68.89	64.44	70.00	64.44	57.78
sonar	57.69	59.14	62.50	74.52	70.67	77.89
tic-tac-toe	83.09	81.11	80.48	74.01	98.96	95.62
vote	84.38	94.14	85.03	91.97	92.19	91.54
zoo	91.09	91.09	87.13	84.16	96.04	92.08
Average:	75.39	75.81	74.78	77.41	76.81	78.86

Table 5.1 Comparison of Referee Effectiveness

	Naïve Bayes		Multilayer Perceptron	
	Attributes	With Heuristics	Attributes	With Heuristics
audiology	78.76	78.32	78.76	72.12
bupa	49.57	55.94	56.81	56.81
cancer	72.68	95.14	89.56	93.71
car evaluation	83.68	92.94	98.73	98.15
cmc	53.50	56.35	56.01	56.76
diabetes	70.96	71.09	68.1	71.75
ecoli	74.41	83.04	82.74	82.44
glass	65.42	68.69	53.74	57.94
haberman	73.20	72.22	68.95	66.34
hayes	77.27	82.58	69.7	81.06
heart statlog	80.37	77.41	72.96	72.59
iris	95.33	92.67	90	89.33
monks	91.67	96.07	97.69	97.69
postOp	68.89	54.44	51.11	64.44
sonar	67.79	73.08	72.6	72.12
tic-tac-toe	77.24	77.98	96.66	96.45
vote	67.90	89.15	94.36	92.84
zoo	93.07	95.95	84.16	75.25
Average:	74.54	78.50	76.81	77.66

Table 5.1 (cont.) Comparison of Referee Effectiveness

Table 5.1 reports the accuracies of the referees developed using an Arbitration strategy. The first column in each section reflects the accuracy of a decision tree referee in predicting whether an instance was correctly or incorrectly classified when it was given information only about the attribute values. The second column reports referee accuracy when trained on both the attribute values and the values produced by heuristics for the instance.

Observing referee behavior when trained on the various heuristics provides added insight into heuristic effectiveness. For example, with the decision tree, there was a 0.42% increase in average predictive accuracy between a referee predicting confidence from attribute values and a referee predicting confidence from the heuristics. Purity of classification and the heuristics that use information about correctness of classification in hold-one-out cross validation were most likely to assign higher confidence values to correctly classified values. However, all heuristics provided useful information to a referee; all were used as splitting criteria for a referee in at least one of the data sets.

The heuristics developed for the rule-based classifier resulted in a 2.63% increase in the predictive accuracy of a referee. The heuristic looking at correctly classified voting neighbors was used more often than the heuristic reporting all correctly classified neighbors. The heuristic determining whether or not an instance was covered by a rule also appeared less often in decision tree referees, likely because it provided extra information for only a minority of instances. Otherwise, heuristics tended to appear with similar frequency in the refereeing decision trees.

The heuristics developed for the instance-based classifier resulted in a 2.05% increase in the predictive accuracy of the referee. Referees created for the instance-based classifiers were more likely to use a majority of the heuristics. Nearly one-third of the data sets required a referee to use all six heuristics in making a decision about correctness of classification. It appears that the individual heuristics for this classifier are less effective at isolating misclassified instances. This may be due to the nature of the instance-based classifier; while other classifiers impose some sort of grouping on the data, the instance-based classifier does not. Weaknesses and strengths in the classifier are dependent only on the distribution of the data.

For the Naïve Bayes classifier, there was a 3.96% increase in the predictive accuracy of a referee when heuristics were added to the referee training information. This was the highest increase in predictive accuracy of any of the five classifiers observed. Although the first three heuristics presented were somewhat correlated, each appears to have provided enough new information that all were used in arbitration.

There was a 0.85% increase in the predictive accuracy of a referee when given the heuristics for the multilayer perceptron. Considered as a group, these heuristics appear to be stronger in predicting correctness of classification than the heuristics for other learners. However, possibly because the multilayer perceptron heuristics do not provide as much unique information, they give a referee less of an advantage in predictive accuracy than the heuristics for some of the other classifiers.

Data Set	Arbitration	Heuristic Weighted Voting
audiology	78.76	78.32
bupa	66.37	71.30
cancer	95.99	96.57
car evaluation	96.18	96.64
cmc	56.14	53.02
diabetes	77.34	76.56
ecoli	86.31	87.80
glass	67.76	72.43
haberman	71.57	74.18
hayes	77.27	74.24
heart statlog	84.44	83.33
iris	96.00	96.00
monks	100.00	99.77
postOp	70.00	71.11
sonar	81.73	83.65
tic-tac-toe	96.55	98.43
vote	96.53	95.66
zoo	95.05	96.04
Average:	83.00	83.61

Table 5.2 Comparison of Arbitration and Heuristic Weighted Voting

5.2 Arbitration

This section provides a comparison of the Arbitration and Heuristic Weighted Voting strategies using the five classifiers mentioned in the previous sections as the component classifiers of the ensemble. Predictive accuracies of each method are shown in Table 5.2. A comparison between Heuristic Weighted Voting and Arbitration show ten wins, seven losses, and one tie with regards to accuracy on individual data sets. The results provided in this section illustrate not only how heuristic-produced values can be used in conjunction with Arbitration, but also how Heuristic Weighted Voting functions on a competitive level with this previously defined strategy.

6. CONCLUSION AND FUTURE WORK

This work presents a viable new method of combining the votes in an ensemble using heuristics to predict confidence in the classification of a given instance. A number of heuristics designed for this task are proposed for each of five different types of classifiers. These heuristics are generally shown to be effective predictors of whether or not an instance will be correctly or incorrectly classified by the given classifier. Weighting the votes of the classifiers in an ensemble by using a single heuristic for each classifier tends to result in an improvement in predictive accuracy over a strategy of straight voting. The strategy of Heuristic Weighted Voting, which employs all of the heuristics presented, is shown to achieve a higher average classification accuracy over eighteen data sets than four standard ensemble strategies. It also compares favorably in an algorithm-by-algorithm comparison of wins and losses in accuracy over the eighteen data sets.

Future research will involve making refinements to the heuristics presented here, developing new heuristics, and experimenting with subsets of heuristics in order to further increase the predictive accuracy of ensembles created with this technique. One important consideration will be finding effective ways of weighting the heuristics when calculating a final confidence measure rather than using a simple averaging strategy. Another issue to be addressed is how the generated confidence measures compare across classifiers. Due to the number of heuristics and the averaging process, the confidence measures used here fell in roughly the same ranges for all the classifiers. Future work will involve developing a more formal method of normalizing the confidence measures across classifiers. It will also include calibrating the heuristics to bring them closer to true output probabilities.

The heuristics presented in this work explore some of the strengths and weaknesses of a given classifier on a given data set. This information could result in the development of new algorithms. For example, a new instance-based classifier might be developed in which only instances that were correctly classified in hold-one-out cross validation would be allowed to vote on the classification of an unseen instance. The probabilities output by a Naïve Bayes classifier might be altered slightly based on information gained through heuristics like the ones presented here. Insights gained by observing the behavior of the heuristics on various data sets may help target areas of improvement to increase classification accuracy of individual classifiers.

.7. BIBLIOGRAPHY

[Bre96] L. Breiman. Bagging Predictors. Machine Learning 24(2), pp. 123-140, 1996.

[CH67] T. M. Cover and P. Hart. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13, pp. 21–27, 1967.

[CN06] R. Caruana and A. Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. In Proceedings of the International Conference on Machine Learning, 2006.

[DZ04] S. Dzeroski and B. Zenko. Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning, 54:255-273, 2004.

[FFH04] C. Ferri, P. Flach, and J. Hernandez-Orallo. Delegating Classifiers. In Proceedings of the Twenty-first International Conference on Machine Learning, pp. 289-296, 2004.

[FS96] Y. Freund and R. E. Schapire, Experiments with a New Boosting Algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, 1996.

[GR00] G. Giacinto and F. Roli. A Theoretical Framework for Dynamic Classifier Selection. In Proceedings of the Fifteenth International Conference on Pattern Recognition, Number II in Lecture Notes in Computer Science, pp. 8–11, Barcelona, Spain, 2000.

[HBM98] S. Hettich, C. L. Blake, and C. J. Merz. UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[JJN91] R. A. Jacobs, M.I. Jordan, S.J Nowlan, and G.E. Hinton. Adaptive Mixture of Local Experts. Neural Computation, 3:79–87, 1991.

[Koh95] R. Kohavi. The Power of Decision Tables. European Conference of Machine Learning, 1995.

[KS97] J. Kim, K. Seo, and K. Chung. A Systematic Approach to Classifier Selection on Combining Multiple Classifiers for Handwritten Digit Recognition. In Proceedings of the Fourth International Conference on Document Analysis and Recognition, pp. 459–462, Ulm, Germany, 1997.

[KW96] R. Kohavi and D. Wolpert. Bias Plus Variance Decomposition for Zero-One Loss Functions. In L. Saitta (Ed.), Machine Learning: Proceedings of the Thirteenth International Conference, pp. 275–283, Morgan Kaufmann, 1996.

[KW03] L. I. Kuncheva and C. J. Whitaker. Measures of Diversity in Classifier Ensembles. Machine Learning, 51:181–207, 2003.

[Lam00] L. Lam. Classifier Combinations: Implementations and Theoretical Issues. In J. Kittler, and F. Roli (Eds.), Multiple classifier systems, Vol. 1857 of Lecture Notes in Computer Science, pp. 78–86, Cagliari, Italy, Springer, 2000.

[Lan95] K. Lang. NewsWeeder: Learning to Filter Netnews. In Proceedings of the Twelfth International Conference on Machine Learning, pp. 331–339, Morgan Kaufmann Publishers Inc., San Mateo, California, 1995.

[Mer95] C. J. Merz. Dynamic Learning Bias Selection. In Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, pp. 386-395, 1995.

[Mit97] T. M. Mitchell. Machine Learning. WCB/McGraw-Hill, 1997.

[MM04] J. Menke and T. R. Martinez. Using Permutations Instead of Student's T Distribution for p-Values in Paired-Difference Algorithm Comparisons. In Proceedings of the 2004 IEEE Joint Conference on Neural Networks IJCNN'04, 2004.

[MN98] A. McCallum and K Nigam. A Comparison of Event Models for Naive Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization, 1998.

[OKA01] J. Ortega, M. Koppel, and S. Argamon. Arbitrating Among Competing Classifiers Using Learned Referees. Knowledge and Information Systems Journal, 3:4, pp. 470-490, 2001.

[PM05] A. H. Peterson and T. R. Martinez. Estimating the Potential for Combining Learning Models. In Proceedings of the ICML Workshop on Meta-Learning, pp. 68–75, 2005.

[Qui86] J. R. Quinlan. Induction of Decision Trees. Machine Learning, 1(1), pp. 81–106, 1986.

[Qui93] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufman, 1993.

[RG01] D. Ruta and B. Gabrys. Analysis of the Correlation between Majority Voting Error and the Diversity Measures in Multiple Classifier Systems. In Proceedings of the Fourth International Symposium on Soft Computing, ISBN: 3-906454-27-4, Paper No. 1824-025, Paisley, UK, 2001.

[RG05] D. Ruta and B. Gabrys, Classifier Selection for Majority Voting, Information Fusion, 2005.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, pp. 318–362, 1986.

[SS98] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-Rated Predictions. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, 1998.

[WF05] I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.