Working Paper Series
ISSN 1170-487X

# A Decision Tree-Based Attribute Weighting Filter for Naive Bayes

**Mark Hall**

Working Paper: 05/2006
May 29, 2006


Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

# A Decision Tree-Based Attribute Weighting Filter for Naive Bayes

Mark Hall
University of Waikato, Hamilton, New Zealand
mhall@cs.waikato.ac.nz

May 29, 2006

**Abstract**

The naive Bayes classifier continues to be a popular learning algorithm for data mining applications due to its simplicity and linear run-time. Many enhancements to the basic algorithm have been proposed to help mitigate its primary weakness—the assumption that attributes are independent given the class. All of them improve the performance of naive Bayes at the expense (to a greater or lesser degree) of execution time and/or simplicity of the final model. In this paper we present a simple filter method for setting attribute weights for use with naive Bayes. Experimental results show that naive Bayes with attribute weights rarely degrades the quality of the model compared to standard naive Bayes and, in many cases, improves it dramatically. The main advantages of this method compared to other approaches for improving naive Bayes is its run-time complexity and the fact that it maintains the simplicity of the final model.

## 1 Introduction

All practical learning algorithms based on Bayes' theorem make some independence assumptions. The naive Bayes method takes this to the extreme by assuming that the attributes are statistically independent given the class. This leads to a simple algorithm where training time is linear in both the number of instances and attributes. Although the independence assumption is grossly violated in practice, naive Bayes performs surprisingly well on many classification problems [7]. However, because of this assumption, the posterior probabilities estimated by naive Bayes are typically poor. For example, in an extreme case where a single redundant attribute (i.e., an attribute that is perfectly correlated with another) is present in the data, that attribute effectively has twice as much influence as the other attributes.

Many techniques have been developed to reduce the 'naiveity' of the naive Bayes algorithm. Zheng and Webb [28] provide a comprehensive overview of

work in this area. One simple approach that often works well is to combine naive Bayes with a preprocessing step that attempts to remove redundant attributes from the training data. Various methods from the attribute selection community have been applied to naive Bayes for just this purpose. However, one related area that has received little attention with regards to naive Bayes is the use of attribute weights[1].

This paper presents a filter method that sets attribute weights for use with naive Bayes. The assumption made is that the weight assigned to a predictive attribute should be inversely related to the degree of dependency it has on other attributes. Our method estimates the degree of attribute dependency by constructing unpruned decision trees and looking at the depth at which attributes are tested in the tree. A bagging procedure is used to stabilize the estimates. Attributes that do not appear in the decision trees receive a weight of zero. Our experimental results show that using attribute weights with naive Bayes improves the quality of the model compared to standard naive Bayes in terms of probability estimation and area under the ROC curve.

This paper is structured as follows. In Section 2 we present our approach for enhancing naive Bayes by learning attribute weights. Section 3 contains experimental results for a collection of benchmark data sets and shows that the performance of naive Bayes can be improved by using attribute weights. Section 4 discusses related work on enhancing the performance of naive Bayes. Section 5 summarizes the contributions made in this paper.

## 2 Using attribute weights with naive Bayes

Naive Bayes computes the posterior probability of class $c_l$ for a test instance with attribute values $a_1, a_2, ..., a_m$ as follows:

$$p(c_l|a_1, a_2, ..., a_m) = \frac{p(c_l)\prod_{i=1}^{m} p(a_i|c_l)}{\sum_{q=1}^{o}[p(c_q)\prod_{i=1}^{m} p(a_i|c_q)]}, \quad (1)$$

where $o$ is the total number of classes. The term in the denominator of the the right-hand side of Equation 1 can be omitted as it is a normalizing factor. The individual probabilities on the right-hand side of this equation are estimated from the training data. In the case of discrete attributes they are computed from frequency counts. If a numeric attribute is present, we make the normality assumption and estimate its mean and variance. Incorporating attribute weights into the formula gives:

$$p(c_l|a_1, a_2, ..., a_m) = p(c_l)\prod_{i=1}^{m} p(a_i|c_l)^{w_i}, \quad (2)$$

where $w_i$ is the weight of attribute $A_i$.

[1] Attribute selection can be viewed as a special case of attribute weighting where the weights are restricted to zero or one.

1. **Repeat $i$ times:**
2. **Randomly sample (with replacement) $j$% of the training data.**
3. **Learn an unpruned decision tree from the resampled data.**
4. **FOR each attribute in the training data DO:**
5. **IF the attribute is NOT tested in the tree THEN**
6. **Record a weight of 0.**
7. **ELSE**
8. **Let $d$ be the minimum depth that the attribute is tested at.**
9. **Record a weight of $1/\sqrt{d}$.**
10. **FOR each attribute in the training data DO:**
11. **Set the final weight equal to the average of the $i$ weights.**
12. **Optionally remove from the data all attributes with zero weight.**
13. **Learn a naive Bayes model using the final attribute weights.**

Figure 1: Attribute weighted Bayesian classifier algorithm

Our method for enhancing naive Bayes aims to weight predictive features according to the degree to which they depend on the values of other attributes. Since naive Bayes makes the independence assumption we want to assign lower weights to those attributes that have many dependencies. To estimate the degree to which an attribute depends on others, we first construct an unpruned decision tree from the training data and then note the minimum depth[2] at which the attribute is tested in the tree. The weight for an attribute is set to $1/\sqrt{d}$, where $d$ is the minimum depth at which the attribute is tested in the tree. Attributes that do not appear in the tree receive a weight of zero. Since decision tree learners are inherently unstable, we stabilize the estimated weights by building multiple decision trees using bagging and then average the weights across the ensemble. The method has two parameters—$i$ the number of bagging iterations, and $j$ the percentage of the training data to use for learning a tree in each iteration. Our experimental results show that the method is relatively insensitive to the value of $j$. Figure 1 shows the algorithm for the attribute weighted Bayesian classifier.

# 3 Experimental results

This section evaluates the performance of attribute weighted naive Bayes (AWNB) using our tree-based weighting scheme on a collection of 28 benchmark data sets from UCI repository [2]. The properties of these data sets are shown in Table 1.

We ran two sets of experiments. The first compares attribute weighted naive Bayes (using 10 bagging iterations and subsamples of the training data of size 50% for weight estimation) with standard naive Bayes. In this experiment we also evaluate the effect of using weights versus feature selection (i.e. zero/one weights) and the effect of varying the size of the random subsamples used to

[2]The root node of the tree has depth 1.

Table 1: Datasets used for the experiments

| Dataset | Instances | % Missing | Numeric | Nominal | Class |
|---|---|---|---|---|---|
| annl | 898 | 0.0 | 6 | 32 | 5 |
| aust | 690 | 0.6 | 6 | 9 | 2 |
| autos | 205 | 1.1 | 15 | 10 | 6 |
| bal-s | 625 | 0.0 | 4 | 0 | 3 |
| brst-c | 286 | 0.3 | 0 | 9 | 2 |
| brst-w | 699 | 0.3 | 9 | 0 | 2 |
| diab | 768 | 0.0 | 8 | 0 | 2 |
| ecoli | 336 | 0.0 | 7 | 0 | 8 |
| crd-g | 1000 | 0.0 | 7 | 13 | 2 |
| glass | 214 | 0.0 | 9 | 0 | 6 |
| hrt-c | 303 | 0.2 | 6 | 7 | 2 |
| hrt-h | 294 | 20.4 | 6 | 7 | 2 |
| hrt-s | 270 | 0.0 | 13 | 0 | 2 |
| hep | 155 | 5.6 | 6 | 13 | 2 |
| colic | 368 | 23.8 | 7 | 15 | 2 |
| hypo | 3772 | 6.0 | 23 | 6 | 4 |
| iono | 351 | 0.0 | 34 | 0 | 2 |
| iris | 150 | 0.0 | 4 | 0 | 3 |
| kr | 3196 | 0.0 | 0 | 36 | 2 |
| labor | 57 | 3.9 | 8 | 8 | 2 |
| lymph | 148 | 0.0 | 3 | 15 | 4 |
| sick | 3772 | 6.0 | 23 | 6 | 2 |
| sonar | 208 | 0.0 | 60 | 0 | 2 |
| splice | 3190 | 0.0 | 0 | 61 | 3 |
| vehic | 846 | 0.0 | 18 | 0 | 4 |
| vote | 435 | 5.6 | 0 | 16 | 2 |
| wave | 5000 | 0.0 | 40 | 0 | 3 |
| zoo | 101 | 0.0 | 1 | 15 | 7 |

build the trees. Finally, we include results for bagged unpruned decision trees, so that the reader can compare AWNB with the performance of just using the source of our weight estimation procedure for prediction. The second experiment compares our tree-based method for setting attribute weights to a simple weighting scheme based on gain ratio, a weighting scheme based on the ReliefF attribute ranking algorithm [14, 22], the CFS attribute selection algorithm [10], a decision tree-based attribute selection scheme [21], the wrapper-based selective Bayes algorithm [18] and the NBTree decision tree/naive Bayes hybrid [15]. The latter is included as an example of an enhanced naive Bayes variant—with a richer representational structure than naive Bayes with or without feature selection/weighting—that still maintains a single interpretable model.

In our experiments we calibrated the probability estimates produced by each learning algorithm by fitting a linear logistic regression function to the outputs of the learner generated during an internal run of 10 fold cross-validation on the training data. For each data set/classifier combination we performed five separate runs of 10 fold cross-validation and computed the average root relative squared error (RRSE) of the probability estimates produced over all 50 folds. For a single train/test split the RRSE is given by the square root of the quadratic loss [26] of the learning algorithm normalized by the quadratic loss of simply predicting the most common class present in the training data. We also computed the area under the ROC curve (AUC) for the 16 two-class datasets.

Throughout we speak of two results for a data set as being "significantly different" if the difference is statistically significant at the 5% level according to the corrected resampled $t$-test [19], which has acceptable Type I error.

Table 2: Experimental results for attribute weighted naive Bayes (AWNB) versus naive Bayes (NB), AWNB using zero/one weights (ASNB) and bagged unpruned decision trees: mean root relative squared error (RRSE) and standard deviation.

| Data | AWNB $i=10$ $j=50$ | NB | AWNB $i=1$ $j=100$ | AWNB $i=10$ $j=25$ | AWNB $i=10$ $j=75$ | AWNB $i=10$ $j=100$ | ASNB $i=10$ $j=50$ | Bagging $i=10$ $j=100$ |
|---|---|---|---|---|---|---|---|---|
| annl | 50.38± 8.7 | 53.64± 7.3 | 54.43± 9.3 | 49.14±11.1 | 51.57± 9.0 | 51.52± 8.9 | 54.38± 6.5 ○ | 20.03±12.8 ● |
| aust | 70.03± 7.0 | 79.27± 5.8 ○ | 72.46± 7.0 | 67.43± 6.7 ● | 71.72± 6.9 ○ | 72.07± 7.0 ○ | 79.57± 5.9 ○ | 66.33± 8.2 |
| autos | 80.36± 9.3 | 85.73± 8.8 ○ | 81.39± 8.9 | 81.14± 9.5 | 80.71± 9.5 | 80.80± 9.2 | 85.57± 8.8 ○ | 54.59±14.9 ● |
| bal-s | 46.55± 6.1 | 41.18± 6.1 ● | 58.79± 6.0 ○ | 46.21± 5.6 | 47.82± 5.8 | 48.41± 6.5 | 41.18± 6.1 ● | 58.97± 6.2 ○ |
| brst-c | 94.84± 6.2 | 94.40± 6.3 | 94.99± 6.0 | 95.20± 5.9 | 94.76± 6.2 | 94.72± 6.1 | 94.40± 6.3 | 97.61± 5.1 |
| brst-w | 37.60±11.5 | 39.29±10.4 | 37.39±12.0 | 38.43±11.6 | 38.14±11.4 | 38.35±11.0 | 39.11±10.6 | 37.54± 9.4 |
| diab | 84.72± 6.9 | 86.28± 6.3 ○ | 84.90± 6.9 | 84.63± 7.0 | 84.78± 6.9 | 84.86± 6.9 | 86.28± 6.3 ○ | 85.31± 5.6 |
| ecoli | 56.18±10.8 | 57.72±11.4 | 56.90±11.2 | 57.31±10.9 | 56.35±11.0 | 57.22±10.9 | 57.40±11.0 | 59.91±10.1 |
| crd-g | 89.54± 4.4 | 89.80± 4.3 | 89.73± 4.3 | 89.42± 4.5 | 89.70± 4.5 | 89.64± 4.4 | 89.80± 4.3 | 93.05± 3.4 ○ |
| glass | 86.07± 5.5 | 85.98± 5.5 | 86.74± 5.5 | 85.90± 5.5 | 85.89± 5.5 | 85.93± 5.3 | 85.98± 5.5 | 73.91± 8.3 ● |
| hrt-c | 68.80±13.8 | 69.37±14.3 | 69.42±13.6 | 69.46±13.2 | 68.68±14.3 | 68.66±14.2 | 69.37±14.3 | 75.00± 9.5 |
| hrt-h | 72.91±13.7 | 71.16±13.2 | 72.63±13.9 | 73.83±13.0 | 72.38±13.9 | 72.16±13.8 | 71.20±13.2 | 75.56±11.5 |
| hrt-s | 70.76±10.1 | 69.86±10.7 | 71.08± 9.5 | 70.73±10.0 | 70.65±10.2 | 70.53±10.0 | 69.86±10.7 | 74.78± 9.7 |
| hep | 83.43±18.3 | 79.84±18.9 | 85.74±16.6 | 84.74±16.9 | 83.97±17.7 | 83.89±16.5 | 79.78±19.5 | 88.38±14.0 |
| colic | 71.35±10.8 | 79.58± 8.6 ○ | 72.31±10.2 | 70.60±11.1 | 71.79±10.4 | 72.09±10.3 | 79.35± 8.8 ○ | 70.62±11.2 |
| hypo | 76.45± 5.6 | 71.87± 5.4 ● | 75.79± 5.6 ● | 77.09± 5.6 ○ | 76.18± 5.6 ● | 76.01± 5.5 ● | 71.35± 5.5 ● | 18.57± 9.6 ● |
| iono | 57.17±14.8 | 72.63±10.8 ○ | 58.23±14.8 | 57.71±14.3 | 56.04±14.2 | 55.77±15.7 | 63.07±13.4 | 50.03±13.9 |
| iris | 26.23±20.5 | 27.77±20.1 | 26.23±20.7 | 25.69±20.8 | 26.23±20.4 | 26.32±20.4 | 26.34±20.0 | 31.67±20.3 |
| kr | 38.98± 3.7 | 59.53± 4.1 ○ | 40.72± 3.8 ○ | 38.79± 3.9 | 39.54± 3.5 ○ | 40.19± 3.6 ○ | 53.65± 3.9 ○ | 12.01± 5.9 ● |
| labor | 58.65±30.3 | 36.50±31.3 | 64.45±32.9 | 67.72±28.7 | 57.08±30.6 | 54.93±31.6 | 45.46±31.0 | 66.22±28.2 |
| lymph | 75.18±15.7 | 72.40±17.2 | 77.75±15.1 | 75.31±15.6 | 74.46±16.5 | 74.26±16.5 | 73.78±16.7 | 77.56±16.3 |
| sick | 64.88± 7.7 | 76.60± 5.4 ○ | 66.03± 7.5 | 65.06± 8.0 | 64.98± 7.5 | 65.14± 7.5 | 75.73± 5.6 ○ | 37.57±10.8 ● |
| sonar | 86.94±10.8 | 91.10± 7.9 ○ | 90.48± 9.5 | 86.73±10.5 | 87.16±11.6 | 86.99±11.1 | 91.82± 8.5 ○ | 75.40±11.9 ● |
| splice | 30.76± 3.5 | 34.66± 3.9 ○ | 31.21± 3.7 | 31.65± 3.4 ○ | 30.41± 3.4 | 30.26± 3.7 | 34.36± 3.8 ○ | 38.64± 4.1 ○ |
| vehic | 88.00± 2.4 | 91.18± 1.8 ○ | 88.36± 2.3 | 87.45± 2.4 | 88.37± 2.3 ○ | 88.41± 2.2 ○ | 91.18± 1.8 ○ | 64.04± 4.5 ● |
| vote | 39.67±12.5 | 58.04±12.1 ○ | 42.00±12.4 | 38.23±12.8 | 40.73±12.3 | 41.46±12.3 | 54.90±12.5 ○ | 32.98±14.9 ● |
| wave | 59.72± 2.3 | 63.52± 2.3 ○ | 59.94± 2.2 | 59.50± 2.4 | 59.77± 2.2 | 59.79± 2.2 | 63.52± 2.3 ○ | 62.50± 2.6 ○ |
| zoo | 39.32±29.7 | 25.15±25.2 | 56.19±25.7 | 51.33±28.9 | 40.99±29.7 | 33.87±29.2 | 34.97±26.3 | 38.62±23.6 |

●, ○ statistically significant improvement or degradation over AWNB with $i = 10, j = 50$.

Table 2 shows the RRSE results for the first experiment. Compared to standard naive Bayes, attribute weighted naive Bayes ($i = 10, j = 50$) has significantly lower RRSE on 12 data sets and significantly higher RRSE on only two data sets. In many cases our method improves the performance of naive Bayes considerably. For example, on the kr data RRSE decreases from 59.5% to 39%. Similar levels of improvement can be seen on vote, iono, aust and colic. In order to determine whether the improvement over standard naive Bayes is due to the attribute weights or just feature selection (recall that attributes that do not appear in the tree(s) receive zero weight) we ran AWNB and set all non-zero weights to 1. This scheme is referred to as ASNB in Table 2. From the results it is clear that the attribute weights do help in improving the quality of the probability estimates produced by naive Bayes. Using attribute weights, as opposed to just eliminating those attributes that do not appear in the trees, results in significant improvements on 12 data sets and significant degradation on two. We also investigated the effect of varying the size of the samples used to build the trees for AWNB. Setting $j$ to 25, 75 and 100 resulted in similar performance to using $j = 50$. $j = 50$ is significantly worse than the other two settings of $j$ on one data set, and significantly better on two data sets ($j = 25$) and three data sets ($j = 75$, $j = 100$). Instead of using 10 bagging iterations to construct 10 trees, we also tried building just one tree ($i = 1$) using all the

Table 3: Experimental results for attribute weighted naive Bayes (AWNB) versus naive Bayes (NB), AWNB using zero/one weights (ASNB) and bagged unpruned decision trees: area under the ROC curve and standard deviation.

| Data | AWNB $i=10$ $j=50$ | NB | AWNB $i=1$ $j=100$ | AWNB $i=10$ $j=25$ | AWNB $i=10$ $j=75$ | AWNB $i=10$ $j=100$ | ASNB $i=10$ $j=50$ | Bagging $i=10$ $j=100$ |
|---|---|---|---|---|---|---|---|---|
| aust | 90.67± 3.2 | 89.53± 3.4 ◦ | 90.13± 3.4 | 91.21± 3.1 | 90.30± 3.3 | 90.25± 3.4 | 89.50± 3.4 ◦ | 91.91± 3.3 |
| brst-c | 69.87±10.3 | 70.38±10.0 | 70.08±10.4 | 69.77±10.1 | 70.08±10.3 | 70.15±10.4 | 70.38±10.0 | 63.66±11.5 |
| brst-w | 98.67± 1.2 | 98.33± 1.2 | 98.78± 1.1 | 98.63± 1.2 | 98.62± 1.3 | 98.60± 1.2 | 98.35± 1.2 | 98.78± 1.0 |
| diab | 82.37± 5.6 | 81.42± 5.4 | 82.26± 5.4 | 82.31± 5.7 | 82.34± 5.5 | 82.27± 5.5 | 81.42± 5.4 | 81.25± 5.0 |
| crd-g | 78.84± 4.7 | 78.66± 4.6 | 78.66± 4.7 | 78.91± 4.9 | 78.74± 4.8 | 78.81± 4.8 | 78.66± 4.6 | 73.15± 4.8 ◦ |
| hrt-c | 90.60± 6.1 | 90.70± 6.4 | 90.49± 6.2 | 90.31± 6.1 | 90.57± 6.5 | 90.63± 6.3 | 90.70± 6.4 | 87.98± 5.7 |
| hrt-h | 90.50± 5.9 | 90.74± 5.5 | 90.24± 6.2 | 90.17± 5.5 | 90.61± 5.7 | 90.47± 5.8 | 90.70± 5.5 | 88.49± 6.7 |
| hrt-s | 89.19± 5.9 | 90.20± 5.5 | 89.83± 5.4 | 89.41± 5.9 | 89.58± 5.9 | 89.76± 5.8 | 90.20± 5.5 | 88.37± 5.9 |
| hep | 84.53±13.6 | 85.26±12.9 | 85.89±11.2 | 85.60±12.0 | 85.29±12.8 | 84.96±12.8 | 85.26±13.0 | 81.72±12.7 |
| colic | 88.16± 6.0 | 84.34± 6.2 ◦ | 87.68± 6.1 | 88.67± 6.0 | 87.72± 6.0 | 87.70± 6.1 | 84.42± 6.2 ◦ | 88.93± 5.2 |
| iono | 94.79± 4.1 | 93.81± 3.6 | 93.78± 4.7 | 94.60± 4.4 | 95.04± 3.8 | 94.99± 4.0 | 94.82± 3.6 | 96.48± 3.7 |
| kr | 98.93± 0.4 | 95.21± 1.2 ◦ | 98.68± 0.5 ◦ | 98.96± 0.4 | 98.85± 0.4 ◦ | 98.76± 0.5 ◦ | 96.76± 0.9 ◦ | 99.95± 0.1 • |
| labor | 95.08±11.4 | 97.38± 6.4 | 90.25±19.1 | 91.58±13.3 | 95.17±11.1 | 95.58±11.1 | 96.67± 8.8 | 92.08±15.8 |
| sick | 93.26± 4.4 | 92.59± 3.7 | 93.26± 4.4 | 93.26± 4.4 | 93.25± 4.3 | 93.24± 4.4 | 92.56± 3.8 | 99.28± 1.3 • |
| sonar | 79.33±10.4 | 78.94±10.2 | 77.18±10.3 | 79.39±11.1 | 79.02±10.9 | 79.38±11.0 | 78.82±10.5 | 87.36± 7.4 • |
| vote | 98.97± 1.0 | 97.39± 1.8 ◦ | 98.66± 1.4 | 99.04± 1.0 | 98.93± 1.0 | 98.89± 0.9 | 97.75± 1.6 ◦ | 98.33± 2.4 |

•, ◦ statistically significant improvement or degradation over AWNB with $i = 10, j = 50$.

training data. Although there are only three significant differences compared to building 10 trees, it can be seen from Table 2 that on all but two data sets this results in either a higher RRSE or larger standard deviation. Finally, we compared AWNB to bagged unpruned decision trees using 10 bagging iterations ($i = 10$) and randomly sampled training sets of the same size as the original training data ($j = 100$). With nine significant wins and four significant losses in favour of bagged trees it is clear that AWNB is inferior to this ensemble method. However, AWNB's single model has the advantage of being interpretable.

Table 3 shows the area under the curve results on the two class data sets for the first experiment. Compared to standard naive Bayes, AWNB ($i = 10, j = 50$) is significantly better on four data sets and significantly worse on none. Varying the percentage of data used to build the trees and the number of bagging iterations has minimal effect. The other settings of $i$ and $j$ are significantly worse on only one data set. Compared to simply eliminating those attributes that do not appear in the trees (ASNB), AWNB is significantly better on four data sets and significantly worse on none. Compared to bagged unpruned decision trees, AWNB is significantly better on one data set and significantly worse on three.

Table 4 shows the RRSE results for the second experiment. In this experiment we compared AWNB against two other attribute weighting schemes for naive Bayes, three feature selection methods and naive Bayes trees. The first of the weighting methods (GRW) assigns weights to attributes proportional to their gain ratio score [27]:

$$w_i = \frac{GainRatio(A_i) \times m}{\sum_{i=1}^{m} GainRatio(A_i)}, \tag{3}$$

where $m$ is the number of attributes. For the purpose of computing these weights, all numeric attributes are discretized in a copy of each train/test split using the supervised discretization method of Fayyad and Irani [8]. Compared to GRW, we can see from Table 4 that our tree-based method for determining

Table 4: Experimental results for attribute weighted naive Bayes (AWNB) versus naive Bayes with gain ratio based weighting (GRW), naive Bayes with ReliefF based weighting (RW), naive Bayes with correlation-based feature selection (CFS), Selective Bayes (SB), the Selective Bayesian classifier (SBC) and NBTree: mean root relative squared error (RRSE) and standard deviation.

| Data | AWNB | GRW | RW | CFS | SB | SBC | NBTree |
|---|---|---|---|---|---|---|---|
| annl | 50.38± 8.7 | 57.21± 7.0 ○ | 49.89± 6.5 | 60.58± 7.3 ○ | 60.46±12.3 | 69.12± 8.2 ○ | 26.18±11.8 ● |
| aust | 70.03± 7.0 | 72.89± 7.6 ○ | 68.62± 7.8 | 77.95± 7.8 ○ | 66.22± 7.6 | 76.58± 7.6 ○ | 68.69± 7.4 |
| autos | 80.36± 9.3 | 86.31± 8.1 ○ | 81.29± 8.9 | 84.82± 7.2 | 88.12±18.1 | 85.15± 8.2 ○ | 71.05±12.9 ● |
| bal-s | 46.55± 6.1 | 47.65± 5.8 | 48.62± 6.5 | 41.52± 5.9 ● | 41.18± 6.1 ● | 41.18± 6.1 ● | 78.03± 5.6 ○ |
| brst-c | 94.84± 6.2 | 94.58± 5.6 | 94.91± 5.8 | 94.52± 6.2 | 95.61± 5.3 | 94.92± 6.2 | 95.21± 5.5 |
| brst-w | 37.60±11.5 | 39.50±10.5 | 37.91±10.1 | 39.29±10.4 | 39.48±11.2 | 39.38±12.1 | 34.85±11.5 |
| diab | 84.72± 6.9 | 86.09± 7.0 ○ | 85.35± 7.2 | 85.17± 6.3 | 85.41± 6.1 | 86.60± 6.5 ○ | 85.99± 6.4 |
| ecoli | 56.18±10.8 | 57.70±10.1 | 58.46±10.1 | 57.39±11.5 | 58.38±10.9 | 58.79±12.5 | 66.03±10.7 ○ |
| crd-g | 89.54± 4.4 | 90.59± 4.2 | 90.85± 4.0 | 91.92± 3.7 ○ | 91.26± 3.5 | 90.61± 4.0 | 91.12± 4.1 |
| glass | 86.07± 5.5 | 86.06± 5.3 | 86.35± 5.3 | 85.96± 6.0 | 89.31± 6.2 | 87.64± 5.7 | 80.32±10.4 |
| hrt-c | 68.80±13.8 | 71.12±14.6 | 71.66±12.2 | 70.39±14.8 | 72.95±12.5 | 75.11±19.0 | 73.77±12.8 |
| hrt-h | 72.91±13.7 | 77.43±13.1 ○ | 80.77±10.4 | 72.24±13.1 | 74.91±13.0 | 79.41±19.0 | 73.80±12.8 |
| hrt-s | 70.76±10.1 | 71.21±10.1 | 72.09±11.0 | 71.39±10.4 | 72.85±10.3 | 72.03±11.5 | 76.72±11.0 |
| hep | 83.43±18.3 | 85.33±17.0 | 81.88±15.6 | 82.77±18.8 | 89.93±13.5 | 88.74±15.6 | 88.24±15.7 |
| hrse-c | 71.35±10.8 | 77.56± 9.5 ○ | 76.87±10.3 ○ | 74.97± 9.0 | 74.26±10.2 | 76.40±10.0 ○ | 76.52± 9.6 |
| hypo | 76.45± 5.6 | 70.20± 6.0 ● | 72.62± 4.7 ● | 78.64± 5.3 ○ | 71.89± 5.6 ● | 73.36± 6.1 ● | 20.16±12.2 ● |
| iono | 57.17±14.8 | 65.43±12.9 | 70.03±11.3 ○ | 59.70±15.6 | 57.01±13.2 | 59.40±14.4 | 60.84±12.1 |
| iris | 26.23±20.5 | 26.87±20.3 | 27.20±20.5 | 24.95±20.2 | 28.39±21.0 | 26.46±19.8 | 34.88±21.3 |
| kr | 38.98± 3.7 | 50.34± 4.4 ○ | 52.55± 5.0 ○ | 48.67± 4.0 ○ | 40.86± 5.0 | 54.91± 5.6 ○ | 23.27± 9.1 ● |
| labor | 58.65±30.3 | 62.77±33.0 | 54.80±33.1 | 58.95±32.8 | 51.87±34.0 | 79.46±28.0 ○ | 42.92±33.7 |
| lymph | 75.18±15.7 | 77.68±15.4 | 78.39±17.5 | 76.81±15.9 | 74.35±14.3 | 78.61±17.6 | 75.30±19.4 |
| sick | 64.88± 7.7 | 81.91± 3.4 ○ | 93.46± 1.8 ○ | 72.42± 6.6 ○ | 66.71± 7.1 | 72.34± 8.1 ○ | 56.45± 7.8 ● |
| sonar | 86.94±10.8 | 93.70± 7.7 ○ | 90.08±11.1 | 92.05± 8.7 ○ | 88.35± 9.5 | 89.73± 9.6 | 82.34±11.5 |
| splice | 30.76± 3.5 | 40.23± 3.7 ○ | 36.75± 3.9 ○ | 32.62± 3.6 ○ | 35.98± 3.8 ○ | 36.38± 4.4 ○ | 34.80± 3.9 ○ |
| vehic | 88.00± 2.4 | 92.17± 1.6 ○ | 93.33± 1.5 ○ | 88.75± 2.4 | 89.86± 2.9 | 90.65± 3.0 ○ | 70.84± 4.1 ● |
| vote | 39.67±12.5 | 47.50±14.2 ○ | 41.12±14.3 | 40.81±13.0 | 39.19±12.8 | 40.17±14.7 | 39.87±14.3 |
| wave | 59.72± 2.3 | 67.15± 2.0 ○ | 63.73± 2.4 ○ | 62.86± 2.3 ○ | 62.53± 2.4 ○ | 63.38± 2.8 ○ | 67.84± 3.5 ○ |
| zoo | 39.32±29.7 | 31.60±25.2 | 39.80±24.7 | 38.19±23.3 | 28.00±25.9 | 73.09±21.1 ○ | 29.58±27.6 |

●, ○ statistically significant improvement or degradation over AWNB with $i = 10, j = 50$.

weights results in significantly lower RRSE on 13 data sets and significantly higher RRSE error on only one. This suggests that information about attribute dependencies captured in the tree structure is useful when setting weights for naive Bayes. The second weighting scheme (RW) applies the ReliefF attribute selection [14, 22] algorithm and uses the resulting attribute relevance scores as weights. Any relevance scores less than zero are set to zero. Compared to AWNB, RW is significantly worse on seven data sets and significantly better on one. The ReliefF algorithm can identify relevant attributes that depend on the values of other attributes. However, for the purposes of naive Bayes this can result in scores that are too high for attributes with many dependencies.

Columns five through seven of Table 4 show the results for naive Bayes when combined with the three feature selection algorithms. Correlation-based feature selection (CFS) [10] is particularly well suited for use with naive Bayes as its evaluation heuristic prefers subsets of attributes with low levels of redundancy. From the results we can see that AWNB is significantly better than CFS on nine data sets and significantly worse on one. Selective Bayes (SB) is a wrapper-based feature selection method developed by Langley and Sage [18]. AWNB achieves results comparable to SB—each significantly outperforms the other on two data sets. The Selective Bayesian Classifier (SBC) is a bagged decision-tree based attribute selection filter for naive Bayes [21]. From Table 4 we can see that SBC is significantly better than AWNB on two data sets and significantly worse on 12. The last column in the table shows the results for naive Bayes

Table 5: Experimental results for attribute weighted naive Bayes (AWNB) versus naive Bayes with gain ratio based weighting (GRW), naive Bayes with ReliefF based weighting (RW), naive Bayes with correlation-based feature selection (CFS), Selective Bayes (SB), the Selective Bayesian classifier (SBC) and NBTree: area under the ROC curve and standard deviation.

| Data | AWNB | GRW | RW | CFS | SB | SBC | NBTree |
|---|---|---|---|---|---|---|---|
| aust | 90.67± 3.2 | 90.51± 3.3 | 91.74± 2.9 | 90.12± 3.4 | 90.96± 3.3 | 89.88± 3.8 | 91.33± 3.5 |
| brst-c | 69.87±10.3 | 70.95±10.0 | 69.28±10.7 | 69.87±11.0 | 67.92±11.0 | 69.57±10.5 | 68.45±10.3 |
| brst-w | 98.67± 1.2 | 98.33± 1.2 | 98.42± 1.2 | 98.33± 1.2 | 98.94± 1.0 | 98.67± 1.3 | 98.75± 1.3 |
| diab | 82.37± 5.6 | 81.99± 5.7 | 82.53± 5.9 | 82.03± 5.6 | 82.14± 5.6 | 81.04± 5.4 | 80.89± 5.8 |
| crd-g | 78.84± 4.7 | 77.29± 5.1 ○ | 77.52± 4.8 | 75.27± 4.9 ○ | 78.39± 4.5 | 77.57± 4.7 | 76.23± 5.8 |
| hrt-c | 90.60± 6.1 | 89.99± 7.0 | 90.57± 5.5 | 89.32± 7.4 | 88.99± 5.4 | 87.16±12.2 | 88.21± 6.7 |
| hrt-h | 90.50± 5.9 | 89.11± 6.4 | 89.37± 5.7 | 90.76± 5.1 | 91.01± 5.8 | 87.22±10.4 | 89.22± 7.4 |
| hrt-st | 89.19± 5.9 | 89.47± 6.2 | 89.72± 5.7 | 88.78± 6.2 | 88.97± 5.7 | 88.78± 6.1 | 85.47± 7.7 |
| hep | 84.53±13.6 | 86.95±12.4 | 86.52±12.3 | 85.63±14.5 | 82.71±13.6 | 80.48±16.0 | 81.34±12.6 |
| colic | 88.16± 6.0 | 87.90± 6.3 | 88.45± 5.9 | 86.11± 6.2 | 87.20± 5.5 | 85.32± 6.2 ○ | 85.96± 6.2 |
| iono | 94.79± 4.1 | 94.64± 3.7 | 94.37± 3.6 | 94.32± 4.3 | 95.99± 3.8 | 93.21± 5.6 | 93.19± 4.3 |
| kr | 98.93± 0.4 | 98.11± 0.6 ○ | 98.50± 0.6 ○ | 96.86± 0.8 ○ | 98.62± 0.6 | 94.31± 1.4 ○ | 99.54± 0.5 ● |
| labor | 95.08±11.4 | 94.25± 9.4 | 94.87± 9.8 | 94.25±12.0 | 96.79± 6.4 | 81.21±22.2 | 95.92±10.6 |
| sick | 93.26± 4.4 | 93.55± 4.0 | 90.86± 3.0 ○ | 93.68± 3.3 | 94.41± 2.7 | 92.85± 4.3 | 93.74± 4.0 |
| sonar | 79.33±10.4 | 79.02±10.9 | 81.82±10.7 | 79.49±10.9 | 83.62± 9.7 | 76.81±11.4 | 83.11± 9.2 |
| vote | 98.97± 1.0 | 98.24± 1.4 ○ | 98.82± 1.1 | 98.88± 1.3 | 98.92± 1.4 | 98.36± 1.5 | 98.77± 1.5 |

●, ○ statistically significant improvement or degradation over AWNB with $i = 10, j = 50$.

Trees (NBTree) [15]. NBTree builds a decision tree with local naive Bayes models at each leaf. NBTree's superior representational power is reflected in six significant wins versus four significant losses against AWNB. However, these gains in predictive performance come at the cost of increased running time over our method.

Table 5 shows the area under the curve results on the two class data sets for the second experiment. It is interesting to note that AWNB is significantly outperformed in terms of AUC in only one case—on the kr-vs-kp data by NBTree. Looking at the two weighting schemes (GR and RW), we can see that AWNB is significantly better that GR on three data sets and significantly better than RW on two data sets. AWNB has two significant wins against both CFS and SBC. For this experiment we modified the wrapper-based SB to optimize AUC rather than accuracy. From Table 5 we can see that AWNB is comparable to SB in terms of AUC as there are no significant differences on any of the data sets. However, AWNB is much faster than SB because it has running time that is still linear in the number of attributes (log-linear in the number of instances) while SB's running time is at least quadratic in the number of attributes.

# 4 Related work

A fair amount of work has been done in investigating attribute weighting schemes in the context of nearest neighbor learning. The primary goal of these methods is to mitigate the "curse of dimensionality", where the number of training cases needed to maintain a given error rate grows rapidly with the number of attributes. Methods can be roughly divided into two groups: wrapper approaches, i.e. those that use performance feedback from the nearest neighbor method to adjust the values of the weights, and filter methods, i.e. those that incorporate another model's fixed bias in a preprocessing step to set the weights. Examples

of the former include Salzberg's EACH system [23], Aha's IB4 [1], The Relief system and its extensions [14, 22] and the DIET scheme [16] (which, unlike the other methods, restricts the weight space to a small user-selectable set of discrete values). Examples of the latter include using information theoretic-based measures such as mutual information and gain ratio to assign feature weights [25], setting weights based on class conditional probabilities [5] and Stanfill and Waltz's value-difference metric [24]. Weighting schemes for nearest neighbor can also be separated into those methods that find one globally applicable set of weights (all the previously mentioned methods fall into this category) and those that find locally applicable weights—either local to training instances or specifically tailored to the test instance. Examples of local weighting schemes include those by Cardie and Howe [4, 11] and the RC algorithm of Domingos [6]. For a good survey of attribute weighting methods for nearest neighbor algorithms see Wettschereck et al. [25].

There is comparatively less work on using attribute weights in conjunction with naive Bayes. Zhang and Sheng [27] investigate a gain ratio-based weighting scheme and several wrapper-based methods for finding attribute weights in order to improve AUC performance for naive Bayes. In a text classification setting Kim et al. [13] explore information gain and chi-square statistics to set attribute weights for a Poisson naive Bayes model. Ferreira et al. [9] discretize numeric attributes and then compute a weight for each attribute that is proportional to how predictive of the class it is. Classification accuracy for a weighted version of naive Bayes is compared to standard naive Bayes and C4.5 on a small selection of UCI data sets. Unfortunately, comparison based on accuracy is unfair if calibration or threshold selection is not used (which would be necessary to maximize the accuracy of each classifier).

Using information captured in decision trees to improve the performance of other learning algorithms was first explored by Cardie [3] and Kubat et al. [17]. Cardie used only those features appearing in a C4.5 [20] tree as input to a nearest neighbor learner, while Kubat et al. did the same for naive Bayes. In an approach very similar to these two, Ratanamahatana and Gunopulos' [21] Selective Bayesian classifier (SBC) uses attributes that appear in only the top three levels of a decision tree to improve the performance of naive Bayes. Similar to our method, SBC uses a bagging procedure to generate multiple trees; unlike our method bagging is used primarily to speed up the tree growing process and so only a small percentage (10%) of the training data is sampled in each iteration. In an approach related to the one presented in this paper, Cardie [4] used an information gain metric based on the position of an attribute in a decision tree to derive feature weights for a nearest neighbor algorithm. This method differs from ours in that it is a local weighting scheme (i.e. weights are derived for each test instance according to the path it takes through the tree), it derives weights directly from information gain scores, it uses a single pruned decision tree and it is aimed at improving the prediction of minority classes.

The literature on feature selection in machine learning is too extensive to review in detail here, so we will mention just a few articles relevant to the investigation reported in this paper. John et al. [12] are credited with coining the

terms "wrapper" and "filter" to describe those methods that use performance feedback from a learning algorithm to guide the search for good features versus those that incorporate another model's fixed bias to select features. An example of the wrapper approach specifically tailored to naive Bayes is Langley and Sage's selective Bayesian classifier [18]. This method uses the accuracy of naive Bayes on the training data to evaluate feature subsets and a conservative forward selection search that continues to add attributes as long as the predictive performance does not decrease. CFS (correlation-based feature selection) [10] is an example of a filter approach that is well suited to naive Bayes. CFS uses a heuristic that is biased towards subsets of features that are highly correlated with the class attribute and have low levels of redundancy.

# 5 Conclusions

This paper has investigated a decision tree-based filter method for setting attribute weights for use with naive Bayes. Empirically, our attribute weighting method for naive Bayes outperforms both standard naive Bayes and weighting methods based on information gain and the ReliefF algorithm. Furthermore, it has performance that is comparable with a more computationally intensive wrapper-based feature subset selection for naive Bayes.

In terms of computational complexity, our weighting method increases naive Bayes' runtime from linear in the number of attributes and examples to linear in the number of attributes and log-linear in the number of instances. This compares favourably with other enhanced versions of naive Bayes that maintain a single interpretable model such as naive Bayes trees [15] and selective Bayes, both of which are quadratic in the number of attributes.

# References


[1] D. W. Aha. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *Int. Journal of Man-Machine Studies*, 36:267–287, 1992.

[2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Science, 1998. [www.ics.uci.edu/∼mlearn/MLRepository.html].

[3] C. Cardie. Using decision trees to improve case-based learning. In *Proc. of the 10th Int. Conf. on Machine Learning*, pages 25–32. Morgan Kaufmann, 1993.

[4] C. Cardie and N. Howe. Improving minority class prediction using case-specific feature weights. In *Proc. of the 14th Int. Conf. on Machine Learning*, pages 57–65. Morgan Kaufmann, 1997.

[5] R. H. Creecy, B. M. Masand, S. J. Smith, and D. L. Waltz. Trading MIPS and memory for knowledge engineering. *Communications of the ACM*, 35:48–64, 1992.

[6] P. Domingos. Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11(227–253), 1997.

[7] P. Domingos and M. J. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learning*, 29(2-3):103–130, 1997.

[8] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the 13th Int. Joint Conf. on AI*, pages 1022–1027. Morgan Kaufmann, 1993.

[9] J. T. A. S. Ferreira, D. G. T Denison, and D. J. Hand. Data mining with products of trees. In *Proc. of the 4th Int. Conf. on Advances in Intelligent Data Analysis*, pages 167–176. Springer, 2001.

[10] M. Hall. Correlation-based feature selection for discrete and numeric c lass machine learning. In *Proc. of the 17th Int. Conf. on Machine Learning*, pages 359–366, 2000.

[11] N. Howe and C. Cardie. Examining locally varying weights for nearest neighbor algorithms. In *Case-Based Reasoning Research and Development: 2nd Int. Conf. on Case-Based Reasoning*, pages 455–466. Springer, 1997.

[12] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. of the 11th Int. Conf. on Machine Learning*, pages 121–129. Morgan Kaufmann, 1994.

[13] S. Kim, H. Seo, and H. Rim. Poisson naive Bayes for text classification with feature weighting. In *Proc. of the 6th Int. Workshop on Information Retrieval with Asian Languages*, pages 33–40, 2003.

[14] K. Kira and L. Rendell. A practical approach to feature selection. In *Proc. of the Ninth Int. Conf. on Machine L earning*, pages 249–256. Morgan Kaufmann, 1992.

[15] R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: a decision tree hybrid. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 202–207, 1996.

[16] R. Kohavi, P. Langley, and Y. Yun. The utility of feature weighting in nearest-neighbor algorithms. In M. van Someren and G. Widmer, editors, *Poster Papers: Ninth European Conf. on Machine Learning*, Prague, Czech Republic, 1997. Unpublished.

[17] M. Kubat, D. Flotzinger, and G. Pfurtscheller. Discovering patterns in EEG signals: Comparative study of a few methods. In *Proc. of the 1993 Europ. Conf. on Mach. Learn.*, pages 367–371. Springer-Verlag, 1993.

[18] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proc.. of the 10th Conf. on Uncertainty in Artificial Intelligence*, pages 399–406. Morgan Kaufmann, 1994.

[19] C. Nadeau and Yoshua Bengio. Inference for the generalization error. In *Advances in Neural Information Processing Systems 12*, pages 307–313. MIT Press, 1999.

[20] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[21] C. A. Ratanamahatana and D. Gunopulos. Feature selection for the naive Bayesian classifier using decision trees. *Applied Artificial Intelligence*, 17(5-6):475–487, 2003.

[22] M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of Relieff and RRelieff. *Mach. Learning*, 53(1-2):23–69, 2003.

[23] S. L. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:251–276, 1991.

[24] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the Assoc. for Computing Machinery*, 29:1213–1228, 1986.

[25] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.

[26] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

[27] H. Zhang and S. Sheng. Learning weighted naive Bayes with accurate ranking. In *Proc. of the 4th IEEE Int. Conf. on Data Mining*, pages 567–570, 2004.

[28] Zijian Zheng and Geoffrey I. Webb. Lazy learning of Bayesian rules. *Machine Learning*, 41(1):53–84, 2000.