# Naive Bayesian Classifier Committees

Zijian Zheng

School of Computing and Mathematics
Deakin University, Geelong, Victoria 3217, Australia
Email: zijian@deakin.edu.au

**Abstract.** The naive Bayesian classifier provides a very simple yet surprisingly accurate technique for machine learning. Some researchers have examined extensions to the naive Bayesian classifier that seek to further improve the accuracy. For example, a naive Bayesian tree approach generates a decision tree with one naive Bayesian classifier at each leaf. Another example is a constructive Bayesian classifier that eliminates attributes and constructs new attributes using Cartesian products of existing attributes. This paper proposes a simple, but effective approach for the same purpose. It generates a naive Bayesian classifier committee for a given classification task. Each member of the committee is a naive Bayesian classifier based on a subset of all the attributes available for the task. During the classification stage, the committee members vote to predict classes. Experiments across a wide variety of natural domains show that this method significantly increases the prediction accuracy of the naive Bayesian classifier on average. It performs better than the two approaches mentioned above in terms of higher prediction accuracy.

## 1 Introduction

Naive Bayesian classifier learning is based on Bayes' theorem and an attribute independence assumption (Duda and Hart 1973; Kononenko 1990; Langley and Sage 1994). Given training examples described using a vector of attribute values together with a known class for each example, the naive Bayesian classifier predicts the class of a new example $V = <v_1, v_2, \cdots, v_n>$ as the one with the highest probability of $C_i$ given $V$:

$$P(C_i|V) = \frac{P(C_i)\prod_j P(v_j|C_i)}{P(V)}. \tag{1}$$

This learning technique is simple and fast. It has been shown that in many domains the prediction accuracy of the naive Bayesian classifier compares surprisingly well with that of other more complex learning algorithms such as decision tree learning, rule learning, and instance-based learning algorithms (Cestnik, Kononenko, and Bratko 1987; Langley, Iba, and Thompson 1992; Domingos and Pazzani 1996). In addition, naive Bayesian classifier learning is robust to noise and irrelevant attributes. Some experts report that the learned theories are easy to understand (Kononenko 1993). However, when the strong attribute independence assumption is violated, which is very common, the performance of the naive Bayesian classifier can be poor.

A few techniques have been developed to improve the performance of the naive Bayesian classifier. Two examples are the constructive Bayesian classifier (BSEJ) (Pazzani 1996), and the naive Bayesian tree (NBTREE) approach (Kohavi 1996). It has been shown that it is possible to improve the naive Bayesian classifier, although Domingos and Pazzani (1996) argue that the naive Bayesian classifier is still in fact optimal when the independence assumption is violated so long as the ranks of the conditional probabilities of classes given an example are correct. The extent to which the above approaches improve upon the performance of the naive Bayesian classifier suggests that these ranks are in practice incorrect in a substantial number of cases.

Most existing techniques for improving the performance of the naive Bayesian classifier require complex induction processes. For example, NBTREE adopts a hybrid model of decision trees and naive Bayesian classifiers. Each leaf of such a tree contains a naive Bayesian classifier. BSEJ employs a wrapper model (John, Kohavi, and Pfleger 1994) with the leave-1-out cross-validation estimation to find the best Cartesian product attributes from existing nominal attributes for the naive Bayesian classifier (Pazzani 1996). It also considers deleting existing attributes. This paper proposes a simple method to improve naive Bayesian classifier learning. It is called the naive Bayesian classifier committee (NBC).

The idea of NBC is inspired by recent promising theoretical and empirical research results in boosting (Freund and Schapire 1996a, 1996b; Quinlan 1996; Schapire, Freund, Bartlett, and Lee 1997). Boosting induces multiple individual classifiers in sequential trials. At the end of each trial, instance weights are adjusted to reflect the importance of each training example for the next induction trial. The objective of the adjustment is to increase the weights of misclassified training examples. Change of instance weights causes the learner to concentrate on different training examples in different trials,[1] thus resulting in different classifiers. Finally, the individual classifiers are combined through voting to form a composite classifier. Quinlan (1996) shows that boosting can significantly increase the prediction accuracy of decision tree learning.

We implemented a boosting algorithm for naive Bayesian classifier using a similar method to that for boosting decision trees (Quinlan 1996). Although the algorithm achieves higher accuracy than the naive Bayesian classifier in some domains, the overall accuracy improvement over the naive Bayesian classifier in a large set of natural domains is very marginal. The reason might be that boosting implicitly requires the instability of the boosted learning systems (Quinlan 1996). Naive Bayesian classifier learning is more stable than decision tree learning. A small change to the training set will have little impact on a naive Bayesian classifier. On the other hand, naive Bayesian classifier learning is not stable in the sense that a small change to the attribute set could lead to very different classifiers. Moreover, due to the attribute independence assumption, a naive Bayesian classifier built on an attribute subset might perform better than a

---

[1] This can be implemented by either changing the weights of training examples directly if the learner can handle it, or drawing a succession of independent bootstrap samples from the original training set.

naive Bayesian classifier created using all attributes (Langley and Sage 1994). Therefore, generating naive Bayesian classifier committees could be an approach to improving the performance of the naive Bayesian classifier. In the committee, each member is a naive Bayesian classifier built using a subset of attributes. The final class prediction is made through committee voting.

## 2 The NBC Algorithm

Table 1 details the naive Bayesian classifier committee learning algorithm, NBC. The idea is to generate a set of naive Bayesian classifiers in sequential trials to form a committee. Each naive Bayesian classifier is based on a different subset of attributes. All committee members make the final decision through voting when classifying examples. The estimated performance of a naive Bayesian classifier is used to guide the formation of the attribute subset for creating the naive Bayesian classifier in the following trial.

Leave-1-out cross-validation is used to estimate the error rates of naive Bayesian classifiers, since the leave-1-out cross-validation error rate is a better estimate than the resubstitution error rate (Breiman, Friedman, Olshen, and Stone 1984). In addition, for a naive Bayesian classifier, the operations of removing and adding an example are very easy and efficient. At the beginning, NBC builds a naive Bayesian classifier (called $NB_{base}$) using all attributes. Its leave-1-out cross-validation error rate is used as the reference for performance comparison when generating the committee.

To decide how to choose a subset of attributes for building a naive Bayesian classifier in a trial, NBC maintains a probability vector $P$ with one element for each attribute. Each trial starts from sampling an attribute subset from the set of all attributes using $P$. The attribute $a$ has the probability $P[a]$ of being selected. Given a learning task, we usually do not know which attributes can be used to built a good naive Bayesian classifier. NBC just initialises each $P[a]$ with 0.5. The idea is to let each attribute has 50% probability of being chosen at the beginning. Therefore, the subset contains about a half of all attributes.

After the attribute subset is created, NBC does not need to do any calculation to build the naive Bayesian classifier using this attribute subset, since all necessary probabilities and conditional probabilities are already available from the generation of the naive Bayesian classifier based on all attributes. The naive Bayesian classifier resulted from each trial only needs to maintain its attribute subset. However, to decide whether this naive Bayesian classifier is accepted as a committee member, it is evaluated using leave-1-out cross-validation on the training set. If its error rate $\epsilon_t$ is lower than the error rate of $NB_{base}$, $\epsilon_{NB_{base}}$, it is accepted. Otherwise, it is discarded.

At the end of each trial, $P[a]$ for each attribute in the subset of the current trial $t$ is modified by multiplying the value $1/\beta_t$ which is defined in Equation 2. Note that $\beta_t < 1$, if $\epsilon_t < \epsilon_{NB_{base}}$; and $\beta_t > 1$, if $\epsilon_t > \epsilon_{NB_{base}}$. The objective of this modification is that the probabilities that the attributes used in this trial will be selected in the next trial should be increased, if the naive Bayesian classifier built

**Table 1.** The naive Bayesian classifier committee learning algorithm

```
NBC($Att, D_{training}, T$)
    INPUT: $Att$: a set of attributes,
           $D_{training}$: a set of training examples described
               using $Att$ and classes,
           $T$: the number of trials for generating the committee
               with the number of attributes, $N$, as its default value.
    OUTPUT: a naive Bayesian committee.
Build a naive Bayesian classifier using $Att$ and $D_{training}$, called $NB_{base}$
$\epsilon_{NB_{base}}$ = Leave-1-out-Evaluation($NB_{base}, D_{training}$)
Add $NB_{base}$ into Committee as the first member which uses all attributes,
    that is, $NB_0 = NB_{base}$
$MaxT = 10 \times T$
Initialise $P[a] = 0.5$ for each attribute $a$ in $Att$
$l = 1$
$t = 1$
WHILE ($t <= T$ and $l <= MaxT$)
{       $Att_{subset}$ = Sample attributes from $Att$ based on $P$
        $NB_{temp}$ = Build a naive Bayesian classifier using $Att_{subset}$
        $\epsilon_t$ =Leave-1-out-Evaluation($NB_{temp}, D_{training}$)
        $\alpha_t = (\epsilon_t - \epsilon_{NB_{base}} + 1)/2$
        $\beta_t = \alpha_t/(1 - \alpha_t)$
        FOR each attribute $a$ in $Att_{subset}$
            $P[a] = P[a]/\beta_t$
        Normalise $P$ such that $\sum_{a=1}^{N} P[a] = 0.5N$
        IF ($\beta_t < 1$)
        { $NB_t = NB_{temp}$
          $t = t + 1$
        }
        $l = l + 1$
}
$T = t - 1$
RETURN the naive Bayesian committee containing $NB_t, t = 0, 1, \cdots, T$
```

in this trial performs better than $NB_{base}$. They should be decreased otherwise. After the modification, the probabilities of all attributes are normalised such that their sum is equal to 0.5 times the number of all attributes. This makes the attribute subset in the following trial also contain about a half of all attributes.

$$\beta_t = \alpha_t/(1 - \alpha_t), \quad \text{where } \alpha_t = (\epsilon_t - \epsilon_{NB_{base}} + 1)/2 \tag{2}$$

NBC generates $T$ naive Bayesian classifiers using different attribute subsets and put them into the committee. $T$ is equal to the number of all attributes by default. To make NBC efficient in practice, NBC is set to carry out at most

$10 \times T$ trials, even if too many naive Bayesian classifiers which have higher error rates than $NB_{base}$ are created. In this situation, the committee may contain less than $T$ naive Bayesian classifiers. To avoid the extreme situation where no naive Bayesian classifier better than $NB_{base}$ is created in any trial, NBC includes $NB_{base}$ in the committee. Therefore, the committee always has at least one member at the end. It usually contains $T + 1$ naive Bayesian classifiers.

To classify a new example, each naive Bayesian classifier in the committee is invoked to produce the probability that this example belongs to each possible class. For each class, the probabilities provided by all committee members are summed up. The class with the largest summed probability wins the vote, and is used as the predicted class for this example. Ties are broken randomly.

## 3 Experiments

In this section, we use experiments in natural domains to study the performance of the naive Bayesian classifier committee learning algorithm by comparing with a naive Bayesian classifier learning algorithm, NB. Note that classifiers generated by NB are identical to the naive Bayesian classifiers created on all attributes in NBC. The performance measure used here is the error rate on the test set (unseen cases). In addition, the computational requirement of NBC is addressed.

### 3.1 Experimental Domains and Methods

Twenty-nine natural domains are used in the experiments. They include all the domains used by Domingos and Pazzani (1996) for studying the naive Bayesian classifier. These twenty-nine domains cover a wide variety of different domains and all are available from the UCI machine learning repository (Merz and Murphy 1997).

In each domain, two stratified 10-fold cross-validations (10-CV) (Breiman *et al.* 1984; Kohavi 1995) are performed for each algorithm. A 10-CV is carried out by randomly splitting the data set into 10 subsets that have similar size and class distribution. For each subset in turn, an algorithm is run using the examples in the remaining nine subsets as a training set and tested on the unseen examples in the hold-out subset. All the algorithms are run with their default option settings on the same training and test set partitions in every domain. An error rate reported in this paper is an average of the 20 trials for an algorithm.

Since the current implementations of the NBC and NB algorithms can only deal with nominal attributes, continuous attributes are discretized as a preprocess in the experiments. An entropy-based discretization method (Fayyad and Irani 1993; Ting 1994) is used. For each pair of training set and test set, the test set is discretized by using cut points found from the training set.

### 3.2 Experimental Results

Table 2 shows error rates of NBC and NB. In the column headed "NBC", boldface font indicates that the error rate of NBC is lower than that of NB at a

**Table 2.** Comparison of the error rates (%) of NBC and NB

| Domain | NB | NBC | NBC / NB | NBC − NB |
|---|---|---|---|---|
| Annealing | 2.78 | 2.73 | .98 | -0.05 |
| Audiology | 23.19 | **17.89** | .77 | -5.30 |
| Breast (W) | 2.65 | 2.72 | 1.03 | 0.07 |
| Chess (KR-KP) | 12.20 | **6.07** | .50 | -6.13 |
| Credit (Aust) | 13.98 | 13.91 | .99 | -0.07 |
| Echocardiogram | 28.95 | 30.49 | 1.05 | 1.54 |
| Glass | 30.91 | 31.59 | 1.02 | 0.68 |
| Heart (C) | 16.82 | 17.32 | 1.03 | 0.50 |
| Hepatitis | 14.19 | 14.50 | 1.02 | 0.31 |
| Horse colic | 20.79 | **16.58** | .80 | -4.21 |
| House votes 84 | 9.75 | **7.69** | .79 | -2.06 |
| Hypothyroid | 1.69 | **1.50** | .89 | -0.19 |
| Iris | 6.33 | 5.33 | .84 | -1.00 |
| Labor | 9.00 | 8.00 | .89 | -1.00 |
| LED-24 | 34.25 | **30.00** | .88 | -4.25 |
| Liver disorders | 35.08 | 34.48 | .98 | -0.60 |
| Lung cancer | 47.08 | 50.83 | 1.08 | 3.75 |
| Lymphography | 16.12 | 17.88 | 1.11 | 1.76 |
| Pima | 25.20 | 24.42 | .97 | -0.78 |
| Postoperative | 33.89 | 30.55 | .90 | -3.34 |
| Primary tumor | 50.91 | 52.66 | 1.03 | 1.75 |
| Promoters | 9.27 | 8.54 | .92 | -0.73 |
| Solar flare | 19.44 | **16.34** | .84 | -3.10 |
| Sonar | 23.55 | 24.55 | 1.04 | 1.00 |
| Soybean | 9.16 | 8.65 | .94 | -0.51 |
| Splice junction | 4.38 | **3.94** | .90 | -0.44 |
| Tic-Tac-Toe | 30.64 | 29.70 | .97 | -0.94 |
| Wine | 2.22 | 2.22 | 1.00 | 0.00 |
| Zoology | 5.45 | 4.00 | .73 | -1.45 |
| average | 18.62 | 17.76 | .93 | -0.85 |
| w/t/l | | | | 20/1/9 |
| significance level | | | | .0436 |

significance level better than 0.05 using a two-tailed pairwise t-test (Chatfield 1978) on the results of the 20 trials in a domain. The error rate ratios and differences of NBC and NB are also included in the table. A ratio less than 1.00 or a difference less than 0.00 means that NBC has lower error rate than NB. The line headed "w/t/l" shows the "won-tied-lost" record in the 29 domains, that is, the number of domains in which NBC has lower, the same, and higher error rates than NB.

From Table 2, the significant advantage of NBC over NB in terms of lower error rate can be clearly seen. On average over the 29 domains, NBC reduces the error rate of NB by 7%. The one-tailed pairwise sign test (Chatfield 1978) on the error rates of NBC and NB in the 29 domains shows that NBC is more accurate than NB at a significance level of 0.0436 (see the last line of the table). In 8 out of these 29 domains, NBC achieves significantly lower error rates than NB. NBC does not obtain any significantly higher error rates than NB in all of these domains. If ignoring the differences between NBC and NB that are not significant (only considering the significant differences), the one-tailed pairwise sign test shows that NBC is significantly more accurate than NB at a level of 0.0039 in these domains.

Since the class prediction of NBC relies on the voting of the naive Bayesian classifier committee, it is interesting to know the effect of the committee size on the performance of NBC. Figure 1 depicts the error rate of NBC as a function of $T$ in the Chess (KR-KP) and Splice junction domains, the two largest domains in the test suite. For each value of $T$, the same experimental method described above is used except that $T$ uses the given value instead of the default one. The error rates of NB, as a reference, are also given in the figure. Since $T$ has nothing to do with NB, the error rates of NB are always the same when $T$ changes.

We can see, from Figure 1, that NBC always has significantly lower error rates than NB in the Chess (KR-KP) domain as $T$ changes. NBC has significantly lower error rates than NB for most values of $T$ in the Splice junction domain. Only when $T$ is equal to 20 (the first point), does NBC has the same error rate as NB in this domain. In the Chess (KR-KP) domain, NBC achieves the lowest error rate when $T$ is 40. This value is close to the number of attributes of this domain, which is 36. In the Splice junction domain, NBC has the lowest error rate at the point 180. It is 3 times the number of attributes of this domain, which is 60. However, the error rate of NBC for $T$ with the value 60 is not significantly higher than this lowest error rate. Therefore, using the number of attributes as the default value of $T$ is reasonable, although it might not be optimum in some domains.

### 3.3 Computational Requirement

NBC is slower than NB. However, the extra cost of NBC for generating each committee member is creating an attribute subset and performing a leave-1-out cross-validation evaluation. Therefore, NBC's time complexity is $O(m \cdot n \cdot T)$,[2] while NB's time complexity is $O(m \cdot n)$, where $m$ is the size of the training set, $n$ is the number of attributes. Since $T$ is equal to $n$ by default, NBC's time complexity is $O(m \cdot n^2)$. Note that $n$ is usually much smaller than $m$.

Figure 2 shows the execution time of NBC (including both learning and classification stages) as a function of $T$ in the Chess (KR-KP) and Splice junction domains. The time is measured using CPU second on a SUN SPARCstation 5. The experimental method is exactly the same as that for drawing the learning curves above. These two curves indicate that the computational requirement of NBC is linear in $T$.

---

[2] The time complexity of NBC is $O(m \cdot n + T \cdot (n + m \cdot n)) = O(m \cdot n \cdot T)$.

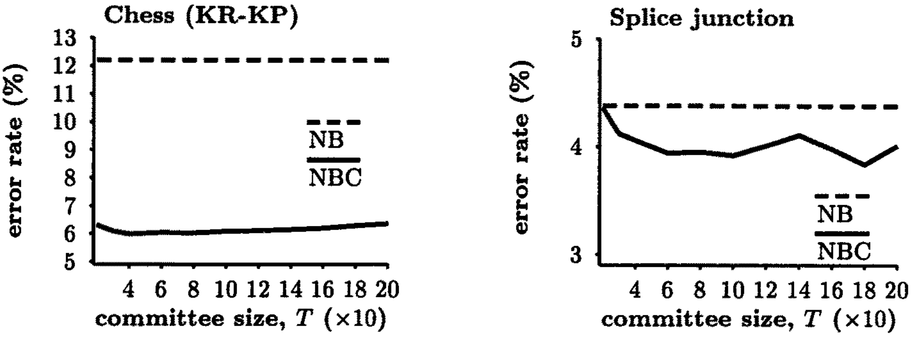

**Fig. 1.** Effect of $T$ on the performance of NBC

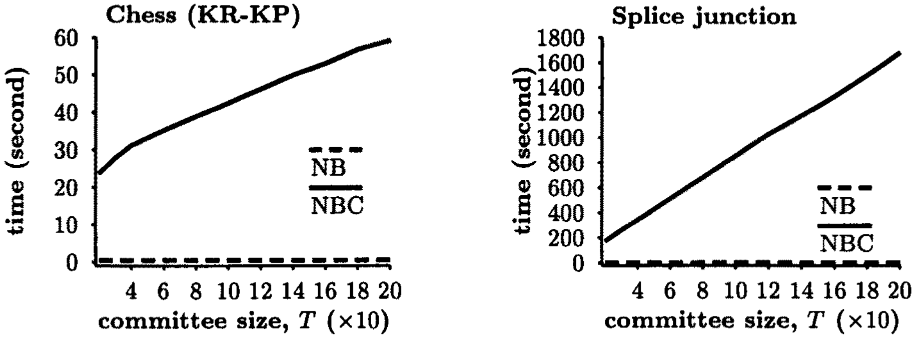

**Fig. 2.** Computational requirement of NBC as a function of $T$

Figure 3 depicts the execution time of NBC (including both learning and classification stages) as a function of training set size in the Chess (KR-KP) and Splice junction domains. $T$ uses its default value (the number of attributes) in this experiment. Each point of a learning curve is an average value over 20 trials. For each trial, the training set used at every point is a randomly selected subset of the training set used in the corresponding trial of the two 10-fold cross-validations on the entire dataset of the domain. In each trial, the training set at a point is a proper subset of the training set at the next adjacent point. The test set at every point of a trial is the same as the test set used in the corresponding trial of the two 10-fold cross-validations.

This figure clearly shows that the computational requirement of NBC is linear in training set size, although the time for NBC increases faster than that for NB as the training set size increases.

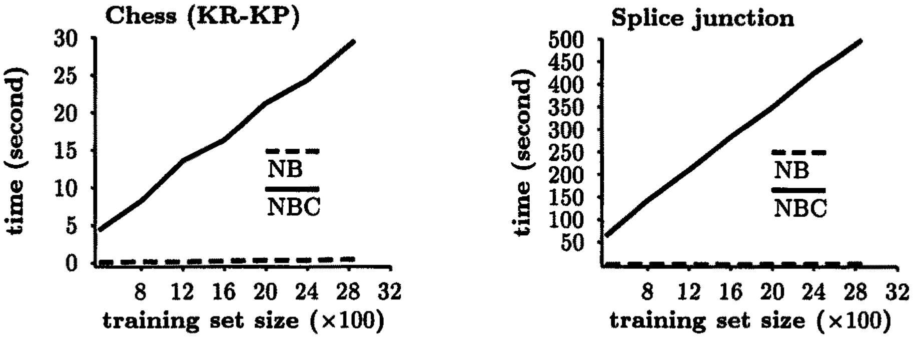

**Fig. 3.** Computational requirement of NBC as a function of training set size

## 4 Discussion

Since the constructive naive Bayesian classifier BSEJ (Pazzani 1996) and the naive Bayesian tree learning algorithm NBTREE (Kohavi 1996) also intend to improve the performance of naive Bayesian classifier learning, it is interesting to compare NBC with them. We implemented BSEJ and NBTREE based on Pazzani (1996) and Kohavi (1996) respectively. Note that the naive Bayesian classifier NB is used in all the implementations of BSEJ, NBTREE, and NBC. Our experiments with these two algorithms using the same experimental method described in the previous section show that BSEJ and NBTREE achieve the average error rates 18.10% and 17.90% respectively in these 29 domains. The one-tailed pairwise sign test indicates that neither the error rate reduction of BSEJ nor the error rate reduction of NBTREE over NB is significant at the level 0.05, while the error rate reduction of NBC over NB is significant. It has been found that both BSEJ and NBTREE obtain significantly higher error rates than NB in two out of the 29 domains. The average error rates of BSEJ and NBTREE are 2% and 1% higher than that of NBC in the 29 domains respectively. These results suggest that NBC performs better than the constructive naive Bayesian classifier and the naive Bayesian tree learning method in terms of lower error rate.

During the classification stage, for each example to be classified, NBC sums up the class distributions produced by all committee members. The class with the highest summed probability is used as the predicted class of the example. An alternative method is the majority vote. Instead of the class distribution, each committee member provides a predicted class for a given example. Then, NBC classifies the example as the class with the support from the largest number of committee members. With this voting technique, the average error rate of NBC over the 29 domains is 17.75%, very close to the error rate of the method described in Section 2.

Another issue is voting weights for classification. NBC does not use weights for committee voting. In other words, the vote of each committee member in

NBC is worth 1 unit. One might think that weighted voting may further improve the performance of NBC. Unfortunately, although this issue is worthy of further investigation, our preliminary exploration has not found any appropriate weighting method that can reduce the error rate of the current NBC on average. For example, NBC with $\log(1/\beta_t)$ as the weight of the naive Bayesian classifier generated in trail $t$ obtains an average error rate of 17.80% in the 29 domains. It is slightly higher than the error rate of NBC without voting weights, although the difference is not significant. It remains an open question whether an appropriate weighting method can be designed to significantly reduce the average error rate of NBC.

## 5 Related Work

From the point of view of improving the performance of naive Bayesian classifier learning, the work related to NBC includes the constructive Bayesian classifier (BSEJ) (Pazzani 1996) and the naive Bayesian tree (NBTREE) approach (Kohavi 1996) mentioned in the introduction section, as well as the semi-naive Bayesian classifier (Kononenko 1991) and the attribute deletion technique (Langley and Sage 1994). Kononenko's semi-naive Bayesian classifier performs exhaustive search to iteratively join pairs of attribute values (Kononenko 1991). The aim is to optimise the tradeoff between the "non-naivety" and the reliability of estimates of probabilities. Langley and Sage (1994) have shown that attribute deletion can improve the performance of the naive Bayesian classifier when attributes are inter-dependent, especially when some attributes are redundant.

The investigation of the naive Bayesian classifier committee learning method is motivated by recent research on boosting (Freund and Schapire 1996a, 1996b; Quinlan 1996; Schapire *et al.* 1997). Another related approach is bagging (Breiman 1996; Quinlan 1996). Bagging builds a set of classifiers with each using a separately sampled training set (with replacement) of the same size from the original training set. Final classification is also through voting among all of these classifiers. Both boosting and bagging generate different classifiers by deriving different training sets from the original one, while NBC creates different classifiers by deriving different attribute subsets. Boosting and bagging have been applied on weak learning algorithms with great success, such as decision tree learning (Quinlan 1996). No published research has been seen so far on applying boosting or classifier committee techniques to naive Bayesian classifier learning. No effort has been made to explore approaches to generating, as a composite classifier, a set of classifiers using different attribute subsets.

When generating a naive Bayesian classifier as a committee member, NBC chooses an attribute subset based on the probability distribution of all attributes that are resulted from the performance of naive Bayesian classifiers created in the previous trials. Attribute subset selection has been studied for a while for classification learning (e.g. Almuallim and Dietterich (1992), Kira and Rendell (1992), John *et al.* (1994), and Langley (1994)). However, all of these existing methods choose an attribute subset to build a single classifier, while NBC generates a set of classifiers with each based a different attribute subset.

## 6 Conclusions and Future Work

This paper presented a method of generating naive Bayesian classifier committees by building individual naive Bayesian classifiers using different attribute subsets in sequential trials. During classification stage, the committees make the class prediction through voting. In the current implementation of the NBC algorithm, no weights are used for voting. Appropriate weighting techniques may further improve the performance of NBC. NBC chooses about a half of attributes, to create a naive Bayesian classifier, using the probability distribution of all attributes, which is built up based on the performance of naive Bayesian classifiers generated previously. Other approaches to attribute subset selection for this purpose are worth exploring.

The experimental study in a wide variety of natural domains shows that the naive Bayesian classifier committee learning can significantly increase the prediction accuracy of naive Bayesian classifier learning on average. It performs better, on average, than the naive Bayesian tree learning and the constructive naive Bayesian classifier learning in the set of domains under investigation.

## Acknowledgements

The author is grateful to Geoffrey Webb for his helpful comments on this research and earlier drafts of the paper. Many thanks to Kai Ming Ting for supplying the discretization program.

## References

Almuallim, H. and Dietterich, T.G.: Efficient algorithms for identifying relevant features. *Proceedings of the 9th Canadian Conference on Artificial Intelligence.* Vancouver, BC: Morgan Kaufmann (1992) 38-45.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J.: *Classification And Regression Trees.* Belmont, CA: Wadsworth (1984).

Breiman, L.: Bagging predictors. *Machine Learning.* **24** (1996) 123-140.

Cestnik, B., Kononenko, I., and Bratko, I.: ASSISTANT 86: A knowledge-elicitation tool for sophisticated users. In I. Bratko & N. Lavrač (Eds.), *Progress in Machine Learning – Proceedings of the 2nd European Working Session on Learning (EWSL87).* Wilmslow, UK: Sigma Press (1987) 31-45.

Chatfield, C.: *Statistics for Technology: A Course in Applied Statistics.* London: Chapman and Hall (1978).

Domingos, P. and Pazzani, M.: Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Proceedings of the 13th International Conference on Machine Learning.* San Francisco, CA: Morgan Kaufmann (1996) 105-112.

Duda, R.O. and Hart, P.E.: *Pattern Classification and Scene Analysis.* New York: John Wiley (1973).

Fayyad, U.M. and Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann (1993) 1022-1027.

Freund, Y. and Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Unpublished manuscript, available from the authors' home pages ("http://www.research.att.com/{~yoav,~schapire}") (1996a).

Freund, Y. and Schapire, R.E.: Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning.* Morgan Kaufmann (1996b) 148-156.

John, G.H., Kohavi, R., and Pfleger, K.: Irrelevant features and the subset selection problem. *Proceedings of the 11th International Conference on Machine Learning.* San Francisco, CA: Morgan Kaufmann (1994) 121-129.

Kira, K. and Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. *Proceedings of the 10th National Conference on Artificial Intelligence.* Menlo Park, CA: AAAI Press/Cambridge, MA: MIT Press (1992) 129-134.

Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence.* San Mateo, CA: Morgan Kaufmann (1995) 1137-1143.

Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining.* Menlo Park, CA: The AAAI Press (1996) 202-207.

Kononenko, I.: Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga *et al.* (Eds.), *Current Trends in Knowledge Acquisition.* Amsterdam: IOS Press (1990).

Kononenko, I.: Semi-naive Bayesian classifier. *Proceedings of European Conference on Artificial Intelligence* (1991) 206-219.

Langley, P., Iba, W.F., and Thompson, K.: An analysis of Bayesian classifiers. *Proceedings of the 10th National Conference on Artificial Intelligence.* Menlo Park, CA: The AAAI Press (1992) 223-228.

Langley, P.: Selection of relevant features in machine learning. *Proceeding of the AAAI Fall Symposium on Relevance*, New Orleans, LA: The AAAI Press (1994).

Langley, P. and Sage, S.: Induction of selective Bayesian classifiers. *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence.* Seattle, WA: Morgan Kaufmann (1994) 339-406.

Merz, C.J. and Murphy, P.M.: 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 (1997).

Pazzani, M.J.: Constructive induction of Cartesian product attributes. *Proceedings of the Conference, ISIS'96: Information, Statistics and Induction in Science.* Singapore: World Scientific (1996) 66-77.

Quinlan, J.R.: Bagging, boosting, and C4.5. *Proceedings of the 13th National Conference on Artificial Intelligence*, Menlo Park: The AAAI Press (1996) 725-730.

Schapire, R.E., Freund, Y., Bartlett, P., and Lee W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. *Proceedings of the 11th International Conference on Machine Learning.* Morgan Kaufmann (1997) 322-330.

Ting, K.M.: Discretization of continuous-valued attributes and instance-based learning (Technical Report 491). Sydney, Australia: University of Sydney, Basser Department of Computer Science (1994).