Pre-publication draft of a paper which appeared in the Proceedings of the Tenth Australian Joint Conference on Artificial Intelligence AI'98 pp 285-295 Publisher: Springer-Verlag

Adjusted Probability Naive Bayesian Induction

Geoffrey I. Webb[1] and Michael J. Pazzani[2]

[1] School of Computing and Mathematics
Deakin University
Geelong, Vic, 3217, Australia.
[2] Department of Information and Computer Science
University of California, Irvine
Irvine, Ca, 92717, USA.

Abstract. Naive Bayesian classifiers utilise a simple mathematical model for induction. While it is known that the assumptions on which this model is based are frequently violated, the predictive accuracy obtained in discriminate classification tasks is surprisingly competitive in comparison to more complex induction techniques. Adjusted probability naive Bayesian induction adds a simple extension to the naive Bayesian classifier. A numeric weight is inferred for each class. During discriminate classification, the naive Bayesian probability of a class is multiplied by its weight to obtain an adjusted value. The use of this adjusted value in place of the naive Bayesian probability is shown to significantly improve predictive accuracy.

1 Introduction

The naive Bayesian classifier (Duda & Hart, 1973) provides a simple approach to discriminate classification learning that has demonstrated competitive predictive accuracy on a range of learning tasks (Clark & Niblett, 1989; Langley, P., Iba, W., & Thompson, 1992). The naive Bayesian classifier is also attractive as it has an explicit and sound theoretical basis which guarantees optimal induction given a set of explicit assumptions. There is a drawback, however, in that it is known that some of these assumptions will be violated in many induction scenarios. In particular, one key assumption that is frequently violated is that the attributes are independent with respect to the class variable. The naive Bayesian classifier has been shown to be remarkably robust in the face of many such violations of its underlying assumptions (Domingos & Pazzani, 1996). However, further improvements in performance have been demonstrated by a number of approaches, collectively called *semi-naive Bayesian classifiers*, that seek to adjust the naive Bayesian classifier to remedy violations of its assumptions. Previous semi-naive Bayesian techniques can be broadly classified into two groups, those that manipulate the attributes to be employed prior to application of naive Bayesian induction (Kononenko, 1991; Langley & Sage, 1994; Pazzani, 1996) and those that select subsets of the training examples prior to the application of naive Bayesian classification of an individual case (Kohavi, 1996; Langley, 1993).

This paper presents an alternative approach that seeks instead to adjust the probabilities produced by a standard naive Bayesian classifier in order to accommodate violations of the assumptions on which it is founded.

2 Adjusted Probability Semi-Naive Bayesian Induction

The naive Bayesian classifier is used to infer the probability that an object j, described by attribute values $A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j}$ belongs to a class C_i. It uses Bayes theorem

$$P(C_i \mid A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j}){=}\frac{P(C_i)P(A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j} \mid C_i)}{P(A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j})} \quad (1)$$

where $P(C_i \mid A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j})$ is the conditional probability of the class C_i given the object description; $P(C_i)$ is the prior probability of class C_i; $P(A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j} \mid C_i)$ is the conditional probability of the object description given the class C_i; and $P(A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j})$ is the prior probability of the object description.

Based on an assumption of attribute conditional independence, this is estimated using

$$\frac{P(C_i)\prod_k P(A_k{=}V_{k_j} \mid C_i)}{P(A_1{=}V_{1_j} \wedge \ldots \wedge A_n{=}V_{n_j})}. \quad (2)$$

Each of the probabilities within the denumerator of (2) are in turn inferred from the relative frequencies of the corresponding elements in the training data. Where discriminate prediction of a single class is required, rather than assigning explicit probabilities to each class, the class is chosen with the highest probability (or with the lowest misclassification risk, if classes are further differentiated by having associated misclassification costs). In this context, the denominator can be omitted from (2) as it does not affect the relative ordering of the classes.

Many violations of the assumptions that underlie naive Bayesian classifiers will result in systematic distortion of the probabilities that the classifier outputs. For example, take a simple two attribute learning task where the attributes A and B and class C all have domains $\{0, 1\}$, for all objects $A{=}B$, the probability of each value of each attribute is 0.5, $P(C{=}0 \mid A{=}0) = 0.75$, and $P(C{=}0 \mid A{=}1) = 0.25$. Given an object $A{=}0, B{=}0$, and perfect estimates of all values within (2),[3] the inferred probability of class $C = 0$ will be 0.5625 and of class $C{=}1$ will be 0.0625. The reason that the class probability estimates are incorrect is that the two attributes violate the independence assumption. In this simple example, the systematic distortion in estimated class probabilities could be corrected by taking the square root of all naive Bayesian class probability estimates.

It is clear that in many cases there will exist functions from the naive Bayesian estimates to the true conditional class probabilities. However, the nature of these

[3] $P(C = 0) = 0.5$, $P(A = 0 \mid C = 0) = 0.75$, $P(B = 0 \mid C = 0) = 0.75$, $P(C = 1) = 0.5$, $P(A = 0 \mid C = 1) = 0.25$, $P(B = 0 \mid C = 1) = 0.25$, and $P(A = 0 \wedge B = 0) = 0.5$.

functions will vary depending upon the type and complexity of the violations of the assumptions of the naive Bayesian approach.

Where a single discrete class prediction is required rather than probabilistic class prediction, it is not even necessary to derive correct class probabilities. Rather, all that is required is to derive values for each class probability such that the most probable class (or class with the lowest misclassification risk) has the highest value. In the two class case, if it is assumed that the inferred values are monotonic with respect to the correct probabilities, all that is required is identification of the inferred value at which the true probability (or misclassification risk) of one class exceeds that of the other.

For example, Domingos & Pazzani (1996) show that the naive Bayesian classifier makes systematic errors on some m-of-n concepts. To illustrate their analysis of this problem, assume that the naive Bayesian classifier is trained with all 2^6 examples of an at-least-2-of-6 concept. This is a classification task for which the class C equals 1 when any two or more of the six binary attributes equal 1. In this case, $P(C{=}1) = 57/64$, $P(C{=}0) = 7/64$, $P(A_k{=}1\,|\,C{=}1) = 31/57$, $P(A_k{=}0\,|\,C{=}1) = 26/57$, $P(A_k{=}1\,|\,C{=}0) = 1/7$ and $P(A_k{=}0\,|\,C{=}0) = 6/7$. Therefore, the naive Bayesian classifier will classify as positive an example for which i attributes equal 1 if

$$\left(\frac{57}{64} * \frac{31}{57}^{i} * \frac{26}{57}^{6-i}\right) > \left(\frac{7}{64} * \frac{1}{7}^{i} * \frac{6}{7}^{6-i}\right). \tag{3}$$

However, this condition is false only for $i = 0$ while the at-least-2-of-6 concept is false for $i < 2$. Note however that both the terms in (3) are monotonic with respect to i, the left-hand-side increasing while the right-hand-side decreases as i increases. Therefore, by multiplying the left-hand-side of (3) by a constant adjustment factor $a : 0.106 < a < 0.758$ we have a function of i that perfectly discriminates positive from negative examples[4]. Care must be taken to avoid using this as probability estimate, but this additional degree of freedom will allow the naive Bayesian classifier to discriminate well on a broader class of problems. For multi-class problems, and problems where the inferred probabilities are not monotonic with respect to the true probabilities, more complex adjustments are required.

This paper presents an approach that attempts to identify and apply linear adjustments to the class probabilities. To this end, an adjustment factor is associated with each class, and the inferred probability for a class is multiplied by the corresponding factor. While it is acknowledged that such simple linear adjustments will not capture the finer detail of the distortions in inferred probabilities in all domains, it is expected that they will frequently assist in assigning more useful probabilities in contexts where discrete single class prediction is required (as it will enable the probability for a class to be boosted above that of the other classes, enabling correct class selection irrespective of accurate probability assignment). The general approach of inferring a function to adjust the class

[4] The lower limit on a is the lowest value at which (3) is true for $i = 2$. The upper limit is the highest value at which (3) is false for $i = 1$.

probabilities obtained through naive Bayesian induction will be referred to as *adjusted probability naive Bayesian classification* (APNBC). This paper restricts itself to considering simple linear adjustments to the inferred probabilities, although it is noted that any other class of functions could be considered in place of simple linear adjustments. We do not believe that linear adjustments are likely to lead to more accurate classifiers than alternative classes of adjustment function. However, linear adjustments do have one advantage over many alternatives, that plausible adjustment factors are relatively inexpensive to compute.

3 The APNBC Technique

Due to the simplicity of naive Bayesian classification and of APNBC, there is relatively low risk of overfitting inferred models to a set of training data (variance is low). For this reason, appropriate adjustments will be directly inferred from resubstitution performance (the performance of the modified classifier on the training data), rather than using a variance management strategy such as estimation by cross validation.

In the two class case, it is necessary only to find an adjustment value for one of the classes. This is because for any combination of adjustments A_1 and A_2 for the classes C_1 and C_2, the same effect will be obtained by setting the adjustment for C_1 to $\frac{A_1}{A_2}$ and the adjustment for C_2 to 1. For this reason, in the two class case, the adjustment for one class is set to 1 and the APNBC technique considers only adjustments to the other class, seeking an adjustment value that maximizes resubstitution accuracy.

In the multiple class case, the search for suitable adjustments is more complex, as the adjustment for one class will greatly affect the appropriate adjustments for other classes. In this context a simple hill-climbing search is employed. All adjustment values are initialized to 1, and a single adjustment that maximizes resubstitution accuracy is found. If a suitable adjustment is found, it is incorporated into the classifier and the process repeated until no suitable adjustments are obtained.

Adjustments are continuous values, and hence the search space of possible adjustments is infinite. However, critical values, in terms of resubstitution performance, are defined by the objects in the training set. If an object o of class i is misclassified as class j by APNBC with the current vector of adjustments A, a tie between classifications for the object will result if A_i is assigned $\frac{A_j P(o,j)}{P(o,i)}$, where $P(o,x)$ is the probability inferred by the naive Bayesian classifier that o belongs to class x. A tie may also result if A_j is assigned $\frac{A_i P(o,i)}{P(o,j)}$, but this will also depend upon the adjusted probability for i being greater than the adjusted probability for any other class. To resolve such ties during the search for a set of values for A, the APNBC induction algorithm employs the critical value for A_i or A_j plus or minus a small value (10^{-5}), as appropriate.

When the search for a set of adjustment values is complete, each selected adjustment is replaced by the midpoint between the two critical values that bound it. This latter step is delayed in this manner solely for reasons of computational

efficiency. It is possible that the hill-climbing search algorithm will select at different stages a number of different adjustment values for any one class, in which case it is desirable to delay the computationally expensive task of identifying the second critical bound on the adjustment until the final adjustment region has been selected.

When two possible adjustments tie for first place with respect to reduction in resubstitution error, the smaller adjustment is selected. This represents a slight inductive bias toward minimizing the degree to which the adjusted probabilities differ from those inferred by the naive Bayesian classifier.

While it is argued that the APNBC approach has low risk of overfitting due to the simplicity of the models that it employs, there is nonetheless some risk of overfitting that might profitably be managed. To this end, before accepting an adjustment, a binomial sign test is performed to determine the probability that the observed improvement in resubstitution accuracy could be obtained by chance. If this probability is greater than a predefined critical value, α, the adjustment is not adopted.

An algorithm for multiclass induction is presented in Appendix A. For two class induction, it is necessary only to pass once through the main loop, and necessary only to examine either upward or downward adjustments, as in the two class case for every upward adjustment for one class there is an equivalent downward adjustment for the other, and vice versa.

The worst case computational complexity of the induction of each adjustment is of order $O(CN^2)$, where C is the number of classes and N is the number of cases. The process is repeated once for each class. For each misclassified case belonging to the class (which in the worst case is proportional to the number of cases), possible adjustments are evaluated. Each such evaluation requires examining each case to consider its reclassification. This does not require recalculation of the raw naive Bayesian probabilities, however, as these can be calculated once only in advance.

In our observation, never has a second adjustment been inferred for a single class, although we do not see an obstacle to this happening in theory (that is, an adjustment is made for class a which then enables an adjustment to be made for class b which in turn allows a different adjustment to be made for class a).

Given that the number of adjustments inferred is usually lower than the number of classes, $O(C^2N^2)$ appears a plausible upper bound on the average case complexity of the algorithm.

4 Experimental Evaluation

The APNBC induction algorithm was implemented in C. This implementation estimates the prior probability of class i ($P(C_i)$) by $\frac{n_i+1}{m+c}$ where n_i is the number of training objects belonging to i, m is the total number of training objects, and c is the number of classes. $P(A_k{=}v \,|\, C_i)$ is estimated by

$$\frac{\#(A_k{=}v \wedge C_i) + 2\frac{\#(A_k=v)}{\#(A_k\neq ?)}}{\#(A_k \neq ? \wedge C_i) + 2} \tag{4}$$

where $\#(A_k{=}v \wedge C_i)$ denotes the number of training examples belonging to class C_i with value v for attribute A_k; $\#(A_k{=}v)$ denotes the number of training examples with value v for attribute A_k; $\#(A_k \neq ?)$ denotes the number of training examples for which the value is known for attribute A_k; and $\#(A_k \neq ? \wedge C_i)$ denotes the number of training examples belonging to class C_i for which the value is known for attribute A_k.

Three variants of APNBC were evaluated, each employing different values of α, the critical value for the binomial test. One used $\alpha = 0.05$, another used $\alpha = 0.1$, and the last used $\alpha = 1$ (the binomial test is ignored). The value 0.05 was chosen because this is a classic critical value employed in statistics. A less stringent value, 0.1, was also considered, as the binomial test controls only the risk of accepting an inappropriate adjustment by chance and it was thought that a less stringent critical value might reduce the risk of type 2 error rejecting an appropriate adjustment by chance, more than it increased the risk of the type 1 error that it explicitly controlled. The third option, ignoring the binomial test, was included in order to assess the efficacy of the test. (These are the only α values with which the software has been evaluated, as it is deemed important not to perform parameter tuning to the available data sets.)

These three variants of APNBC were also compared with a standard naive Bayesian classifier (the same computer program with the adjustment induction phase disabled).

Thirty representative data sets from the UCI repository (Merz & Murphy, 1998) were employed. These are presented in Table 1. Continuous attributes were discretized at induction time by finding cut points in the training data that resulted in the formation of ten groups containing as near as possible to equal numbers of training examples.

For each data set, ten-fold cross validation experiments were run ten times. That is, each data set was divided into ten random partitions of as near as possible to equal size. For each of these partitions in turn, every variant of the system was trained on the remaining nine partitions and predictive accuracy evaluated on the with-held partition. This was repeated with ten different random partitionings for each data set.

Table 2 presents a summary of the results of this experiment. For each data set, the mean percentage predictive error is presented for each variant of the system. For each of the treatments using probability adjustments, a summary is provided of the number of wins, losses and draws, when the mean error is compared to that of the naive Bayesian classifier. The p value from a one-tailed binomial sign test is also provided to evaluate the significance of these win/loss/draw results.

It can be seen that with $\alpha = 0.05$, APNBC is selective about inferring adjustments that have measurable effect. For only eight out of thirty data sets are differences in predictive error evident. In seven of these the adjustments lead to a decline in error while for only one does error increase. The one data set on which error does increase, monk1, is an artificial data set. A binomial sign test reveals that the probability of such an outcome occurring by chance is just 0.035,

Table 1. UCI data sets used in experimentation

Domain	Cases	Attributes
adult	48843	18
audio	226	69
balance-scale	625	25
breast cancer Slov.	286	9
breast cancer Wisc.	699	9
cleveland	303	13
crx (Aust. credit)	690	15
echocardiogram	74	6
glass	214	9
horse-colic	368	21
house-votes-84	435	16
hungarian	294	13
hypo	3772	29
iris	150	4
kr-vs-kp	3196	36
lenses	24	4
lymphography	148	18
monk1	556	6
monk2	601	6
monk3	554	6
mp11	500	11
mush	8124	22
phoneme	5438	7
Pima diabetes	768	8
promoters	106	57
primary tumor	339	17
soybean large	683	35
splice-c4.5	3177	60
tic-tac-toe	958	10
waveform	300	21

and hence is the advantage is significant at the 0.05 level. It can be seen that most of the differences are of large effect when the ratio of the new error over the old error is considered. Of the data sets for which a difference is obtained, the average ratio is 0.84, indicating that an average improvement of 16% is obtained. Even once all the data sets for which there is no difference are included, the average ratio is 0.96 indicating an average reduction in error by 4%.

As the value of α is relaxed, however, there is an increase in the number of differences in performance. At $\alpha = 0.1$, there are differences for fourteen out of the thirty data sets. However the ratio of positive to negative effects is nine to five, which a one tailed sign test reveals as not statistically significant. The error ratio at this level indicates that error is reduced by 3% on average over

Table 2. Summary of results (mean percentage error)

		$\alpha = 0.05$		$\alpha = 0.1$		$\alpha = 1$	
Domain	naive	mean	ratio	mean	ratio	mean	ratio
adult	18.2	16.1	0.88	16.1	0.88	16.1	0.88
audio	27.5	27.5	1.00	27.5	1.00	25.7	0.93
balance-scale	9.0	8.8	0.98	9.9	1.10	10.7	1.19
breast cancer Slov.	28.7	28.7	1.00	28.7	1.00	29.4	1.02
breast cancer Wisc.	2.4	2.4	1.00	2.4	1.00	2.6	1.08
cleveland	16.5	16.5	1.00	17.5	1.06	17.5	1.06
crx (Aust. credit)	14.3	14.3	1.00	14.3	1.00	14.5	1.01
echocardiogram	29.8	29.8	1.00	29.8	1.00	31.1	1.04
glass	31.9	31.9	1.00	31.9	1.00	33.6	1.05
horse-colic	18.7	18.7	1.00	20.6	1.10	20.6	1.10
house-votes-84	9.7	9.7	1.00	9.7	1.00	12.9	1.33
hungarian	15.6	15.6	1.00	15.6	1.00	16.6	1.06
hypo	3.5	3.5	1.00	3.3	0.94	3.3	0.94
iris	7.3	7.3	1.00	7.3	1.00	6.7	0.92
kr-vs-kp	12.6	12.6	1.00	12.7	1.01	12.6	1.00
lenses	28.3	28.3	1.00	28.3	1.00	21.7	0.77
lymphography	16.9	16.9	1.00	16.9	1.00	15.5	0.92
monk1	25.4	27.4	1.08	27.9	1.10	30.4	1.20
monk2	39.1	37.1	0.95	35.1	0.90	34.3	0.88
monk3	3.6	2.2	0.61	2.2	0.61	2.2	0.61
mp11	41.2	41.2	1.00	41.2	1.00	42.6	1.03
mush	1.8	1.2	0.67	1.2	0.67	1.2	0.67
phoneme	27.4	27.4	1.00	27.3	1.00	26.4	0.96
Pima diabetes	24.4	24.4	1.00	24.2	0.99	24.9	1.02
promoters	11.3	11.3	1.00	11.3	1.00	14	1.24
primary tumor	52.2	52.2	1.00	52.2	1.00	54.9	1.05
soybean large	6.6	6.6	1.00	6.6	1.00	8.1	1.23
splice-c4.5	4.5	4.5	1.00	4.5	1.00	4.7	1.04
tic-tac-toe	33.1	26.2	0.79	26.7	0.81	26.7	0.81
waveform	23.7	18.7	0.79	18.7	0.79	18.7	0.79
Mean ratio			0.96		0.97		1.00
Win/loss/draw		7/1/22		9/5/16		12/17/1	
Win/loss/draw p		0.035		0.212		0.229	

these data sets at this α level. At $\alpha = 1$, there are differences for twenty-nine out of thirty data sets, of which twelve are decreases in error and seventeen are increases in error. A one-tailed t-test also reveals this ratio as not statistically significant at the 0.05 level.

While APNBC with $\alpha = 1$ (which, it should be recalled, has the effect of disabling the binomial test) results in more increases in error than decreases, when compared with the naive Bayesian classifier, the mean ratio of error rates is 1.00,

indicating that the individual positive effects tend to be greater than individual negative effects, although this is counter-balanced by a greater frequency of negative effects.

5 Conclusion

We have proposed that the probabilities produced by a naive Bayesian classifier could be systematically adjusted to accommodate violations of the assumptions on which it is based. We have investigated induction of simple linear adjustments in the form of a numeric weights by which the inferred probabilities for a class are multiplied. This was performed in a context where discrete class prediction was performed, rather than probabilistic prediction, so our concern has not been to obtain accurate probabilities from the classifier, but rather to obtain probabilities weighted in favor of the correct class.

For many data sets, accepting any adjustment that improves resubstitution accuracy results in adjustments that produce small increases in predictive error. The use of a binomial test, to limit adjustments to those that result in alterations in resubstitution error that are unlikely to occur by chance, blocks most adjustments with negative effect. The resulting system infers adjustments for approximately one quarter of data sets, but almost all adjustments inferred result in reductions in predictive error. Further, many of those reductions are of substantial magnitude.

Acknowledgments

We are grateful to Zijian Zheng for supplying the source code for a naive Bayesian classifier which we subsequently modified for our experiments. We are also grateful to the maintainers and contributors to the UCI repository of machine learning databases, including the Ljubljana Oncology Institute, Slovenia, Dr. William H. Wolberg of the University of Wisconsin Hospitals, Madison, and Prof. Jergen of the Baylor College of Medicine, for making available the data with which this research was conducted. Much of this research was conducted while the first named author was visiting the University of California, Irvine, Department of Information and Computer Science. He is grateful to the Department for its support during this visit.

A The APNBC Induction Algorithm

n is the number of training objects.

A_i is the APNBC adjustment factor for class i.

$C(o)$ returns the true class of object o.

$P(o, i)$ returns the probability inferred by the naive Bayesian classifier that object o belongs to class i.

$APNBC(o)$ returns the class assigned to object o given the current adjustment values. This equals $argmax_i\, A_i P(o,i)$.

$error()$ returns the number of training objects misclassified by APNBC given the current adjustment values. This equals $|\{o : APNBC(o) \neq C(o)\}|$.

ϵ is a very small value. The current implementation uses 10^{-5}. This is used to alter adjustments from a value at which there is a tie between two classes.

$adjustup(a,c)$ returns $a - \frac{a-b}{2}$, where b is the lowest value greater than a for which $A_c = b$ results in higher error than $A_c = a$. This is used to select the midpoint in a range of adjustment values all of which have the same effect, where a is the lower limit of the range.

$adjustdown(a,c)$ returns $a - \frac{a-b}{2}$, where b is the highest value less than a for which $A_c = b$ results in higher error than $A_c = a$. This is used to select the midpoint in a range of adjustment values all of which have the same effect, where a is the upper limit of the range.

$binomial(n,t,p)$ returns the binomial probability that of obtaining n positive results out of t trials if the true underlying proportion of positives is p. This is compared against a predefined critical value, α, which in the current implementation defaults to 0.05.

For each class i, $A_i \leftarrow 1$.
$best \leftarrow error()$.
Repeat
    For each training object o such that $C(o) \neq APNBC(o)$
        $e \leftarrow APNBC(o)$.
        $saveA \leftarrow A$.
        $adj \leftarrow \frac{A_e P(o,e)}{P(o,C(o))}$.
        $A_{C(o)} \leftarrow adj + \epsilon$.
        If $error() < best$ or ($error() = best$ and $adj - saveA_{C(o)} < bestd$)
            $best \leftarrow error()$.
            $bestc \leftarrow C(o)$.
            $besta \leftarrow adj$.
            $bests \leftarrow +$.
            $bestd \leftarrow adj - saveA_{C(o)}$.
        $A \leftarrow saveA$.
        $adj \leftarrow \frac{A_{C(o)} P(o,C(o))}{P(o,APNBC(o))}$.
        $A_e \leftarrow adj - \epsilon$.
        If $error() < best$ or ($error() = best$ and $saveA_e - adj < bestd$)
            $best \leftarrow error()$.
            $bestc \leftarrow e$.
            $besta \leftarrow adj$.
            $bests \leftarrow -$.
            $bestd \leftarrow saveA_e - adj$.
        $A \leftarrow saveA$.

If $binomial(best, n, \frac{error()}{n}) < \alpha$
If $bests = +$
$A_{bestc} \leftarrow adjustup(besta, bestc)$.
else
$A_{bestc} \leftarrow adjustdown(besta, bestc)$.
$continue \leftarrow true$.
else
$continue \leftarrow false$.
until $continue = false$.

References

Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–284.

Domingos, P. & Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 105–112, Bari, Italy. Morgan Kaufmann.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York.

Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *KDD-96* Portland, Or.

Kononenko, I. (1991). Semi-naive Bayesian classifier. In *ECAI-91*, pp. 206–219.

Langley, P. (1993). Induction of recursive Bayesian classifiers. In *Proceedings of the 1993 European Conference on Machine Leanring*, pp. 153–164, Vienna. Springer-Verlag.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 223–228, San Jose, CA. AAAI Press.

Langley, P. & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399–406, Seattle, WA. Morgan Kaufmann.

Merz, C. J. & Murphy, P. M. (1998) *UCI Repository of Machine Learning Databases.* [Machine-readable data repository]. University of California, Department of Information and Computer Science, Irvine, CA.

Pazzani, M. J. (1996). Constructive induction of Cartesian product attributes. In *ISIS: Information, Statistics and Induction in Science*, pp. 66–77, Melbourne, Aust. World Scientific.