Case based imprecision estimates for Bayes classifiers with the Bayesian bootstrap

Niklas Norén (niklas.noren@neurologic.se)
NeuroLogic, Johan Enbergs v. 28, 171 61 Solna, Sweden

Roland Orre (roland.orre@neurologic.se, orre@math.su.se)
NeuroLogic, Johan Enbergs v. 28, 171 61 Solna, Sweden
Department of Mathematical Statistics, Stockholm University, Stockholm, Sweden

Abstract. This article outlines a Bayesian bootstrap method for case based imprecision estimates in Bayes classification. We argue that this approach is an important complement to methods such as k-fold cross validation that are based on overall error rates. It is shown how case based imprecision estimates may be used to improve Bayes classifiers under asymmetrical loss functions. In addition, other approaches to making use of case based imprecision estimates are discussed, and illustrated on two real world data sets. Contrary to the common assumption, Bayesian bootstrap simulations indicate that the uncertainty associated with the output of a Bayes classifier is often far from normally distributed.

Keywords: Case based imprecision estimates, Bayes classifier, Bayesian bootstrap, naive Bayes

1. Introduction

In supervised learning, a set of labelled training examples, with known values for both predictor and response variables, is provided. The general aim is to train a classifier to predict unobserved variable values of new instances, based on the characteristics of the labelled instances.

Bayes classifiers put supervised learning in a probabilistic framework where all possible values of a missing response variable are assigned estimated probabilities, based on the values of the observed variables and on prior probabilities for the unobserved variables. This is especially useful when the response variable is not fully determined by the predictor variables, *i.e.* when two cases with identical values for the predictor variables may have different values for the response variable. Under such circumstances there is clearly uncertainty associated with any output of the classifier.

The accuracy of Bayes classifiers has previously been studied with methods such as k-fold cross-validation (Kohavi, 1995), which give overall accuracy estimates for a classifier given some training data. Such methods do not account for the variability between cases in the associated uncertainty, and relate to the number correctly classified cases rather than to the precision of the attributed probabilities.

For neural networks, MacKay (1992) has suggested case specific uncertainty estimates based on Bayesian inference and the assumption that the uncertainty associated with the weights in the network can be described by normal distributions (see also Bishop, 1995). This approach allows each classification performed by such a neural network to be accompanied by precision estimates.

In this article, we propose a Bayesian bootstrap method for case based imprecision estimates similar to those of MacKay, but for Bayes classifiers. For clarity of presentation, we focus on the naive Bayes classifier (Kononenko, 1990), but the methods apply equally well to generalized Bayes classifiers such as the semi-naive classifiers discussed in for example Kononenko (1991) and Domingos and Pazzani (1997).

The aims of this article is to emphasize the importance of case based imprecision estimates, to show how the Bayesian bootstrap may be used to generate precise case based estimates and to indicate how this information can be used for better informed Bayes classification.

In earlier work, Orre et al. (2000) proposed case specific precision estimates for a semi-naive Bayes classifier based on a normal approximation, and Orre and Lansner (1996) described a method for how similar uncertainty estimates may be obtained for real-valued variables.

2. The Bayesian bootstrap

Bootstrap methods in general (Efron, 1979) study how parameter estimates vary when a data set is resampled. A special type of bootstrap method is the Bayesian bootstrap (Rubin, 1981). In the Bayesian bootstrap, replicates of a given data set $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ are generated by assigning Dirichlet distributed random weights to the cases $\mathbf{x}_i$ in the original data set. The parameter of interest is calculated for each bootstrap replicate, and as shown by Rubin (1981), the distribution of the calculated parameter values over the replicated data sets approximates the posterior distribution of this parameter. This is very helpful in situations where no closed form expression for the posterior distribution is known. Based on the Bayesian bootstrap replicates of a data set, we may form a histogram to estimate the full posterior distribution or to calculate point estimates for *e.g.* the posterior mean or a certain percentile of the distribution.

The most straightforward approach to resampling in the Bayesian bootstrap, is to assign $Di(1, \ldots, 1)_n$ distributed weights to the observed instances $\mathbf{x}_i$. However, when many $\mathbf{x}_i$ are equal, it is more efficient to assign $Di(n_1, \ldots, n_m)$ weights to the m distinct values $\mathbf{d}_j$ of $\mathbf{X}$, where n_j is the number of $\mathbf{x}_i$ equal to $\mathbf{d}_j$. Due to the nature of the Dirichlet distribution, these two operations are mathematically equivalent.

Let $\theta = \{\theta_1, \ldots, \theta_m\}$ be the vector of probabilities $\theta_j = P(\mathbf{X} = \mathbf{d}_j \mid \theta)$. With $Di(n_1, \ldots, n_m)$ distributed weights, the Bayesian bootstrap posterior distribution is proportional to:

$$\prod_{j=1}^{m} \theta_j^{n_j - 1} \tag{1}$$

and the corresponding implicit prior distribution is proportional to (Rubin, 1981):

$$\prod_{j=1}^{m} \theta_j^{-1} \tag{2}$$

This is sometimes referred to as Haldane's prior.

Bayesian bootstrap simulation based on a more general prior distribution is however possible. A prior distribution proportional to:

$$\prod_{j=1}^{m} \theta_j^{l_j} \tag{3}$$

yields a posterior distribution proportional to (Rubin, 1981):

$$\prod_{j=1}^{m} \theta_j^{n_j + l_j} \tag{4}$$

It can be simulated by assigning $Di(n_1 + l_1 + 1, \ldots, n_m + l_m + 1)$ distributed weights to the vector of distinct values $\mathbf{d} = \{\mathbf{d}_1, \ldots, \mathbf{d}_m\}$.

A major advantage of such a more general prior distribution is that zero counts (when two variable values have never been observed together) may be handled in a better way than with Haldane's prior or with classical statistics. In fact, Domingos and Pazzani (1997) uses this argument to motivate their use of a Laplace corrector, which corresponds to a Dirichlet prior distribution with $l_j = 1$.

One limitation of bootstrap methods in general is the indirect assumption that all possible variable values have been observed. In fact, with several variables, the bootstrap methods effectively assume that all possible *combinations* of variable values have been observed, but in Section 4.1 we propose a modification to the Bayesian bootstrap that reduces the negative impact of this assumption.

The choice of prior distribution clearly has an impact on any analysis based on Bayesian bootstrap simulation. In the experiments presented in this article, we have aimed to minimize the prior's impact on the final result, while retaining a moderating effect, by using a prior distribution with small prior sample size (Gelman et al., 1995). The sensitivity of the results to the choice of prior has also been investigated.

3. Bayes classifiers

The aim of Bayes classifiers is to assign the value:

$$\operatorname*{argmax}_{y_j} \; P(Y = y_j \mid X_1 = x_1, \ldots, X_m = x_m) \tag{5}$$

to Y, for any unlabelled instance

$$(x_1, \ldots, x_m)$$

The actual class probabilities are however unknown parameters of an underlying model, wherefore classifiers are generally based on estimates $\hat{P}(y_j \mid \mathbf{x})$ from a batch of labelled training data:

$$(\mathbf{x}(k), y(k)), \;\; k = 1, \ldots, n$$

with observed values for both predictor and response variables.

Implemented Bayes classifiers often use classical maximum likelihood estimates, but the method proposed in this article is based on a full Bayesian approach. Bayesian inference combines prior information on a parameter's value with observed data, to yield the posterior probability distribution of the parameter (Gelman et al., 1995). In the following, we either consider the full posterior distribution of a parameter or use Bayesian point estimates such as the *posterior mean estimate* or the *maximum à posteriori estimate*.

If the number of training examples is large compared to the number of possible predictor variable value configurations, the outcome probabilities

$$P(y \mid \mathbf{x}) \tag{6}$$

may be directly estimated from data. Classifiers based on full conditional probabilities are sometimes referred to as optimal Bayes classifiers (Mitchell, 1997), since they rely on no assumptions of mutual independence between the different predictor variables.

The drawback with the optimal Bayes approach is that in real applications, there are seldom large enough numbers of training examples to sufficiently populate the entire domain of possible predictor variable configurations. This sparsity of data increases rapidly with the number of predictor variables used, due to the exponential increase in the number of possible configurations, something that is commonly referred to as the *curse of dimensionality*.

3.1. The naive Bayes classifier

In naive Bayes classification, the predictor variable values are assumed to be mutually independent conditional on the class, and this assumption allows for the following modified expression:

$$\begin{aligned} P(y \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid y)}{P(\mathbf{x})} \cdot P(y) \propto P(\mathbf{x} \mid y) \cdot P(y) = \\ = P(x_1 \mid y) \cdot \ldots \cdot P(x_m \mid y) \cdot P(y) \end{aligned} \tag{7}$$

which is normalized through division by the sum of the different class probabilities:

$$\sum_j P(y_j \mid \mathbf{x})$$

With respect to the curse of dimensionality, the main advantage of the naive Bayes approach is that it is based on estimates of *marginal* probabilities, $P(x_i \mid y)$, rather than of full conditional probabilities, $P(y \mid \mathbf{x})$, something that significantly reduces the amount of training data required for reliable parameter estimation.

The drawback with the naive Bayes approach is that the underlying assumption of mutual independence between all predictor variables is commonly violated. Nevertheless, this approach has proven to be very versatile and to often compare well with more sophisticated methods (Domingos and Pazzani, 1997; Hand and Yu, 2001).

3.2. The semi-naive Bayes classifier

Semi-naive Bayes classifiers (Kononenko, 1991) allow for data models where some but not all dependencies between variables are accounted for. Groups of dependent predictor variables are encoded as composite variables whose possible values are combinations of the original variables' values. The aim is to identify a set of mutual independence assumptions that optimizes the trade-off between accuracy and computational efficiency.

Consider for example a Bayes classifier where on one hand x_1, x_2 and x_3 and on the other hand x_4 and x_5 are coencoded as composite variables. Equation 7 becomes:

$$P(y \mid \mathbf{x}) \propto P(x_1, x_2, x_3 \mid y) \cdot P(x_4, x_5 \mid y) \cdot \ldots \cdot P(x_m \mid y) \cdot P(y) \quad (8)$$

The semi-naive Bayes classifier may be regarded as a naive Bayes classifier with respect to the coencoded variables. For clarity of presentation we will therefore focus the remainder of our discussion on the naive Bayes classifier, but it should be kept in mind that all presented methods apply to other Bayes classifier as well.

4. Methodology

Because Bayes classifiers are based on re-expressions of $P(y \mid x_1, \ldots, x_m)$ as products of several unknown probability parameters (see Equations 7 and 8), no analytical form for the posterior distributions of the class probabilities is known. We propose the Bayesian bootstrap method be used to obtain accurate estimates for these posterior distributions. Given a Bayes classifier and a large enough number of bootstrap replicates, the Bayesian bootstrap yields arbitrarily accurate estimates, and unlike methods proposed in earlier work (MacKay, 1992; Orre et al., 2000), it does not rely on normal approximations.

4.1. An adjusted Bayesian bootstrap

The original Bayesian bootstrap method requires the assignment of a random weight to each individual case in the training data. For large data sets this may be intractable — to draw 10 000 bootstrap replicates from a data set with a million cases requires around 10 billion non-uniform random numbers to be generated. Clearly, under such circumstances a more efficient approach is necessary.

In principle, the computational complexity of the Bayesian bootstrap may be reduced by assigning random weights to each distinct set of predictor variable values rather than to each specific case in the training data set, as discussed in Section 2. However, the number of predictor variables is in practice often large enough that there is only a small number of cases with the exact same sets of predictor variable values. Consequently, this may not be sufficient to make the Bayesian bootstrap computationally tractable.

One way to further decrease the computational complexity of the Bayesian bootstrap is to incorporate the mutual independence assumptions on which the Bayes classifier relies into the resampling procedure. In such an adjusted Bayesian bootstrap approach, each factor in the Bayes classifier formula is simulated independently, and bootstrap replicates are generated as follows:

1. First draw bootstrap replicates for the marginal probabilities $P(y_j)$ from the $Di(n_1+l_1, n_2+l_2, \ldots)$ distribution where n_j is the number of cases with $Y = y_j$ and l_j is a prior parameter

2. For each predictor variable x_i:

   a) Draw bootstrap replicates for the conditional probabilities $P(x_i|y_j)$ from the $Di(n_1 + l_1, n_2 + l_2, \ldots)$ distribution where n_j is the number of cases with $Y = y_j$ and $X = x_i$, and where l_j is a prior parameter

3. Calculate the unnormalized posterior probability: $P^*(y_j|\mathbf{x}) = P^*(x_1|y_j) \cdot \ldots \cdot P^*(x_m|y_j) \cdot P^*(y_j)$

4. Normalize each $P^*(y_j|\mathbf{x})$ through division by $\sum_j P^*(y_j|\mathbf{x})$

The adjusted Bayesian bootstrap method produces the posterior class distribution under the given model assumptions, *i.e.* it accounts for the mutual independence assumptions in the resampling. This further reduces the impact of the bootstrap assumption that all possible combinations of variable values have been observed (see Section 2). By simulating all predictor variables separately, any two variable values that occur separately in the training data are assumed to have a positive probability to cooccur.

Furthermore, this adjustment to the Bayesian bootstrap facilitates the assertion of a prior distribution. In the adjusted Bayesian bootstrap, each variable has its own prior distribution, so the number of prior parameters is equal to $\sum_i v_i$ (where v_i is the number of distinct variable values for variable X_i) instead of to $\prod_i v_i$ as in the original Bayesian bootstrap.

4.2. How to make use of the posterior class probability distributions

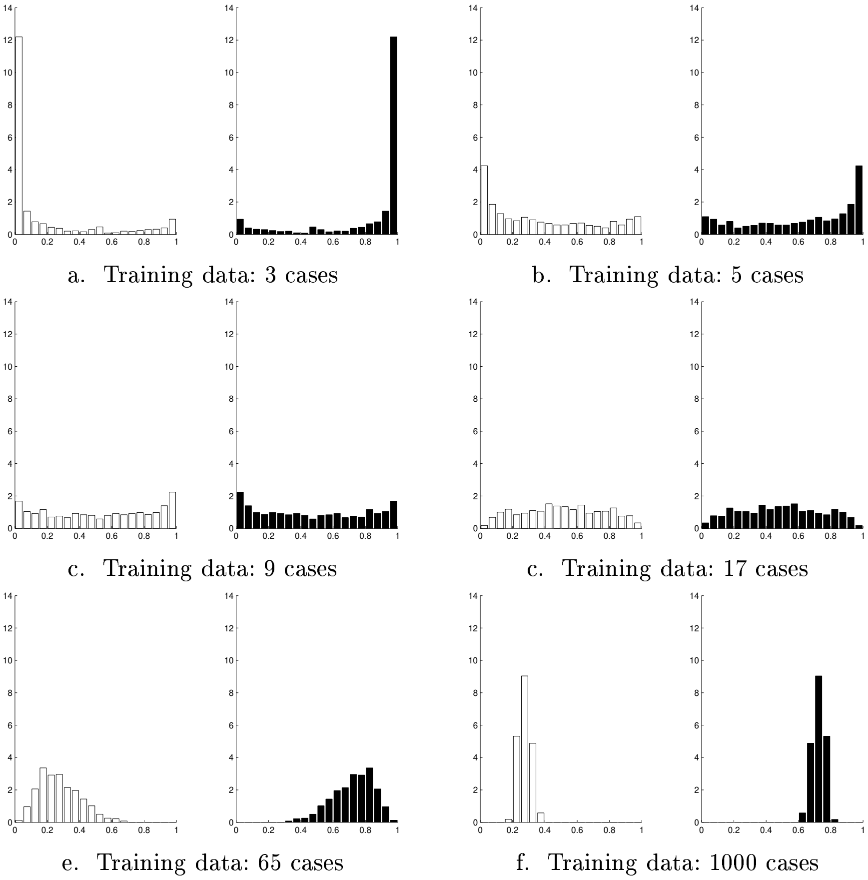
There are several ways to put the Bayesian bootstrap distributions to use. Detailed information about imprecision in the output probabilities of a Bayes classifier may be used to test model assumptions such as *e.g.* normal approximations, and to investigate what factors influence the imprecision in Bayes classification.

MacKay (1992) proposes marginalization (averaging) over the posterior distribution of a classification as a way to moderate the output of a neural network. For binary variables, this tends to pull the output probabilities toward 0.5, and the effect is stronger the less training data there is available. In effect this corresponds to the use of posterior means instead of maximum likelihood estimates, and it would be straightforward to implement the same principle for Bayesian bootstrap analysis of Bayes classifiers. A slightly generalized approach is to assert a baseline probability for each class, and to only output a different value (the closest credibility interval limit) for the probability if the credibility interval excludes the baseline value. This allows for a stronger moderating effect, which can be fine tuned by varying the coverage of the credibility interval. It also allows for moderation toward other values than 0.5 between 0 and 1. Another approach is to altogether refrain from making classifications with too little support in data, and instead flag cases as *uncertain* if the posterior intervals span the prior probabilities.

A situation where detailed information about the output class probability posterior distributions may be particularly useful is in Bayes classification under asymmetrical loss functions. Bayes classifiers typically output the most probable class for an unlabelled case, but if the different types of misclassifications have different associated losses, this is suboptimal. For binary classifiers and loss functions based on variable misclassifications costs, classification based on percentiles equal to the ratios of the misclassification costs minimizes the expected loss. Section 5.3 presents a detailed example of this. Another example is filters for unwanted e-mail messages (spam), where it is generally more severe to misclassify a wanted e-mail message as *spam* than to misclassify spam as *wanted.*

5. Examples

To illustrate the usefulness of the proposed approach for real data, we have applied it to data sets from the UCI machine learning repository (Blake and Merz, 1998). The results are presented in this section.



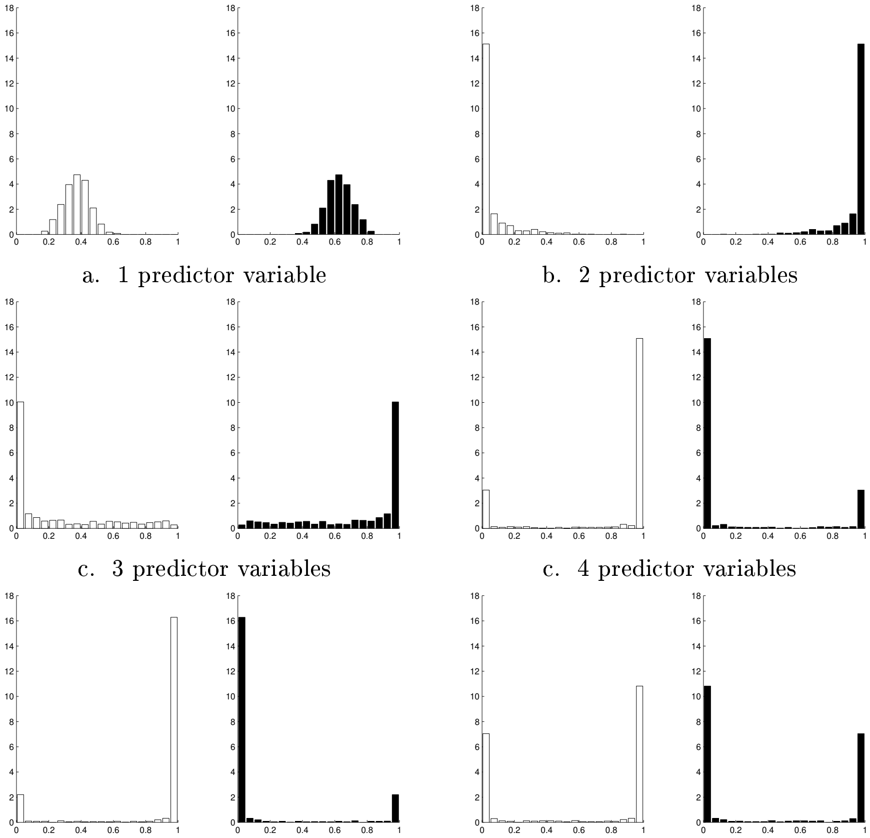
a. Training data: 3 cases
b. Training data: 5 cases
c. Training data: 9 cases
c. Training data: 17 cases
e. Training data: 65 cases
f. Training data: 1000 cases

Figure 1. Posterior distributions for the probability that mushroom number 8124 in the UCI ML repository data set is edible (the leftmost distribution in each pair) and poisonous (rightmost) for varying amounts of training data, using the first 4 attributes (*cap shape*, *cap surface*, *cap color* and *bruises*).

5.1. Setup of experiments

Two data sets from the UCI machine learning repository were selected: the mushrooms data set and the zoology data set. From the mushrooms data set, we excluded the predictor variable *veil-type* for which only one value (*partial*) is ever observed. Some of the analyses were based on subsets of the available cases and/or predictor variables in the data sets, in order to better illustrate the impact of uncertainty on the classification.

For the predictor variables, we used hyper parameters $l_i = \frac{v_r}{v_i}$ (see Equation 4) where v_r is the number of distinct values for the response variable and v_i is the number of distinct values for the predictor variable



a. 1 predictor variable — b. 2 predictor variables

c. 3 predictor variables — c. 4 predictor variables

e. 10 predictor variables — f. All 21 predictor variables

Figure 2. Posterior distributions for the probability that mushroom number 8124 in the UCI ML repository data set is edible (the leftmost distribution in each pair) or poisonous (rightmost) for varying numbers of predictor variables, based on 80 training cases. The predictor variables were added in the following order: *stalk shape*, *population*, *odor*, *stalk color below ring*, *cap color*, *gill spacing*, *stalk surface below ring*, *ring number*, *ring type* and then the rest.

of interest. For the response variable this amounts to hyper parameters $l_j = 1$. This is a low impact prior distribution with prior sample size v_r.

5.2. Precise posterior distribution estimation

To illustrate how the Bayesian bootstrap may be used to infer detailed knowledge about the posterior distribution of the Bayes classifier's output, we have used precise Bayesian bootstrap simulations (1000 replicates) to study the posterior distributions of different naive Bayes classifiers applied to the mushrooms data set.

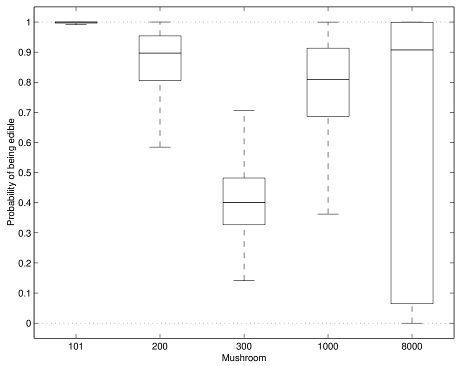


Figure 3. Box plots for the output classification of different mushrooms based on the predictor variables: *cap shape*, *cap surface*, *cap color* and *bruises* and the first 100 mushrooms in the data set. Clearly, the degree of uncertainty varies varies for the five mushrooms, even though the same Bayes classifier has been used.

All distributions in Figure 1 are produced by the same naive Bayes classifier, which uses the following four predictor variables: *cap shape*, *cap surface*, *cap color* and *bruises*. The aim is to show how the uncertainty in the output decreases as more training data is added. Please note how the shape of the posterior distribution transforms from its bathtub shape for small amounts of training data over an almost uniform distribution for intermediate amounts of training data to a more normal-like posterior distribution for large amounts of training data.

In Figure 2 we have used constant training data, but different naive Bayes classifiers. The difference between the classifiers is the number of predictor variables on which they are based, and the aim is to illustrate how the uncertainty in the output increases as more predictor variables are added.

5.3. Classifying mushrooms with variable misclassification costs

As an illustration of how case based precision estimates may be incorporated into a Bayes classifier, a naive Bayes classifier was trained on the first 100 cases in the mushrooms data set using the first four predictor variables: *cap shape*, *cap surface*, *cap color* and *bruises*. Figure 3 displays box plots for the uncertainty associated with the classifications of five different mushrooms that were not included in the training data.

If the loss associated with the misclassification of a poisonous mushroom as edible is higher than the loss associated with the misclassification of an edible mushroom as poisonous, simple naive Bayes classification (outputting the class estimated to be the most likely) is suboptimal. Assume for simplicity that the loss associated with the former type of misclassification is twice that of the latter; without case based imprecision estimates, the easiest way to account for this asymmetry is to change the cut-off from 50% to $66.\bar{6}\%$ (*i.e.* only label a mushroom as edible if the point estimate for this probability is greater than 2/3, since this is the level at which the expected loss of classifying the mushroom as *edible* is the same as that of classifying the mushroom as *poisonous*). The reliability of the naive Bayes probability estimates is however questionable as the naive Bayes classifier tends to overestimate the confidence in its predictions (Hand and Yu, 2001). With case based uncertainty estimates, an alternative approach is to instead consider variation in the output classification (with cut-off 50%) over the Bayesian bootstrap distribution and only label the mushroom as *edible* if more than 2/3 of the bootstrap replicates indicate *edible*.

To compare these two approaches, we have used five different naive Bayes classifiers each trained on 100 out of the first 500 mushrooms in the data set (and the predictor variables: *cap shape*, *cap surface*, *cap color* and *bruises*) to classify the last 100 mushrooms in the data set. A comparison of the two decision rules and the standard naive Bayes decision rule is displayed in Table I.

5.4. Sensitivity to the choice of prior

To evaluate how sensitive the Bayesian bootstrap method is to variations in the choice of prior distribution, we have compared the uncertainty estimates for mushroom 8124 with the chosen prior ($l_i = \frac{v_r}{v_i}$) to uncertainty estimates based on two other data sensitive priors: the uniform prior ($l_i = 1$) and Haldane's prior ($l_i = 0$). The results are displayed in Figure 4.

5.5. Function approximation

Figures 5 and 6 display bootstrap posterior distributions for cases in the UCI zoology and mushrooms data sets, together with fitted beta distributions.

6. Discussion

The results reported on in Section 5.3 indicate that case based precision estimates may allow for more robust decision rules when the loss function is asymmetrical. Table I shows that over a rather large domain in the UCI mushrooms data set (five classifiers trained on different portions of the database and each applied to 100 cases) the Bayesian

Table I. The efficiency of three naive Bayes decision rules, for five different naive Bayes classifiers trained on subsets of the UCI mushrooms data set.

a. $\hat{P}(edible) > 1/2$

Training data	True pos.	False pos.	Sens.	Spec.	Loss
1–100	54	33	1.00	0.28	66
101–200	54	25	1.00	0.46	50
201–300	54	25	1.00	0.46	50
301–400	54	13	1.00	0.72	26
401–500	40	4	0.74	0.91	22
Averages:	51.2	20.0	0.95	0.57	42.8

b. $\hat{P}(edible) > 2/3$

Training data	True pos.	False pos.	Sens.	Spec.	Loss
1–100	54	25	1.00	0.46	50
101–200	54	12	1.00	0.74	24
201–300	40	5	0.74	0.89	24
301–400	40	4	0.74	0.91	22
401–500	40	3	0.74	0.93	20
Averages:	45.6	9.8	0.84	0.79	28.0

c. $P(P^*(edible) > 1/2) > 2/3$

Training data	True pos.	False pos.	Sens.	Spec.	Loss
1–100	43	4	0.80	0.91	19
101–200	39	2	0.72	0.96	19
201–300	38	2	0.70	0.96	20
301–400	40	3	0.74	0.93	20
401–500	40	2	0.74	0.96	18
Averages:	40.0	2.6	0.74	0.94	19.2

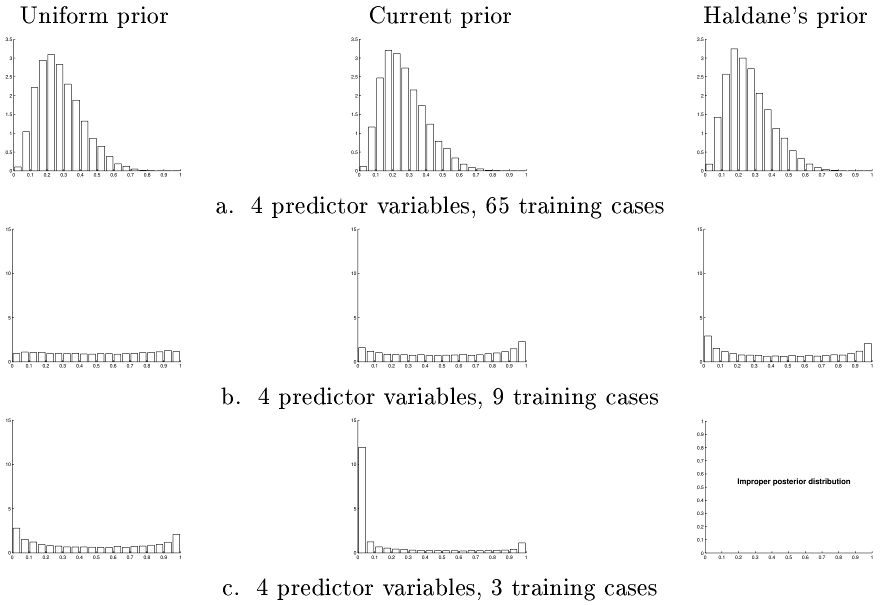


Figure 4. Sensitivity to the choice of prior hyper parameters of the uncertainty estimates for the classification of mushroom 8124 based on various amounts of training data.

bootstrap based decision rule has better specificity, and lower average loss for variable misclassification costs. A plausible explanation for this is the tendency of the naive Bayes classifier to "probability overshoot", *i.e.* to overestimate the confidence in its predictions (Hand and Yu, 2001). Figure 2 illustrates this: even when there is significant uncertainty due to a large number of predictor variables compared to the amount of training data available, the most likely value of the output probability tends to 0 or 1. Due to this low reliability of the naive Bayes probability estimates, even crude Bayesian bootstrap simulation may yield better uncertainty estimates for Bayes classifiers.

However, case based imprecision estimates focus solely on imprecision due to limits in the amount of relevant training data available, and do not account for erroneous assumptions in the design of the classifier. Other methods such as k-fold cross-validation should therefore always be used to test the accuracy of a Bayes classifier. If the accuracy of the classifier is poor, the Bayesian bootstrap may be used to deduce whether this could be due to limits in the amount of relevant training data, or whether it is solely attributable to incorrect model assumptions.

A good example of this distinction between precision and accuracy is the classification of mushroom number 8124 in Figure 1. As more

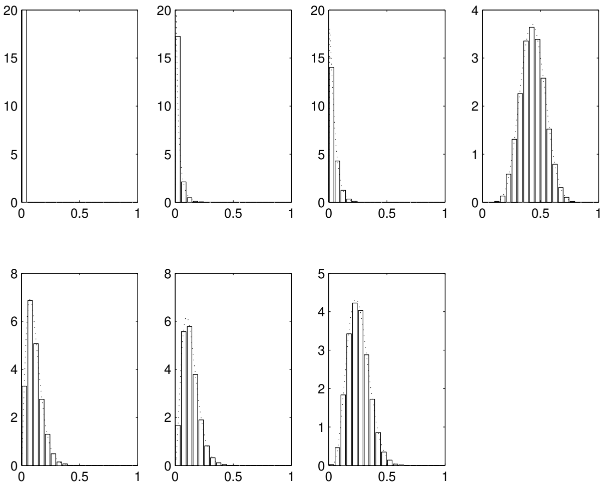
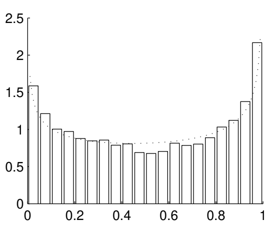
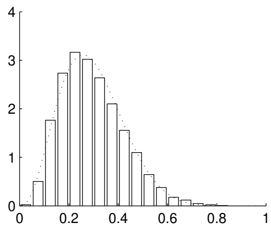


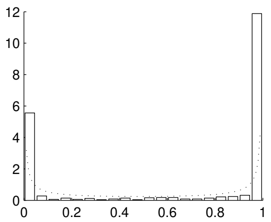
Figure 5. Fitted beta distributions (dotted curves) for the 7 output distributions (one for each class) of animal number 81 in the UCI zoology data set, based on the first 80 animals in the data set and the first four predictor variables.



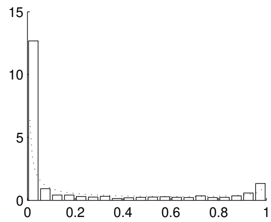
a. 4 predictor variables, 8 training cases



b. 4 predictor variables, 80 training cases



c. 9 predictor variables, 8 training cases



d. 9 predictor variables, 80 training cases

Figure 6. Fitted beta distributions (dotted curves) for the probability of mushroom number 8124 to be edible, with different numbers of predictor variables and varying amounts of training data.

and more mushrooms are added to the training data, the imprecision in the classification is gradually reduced, and with 1000 mushrooms in the training data, the output probabilities are quite precise (centered at around $P(poisonous) = 0.7$). However, if all predictor variables are used, the output for $P(poisonous)$ is instead very close to 0, and indeed the true label of mushroom number 8124 is *edible*. Clearly the problem here is not the amount of relevant training data, but the design (*e.g.* the feature selection) of the Bayes classifier.

The experiments reported on in this article are in agreement with the observation by Hand and Yu (2001) that the more predictor variables that are used in the Bayes classifier, the more training data is required for reliable prediction. This relates to the issue of irrelevant variables: if a predictor variable is completely unassociated with the response variable, its average effect will be to only increase the uncertainty in every prediction. This is a good incentive for identifying and excluding from the analysis any irrelevant variables.

Instead of the Bayesian bootstrap, the original bootstrap (Efron, 1979) could be used to generate case based precision estimates (based on sampling distributions for parameter estimates rather than on posterior distributions for the parameters). However, the non-Bayesian bootstrap does not allow for the use of prior distributions, which as discussed in Section 4.1 may reduce the negative impact of the bootstrap assumption that all possible data points have been observed. In addition, the parameter estimate distribution is discrete, and may be inconsistent with the observed data: consider for example an attempt to study the probability p of a binomial distribution with the original bootstrap. With four observations — two successes and two failures — each bootstrap replicate has a 1/16 chance of yielding $\hat{p}^* = 1$ for the probability of success, which is clearly in disagreement with data since under this model, the probability of observing the two failures is 0.

As discussed in Section 4.1, the proposed adjustment to the Bayesian bootstrap helps to further reduce the negative impact of the bootstrap assumption that all possible data points have been observed. In addition, it often allows for more efficient simulation: if v_i is the number of distinct variable values for variable X_i, then the computational complexity of the original Bayesian bootstrap is proportional to $\alpha \cdot \prod_i v_i$, whereas the computational complexity of the adjusted Bayesian bootstrap is proportional to $\sum_i v_i$ (α is a factor that relates to how many of the possible variable value combinations that actually occur in the data set).

It is difficult to give general guidelines for how many Bayesian bootstrap replicates are required for a given reliability. To a large extent, this depends on the specific purpose for which the Bayesian bootstrap is carried out — the expected number of replicates for accurate simulation of the posterior mean is for example lower than for the 0.01 quantile (Gelman et al., 1995). For standard purpose simulations a pragmatic approach may be to, in advance, study the sampling variability of the Bayesian bootstrap estimates for a given statistic and a given number of replicates.

The main drawback of the Bayesian bootstrap approach is the computational complexity. Analytical expressions based on normal approximations have been proposed, but the results in Section 5 indicate that the true output of a Bayes classifier is often far from normally distributed. On the other hand, the results in Section 5.5 suggest that fitted beta distributions often provide good approximations, and it would be a great advantage if approximate closed form expressions for the hyper parameters of the best fit beta distribution could be derived. Computationally intense Bayesian bootstrap simulations could then be replaced by simple formulae for the hyper parameters of a beta distribution. Clearly, this is a highly relevant area for future research related to this article.

The Bayesian bootstrap is by definition based on Dirichlet priors, but Monte Carlo simulation based on a different data model could be carried out and may generate different results. The choice of hyper parameters clearly affects the final output distribution, but as Figure 4 indicates, the output class probabilities of a Bayes classifier are rather insensitive to such changes, as long as fairly weak priors are used and as long as training data consists of more than just a few cases. The prior used throughout this article typically has the bath tub shape (its exact shape depends on the number of variables and the numbers of variable values), but it may be argued that a better approach is to set the prior parameters so that the prior distribution for the class probability is always uniform. This would however result in a larger prior sample size, and a less data sensitive Bayes classifier.

7. Conclusions

The usefulness of case based uncertainty estimates for Bayes classifiers was demonstrated on real world data. It was shown how detailed information about the posterior distributions of the class probabilities may allow for better informed decisions and improve classification under asymmetrical loss functions. Contrary to assumptions of previous models, Bayesian bootstrap simulations indicate that the posterior class probability distributions of Bayes classifiers are often far from normally distributed.

Acknowledgements

The authors would like to thank A. Bate and E. Swahn for helpful comments on earlier drafts of this article. In addition, the authors would like to thank those who contributed the data sets used in the empirical studies (please see the documentation of the UCI machine learning repository for details).

References

Bishop, C. M.: 1995, *Neural Networks for Pattern Recognition*. Oxford University Press.

Blake, C. and C. Merz: 1998, 'UCI Repository of machine learning databases'.

Domingos, P. and M. Pazzani: 1997, 'On the optimatility of the simple Bayesian classifier under zero-one loss'. *Machine learning* **29**, 103–130.

Efron, B.: 1979, 'Bootstrap methods: Another look at the jackknife'. *Annals of Statistics* **7**, 1–26.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin: 1995, *Bayesian Data Analysis*. Chapman & Hall.

Hand, D. J. and K. Yu: 2001, 'Idiot's Bayes—not so stupid after all?'. *International Statistical Review* **69**(3), 385–398.

Kohavi, R.: 1995, 'A study of cross-validation and bootstrap for accuracy estimation and model selection'. In: *International joint conference on Artificial Intelligence (IJCAI)*. pp. 1137–1145.

Kononenko, I.: 1990, 'Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition'. In: B. Wielinga, J. Boose, B. Gaines, G. Schreiber, and M. van Someren (eds.): *Current trends in knowledge acquisition*. IOS Press.

Kononenko, I.: 1991, 'Semi-naive Bayesian classifier'. In: *Proceedings of the Sixth European Working Session on Learning*. pp. 206–219, Springer.

MacKay, D. J.: 1992, 'The evidence framework applied to classification networks'. *Computation and neural systems* **4**(5), 698–714.

Mitchell, T. M.: 1997, *Machine Learning*. McGraw-Hill, 1 edition.

Orre, R. and A. Lansner: 1996, 'Pulp quality modelling using Bayesian mixture density neural networks'. *Journal of Systems Engineering* **6**, 128–136.

Orre, R., A. Lansner, A. Bate, and M. Lindquist: 2000, 'Bayesian neural networks with confidence estimations applied to data mining'. *Computational Statistics & Data Analysis* **34**, 473–493.

Rubin, D. B.: 1981, 'The Bayesian bootstrap'. *Annals of Statistics* **9**(1), 130–134.