Compression-Based Averaging of Selective Naive Bayes Classifiers

Marc Boullé MARC.BOULLE@ORANGE-FTGROUP.COM
France Telecom R&D
2, avenue Pierre Marzin
22300 Lannion, France

Editors: Isabelle Guyon and Amir Saffari

Abstract

The naive Bayes classifier has proved to be very effective on many real data applications. Its performance usually benefits from an accurate estimation of univariate conditional probabilities and from variable selection. However, although variable selection is a desirable feature, it is prone to overfitting. In this paper, we introduce a Bayesian regularization technique to select the most probable subset of variables compliant with the naive Bayes assumption. We also study the limits of Bayesian model averaging in the case of the naive Bayes assumption and introduce a new weighting scheme based on the ability of the models to conditionally compress the class labels. The weighting scheme on the models reduces to a weighting scheme on the variables, and finally results in a naive Bayes classifier with “soft variable selection”. Extensive experiments show that the compression-based averaged classifier outperforms the Bayesian model averaging scheme.

Keywords: naive Bayes, Bayesian, model selection, model averaging

1. Introduction

The naive Bayes classification approach (see Langley et al., 1992; Mitchell, 1997; Domingos and Pazzani, 1997; Hand and Yu, 2001) is based on the assumption that the variables are independent within each output label, and simply relies on the estimation of univariate conditional probabilities. The evaluation of the probabilities for numeric variables has already been discussed in the literature (see Dougherty et al., 1995; Liu et al., 2002; Yang and Webb, 2002). Experiments demonstrate that even a simple equal width discretization brings superior performance compared to the assumption using a Gaussian distribution.

The naive independence assumption can harm the performance when violated. In order to better deal with highly correlated variables, the selective naive Bayes approach of Langley and Sage (1994) exploits a wrapper approach (see Kohavi and John, 1997) to select the subset of variables which optimizes the classification accuracy. In the method of Boullé (2006a), the area under the receiver operating characteristic (ROC) curve (see Fawcett, 2003) is used as a selection criterion and exhibits a better predictive performance than the accuracy criterion.

Although the selective naive Bayes approach performs quite well on data sets with a reasonable number of variables, it does not scale on very large data sets with hundreds of thousands of instances and thousands of variables, such as in marketing applications. The problem comes both from the search algorithm, whose complexity is quadratic in the number of the variables, and from the selection process which is prone to overfitting.

In this paper, we present a new regularization technique to compromise between the number of selected variables and the performance of the classifier. The resulting variable selection criterion is optimized owing to an efficient search heuristic whose computational complexity is $\mathrm{O}(KN\log(KN))$, where N is the number of instances and K the number of variables. We also apply the Bayesian model averaging approach of Hoeting et al. (1999) and extend it with a compression-based averaging scheme, which better accounts for the distribution of the models. We show that averaging the models turns into averaging the contribution of the variables in the case of the selective naive Bayes classifier. Finally we proceed with extensive experiments to evaluate our method.

The remainder of the paper[1] is organized as follows. Section 2 introduces the assumptions and recalls the principles of the naive Bayes and selective naive Bayes classifiers. Section 3 presents the regularization technique for variable selection based on Bayesian model selection and Section 4 applies the Bayesian model averaging method to selective naive Bayes classifiers. In Section 5, the new selective naive Bayes classifiers are evaluated on an illustrative example. Section 6 analyzes the limits of Bayesian model averaging and proposes a new model averaging technique based on model compression coefficients. Section 7 proceeds with extensive experimental evaluations and Section 8 reports the results obtained in the performance prediction challenge organized by Guyon et al. (2006c). Finally, Section 9 concludes this paper and outlines research directions.

2. Selective Naive Bayes Classifier

This section formally states the assumptions and notations and recalls the naive Bayes and selective naive Bayes approaches.

2.1 Assumptions and Notation

Let $X = (X_1, X_2, \dots X_K)$ be the vector of the K explanatory variables and Y the class variable. Let $\lambda_1, \lambda_2, \dots \lambda_J$ be the J class labels of Y.

Let N be the number of instances and $D = \{D_1, D_2, \dots, D_N\}$ the labeled database containing the instances $D_n = (x^{(n)}, y^{(n)})$.

Let $\mathcal{M} = \{M_m\}$ be the set of all the potential selective naive Bayes models. Each model M_m is described by K parameter values a_{mk}, where a_{mk} is 1 if variable k is selected in model M_m and 0 otherwise.

Let us denote by $P(\lambda_j)$ the prior probabilities $P(Y = \lambda_j)$ of the class values, and by $P(X_k|\lambda_j)$ the conditional probability distributions $P(X_k|Y = \lambda_j)$ of the explanatory variables given the class values.

We assume that the prior probabilities $P(\lambda_j)$ and the conditional probability distributions $P(X_k|\lambda_j)$ are known, once the preprocessing is performed.

In the paper, the class conditional probabilities are estimated using the MODL discretization method of Boullé (2006c) for the numeric variables and the MODL grouping method of Boullé (2005a,b) for the categorical variables. MODL stands for minimum optimized description length and refers to the principle of minimum description length (MDL) of Rissanen (1978) as a model selection technique. More specifically, the MODL preprocessing methods exploit a maximum a posteriori (MAP) technique (see Robert, 1997) to select the most probable model of discretization

[1] This paper is an extended version of the 2006 IJCNN conference paper (Boullé, 2006b).

(resp. value grouping) given the input data. The choice of the prior distribution of the models is *optimized* for the task of data preparation, and the search algorithms are deeply *optimized*.

Using the Bayes optimal MODL preprocessing methods to estimate the conditional probabilities has proved to be very efficient in detecting irrelevant variables (see Boullé, 2006a). In the experimental section, the $P(\lambda_j)$ are estimated by counting and the $P(X_k|\lambda_j)$ are computed using the contingency tables, resulting from the preprocessing of the explanatory variables. The conditional probabilities are estimated using a m-estimate $(support + mp)/(coverage + m)$ with $m = J/N$ and $p = 1/J$, in order to avoid zero probabilities.

2.2 Naive Bayes Classifier

The naive Bayes classifier assigns to each instance the class value having the highest conditional probability

$$P(\lambda_j|X) = \frac{P(\lambda_j)P(X|\lambda_j)}{P(X)}.$$

Using the assumption that the explanatory variables are independent conditionally to the class variable, we get

$$P(\lambda_j|X) = \frac{P(\lambda_j)\prod_{k=1}^{K} P(X_k|\lambda_j)}{P(X)}. \tag{1}$$

In classification problems, Equation (1) is sufficient to predict the most probable class given the input data, since $P(X)$ is constant. In problems where a prediction score is needed, the class conditional probability can be estimated using

$$P(\lambda_j|X) = \frac{P(\lambda_j)\prod_{k=1}^{K} P(X_k|\lambda_j)}{\sum_{i=1}^{J} P(\lambda_i)\prod_{k=1}^{K} P(X_k|\lambda_i)}. \tag{2}$$

The naive Bayes classifier is poor at predicting the true class conditional probabilities, since the independence assumption is usually violated in real data applications. However, Hand and Yu (2001) show that the prediction score given by Equation (2) often provides an effective ranking of the instances for each class value.

2.3 Selective Naive Bayes Classifier

The selective naive Bayes classifier reduces the strong bias of the naive independence assumption, owing to variable selection. The objective is to search among all the subsets of variables, in order to find the best possible classifier, compliant with the naive Bayes assumption.

Langley and Sage (1994) propose to evaluate the selection process with the accuracy criterion, estimated on the train data set. However, this criterion suffers from some limits, even when the predictive performance is the only concern. In case of a skewed distribution of class labels for example, the accuracy may never be better than the majority accuracy, so that the selection process ends with an empty set of variables. This problem also arises when several consecutive selected variables are necessary to improve the accuracy. In the method proposed by Langley and Sage (1994), the selection process is iterated as long as there is no decay in the accuracy. This solution raises new problems, such as the selection of irrelevant variables with no effect on accuracy, or even the selection of redundant variables with either insignificant effect or no effect on accuracy.

Provost et al. (1998) propose to use receiver operating characteristic (ROC) analysis rather than the accuracy to evaluate induction models. This ROC criterion, estimated on the train data set (as in Langley and Sage, 1994), is used by Boullé (2006a) to assess the quality of variable selection for naive Bayes classifier. The method exploits the forward selection algorithm to select the variables, starting from an empty subset of variables. At each step of the algorithm, the variable which brings the best increase of the area under the ROC curve (AUC) is chosen and the selection process stops as soon as this area does not rise anymore. This allows capturing slight enhancements in the learning process and helps avoiding the selection of redundant variables or probes that have no effect on the ROC curve.

Altogether, the variable selection method can be implemented in $O(K^2 N \log N)$ time. The preprocessing step needs $O(KN \log N)$ to discretize or group the values of all the variables. The forward selection process requires $O(K^2 N \log N)$ time, owing to the decomposability of the naive Bayes formula on the variables. The $O(N \log N)$ term in the complexity is due to the evaluation of the area under the ROC curve, based on the sort of the training instances.

3. MAP Approach for Variable Selection

After introducing the aim of regularization, this section applies the Bayesian approach to derive a new evaluation criterion for variable selection and presents the search algorithm used to optimize this criterion.

3.1 Introduction

The naive Bayes classifier is a very robust algorithm. It can hardly overfit the data, since no hypothesis space is explored during the learning process. On the opposite, the selective naive Bayes classifier explores the space of all subsets of variables to reduce the strong bias of the naive independence assumption. The size of the searched hypothesis space grows exponentially with the number of variables, which might cause overfitting. Experiments show that during the variable selection process, the last added variables raise the "complexity" of the classifier while having an insignificant impact on the evaluation criterion (AUC for example). These slight improvements during the training step, which have an insignificant impact on the test performance, are detrimental to the ease of deployment of the models and to their understandability.

We propose to tackle this overfitting problem by relying on a Bayesian approach, where the MAP model is found by maximizing the probability $P(Model|Data)$ of the model given the data. In the following, we describe how we compute the likelihood of the models $P(Data|Model)$ and propose a prior distribution $P(Model)$ for variable selection.

3.2 Likelihood of Models

For a given model M_m parameterized by the set of selected variable indicators $\{a_{mk}\}$, the estimation of the class conditional probability $P_m(\lambda_j|X)$ turns into

$$\begin{aligned} P_m(\lambda_j|X) &= \frac{P(\lambda_j)\prod_{k=1}^{K} P(X_k|\lambda_j)^{a_{mk}}}{P(X)} \\ &= \frac{P(\lambda_j)\prod_{k=1}^{K} P(X_k|\lambda_j)^{a_{mk}}}{\sum_{i=1}^{J} P(\lambda_i)\prod_{k=1}^{K} P(X_k|\lambda_i)^{a_{mk}}}. \end{aligned} \tag{3}$$

Equation (3) provides the class conditional probability distribution for each model M_m on the basis of the parameter values a_{mk} of the model. For a given instance D_n, the probability of observing the class value $y^{(n)}$ given the explanatory values $x^{(n)}$ and given the model M_m is $P_m(Y = y^{(n)}|X = x^{(n)})$. The likelihood of the model is obtained by computing the product of these quantities on the whole data set. The negative log-likelihood of the model is given by

$$-\log P(D|M_m) = \sum_{n=1}^{N} -\log P_m(Y = y^{(n)}|X = x^{(n)}).$$

3.3 Prior for Variable Selection

The parameters of a variable selection model M_m are the Boolean values a_{mk}. We propose a hierarchic prior, by first choosing the number of selected variables and second choosing the subset of selected variables.

For the number K_m of variables, we propose to use a uniform prior between 0 and K variables, representing $(K+1)$ equiprobable alternatives.

For the choice of the K_m variables, we assign the same probability to every subset of K_m variables. The number of combinations $\binom{K}{K_m}$ seems the natural way to compute this prior, but it has the disadvantage of being symmetric. Beyond $K/2$ variables, every new variable makes the selection more probable. Thus, adding irrelevant variables is favored, provided that this has an insignificant impact on the likelihood of the model. As we prefer simpler models, we propose to use the number of combinations with replacement $\binom{K+K_m-1}{K_m}$.

Taking the negative log of this prior, we get the following code length $l(M_m)$ for the variable selection models

$$l(M_m) = \log(K+1) + \log\binom{K+K_m-1}{K_m}.$$

Using this prior, the "informational cost" of the first selected variables is about $\log K$ and about $\log 2$ for the last variables.

To summarize our prior, each number of K_m variable is equiprobable, and for a given K_m, each subset of K_m variables randomly chosen with replacement is equiprobable. This means that each specific small subset of variables has a greater probability than each specific large subset of variables, since the number of variable subsets of given size grows with K_m.

3.4 Posterior Distribution of the Models

The posterior probability of a model M_m is evaluated as the product of the prior and the likelihood. This is equivalent to the MDL approach of Rissanen (1978), where the code length of the model plus the data given the model has to be minimized:

$$l(M_m) + l(D|M_m) = \log(K+1) + \log\binom{K+K_m-1}{K_m} - \sum_{n=1}^{N} \log P_m(y^{(n)}|X = x^{(n)}). \tag{4}$$

The first two terms encode the complexity of the model and the last one the fit of the data. The compromise is found by minimizing this criterion.

We can notice a trend of increasing attention to the predicted probabilities in the evaluation criteria proposed for variable selection. Whereas the accuracy criterion focuses only on the majority class and the area under the ROC curve evaluates the correct ordering of the predicted probabilities, our regularized criterion evaluates the correctness of all the predicted probabilities (not only their rank) and introduces a regularization term to balance the complexity of the models.

3.5 An Efficient Search Heuristic

Many heuristics have been used for variable selection (see Guyon et al., 2006b). The greedy forward selection heuristic evaluates all the variables, starting from an empty set of variables. The best variable is added to the current selection, and the process is iterated until no new variable improves the evaluation criterion. This heuristic may fall in local optima and has a quadratic time complexity with respect to the number of variables. The forward backward selection heuristic allows to add or drop one variable at each step, in order to avoid local optima. The fast forward selection heuristic evaluates each variable one at a time, and adds it to the selection as soon as this improves the criterion. This last heuristic is time effective, but its results exhibit a large variance caused by the dependence over the order of the variables.

Algorithm 1 Algorithm MS(FFWBW)

```
Require: $X \leftarrow (X_1, X_2, \ldots X_K)$ {Set of input variables}
Ensure: $B$ {Best subset of variables}
$B \leftarrow \emptyset$ {Start with an empty subset of variables}
for Step=1 to $\log_2 KN$ do
    {Fast forward backward selection}
    $S \leftarrow \emptyset$ {Initialize an empty subset of variables}
    $Iter \leftarrow 0$
    repeat
        $Iter \leftarrow Iter + 1$
        $X' \leftarrow$ Shuffle$(X)$ {Randomly reorder the variables to add}
        {Fast forward selection}
        for $X_k \in X'$ do
            if $cost(S \cup \{X_k\}) < cost(S)$ then
                $S \leftarrow S \cup \{X_k\}$
            end if
        end for
        $X' \leftarrow$ Shuffle$(X)$ {Randomly reorder the variables to remove}
        {Fast backward selection}
        for $X_k \in X'$ do
            if $cost(S - \{X_k\}) < cost(S)$ then
                $S \leftarrow S - \{X_k\}$
            end if
        end for
    until no improvement or $Iter \geq MaxIter$
    {Update best subset of variables}
    if $cost(S) < cost(B)$ then
        $B \leftarrow S$
    end if
end for
```

We introduce a new search heuristic called fast forward backward selection (FFWBW), based on a mix of the preceding approaches. It consists in a sequence of fast forward selection and fast backward selection steps. The variables are randomly reordered between each step, and evaluated only once during each forward or backward search. This process is iterated as long as two successive (forward and backward) search steps bring at least one improvement of the criterion or when the iteration number exceeds a given parameter $MaxIter$. In practice, the whole process converges very quickly, in one or two steps in most of the cases. Setting $MaxIter = 5$ for example is sufficient to bound the worst case complexity without decreasing the quality of the search algorithm.

Evaluating a selective naive Bayes model requires $\mathrm{O}(KN)$ computation time, mainly to evaluate all the class conditional probabilities. According to Equation (3), these class conditional probabilities can be updated in $\mathrm{O}(1)$ per instance and $\mathrm{O}(N)$ for the whole data set when one variable is added or removed from the current subset of selected variables. Each fast forward selection or fast backward selection step considers $\mathrm{O}(K)$ additions or removals of variables and requires $\mathrm{O}(KN)$ computation time. The total time complexity of the FFWBW heuristic is $\mathrm{O}(KN)$, since the number of search steps is bounded by the constant parameter $MaxIter$.

In order to further reduce both the possibility of local optima and the variance of the results, this FFWBW heuristic is embedded into a multi-start (MS) algorithm, by repeating the search heuristic starting from several random orderings of the variables. The number of repetitions is set to $\log_2 KN$, which offers a reasonable compromise between time complexity and quality of the optimization. Overall, the time complexity of the MS(FFWBW) heuristic is $\mathrm{O}(KN \log KN)$. The heuristic is detailed in Algorithm 1.

4. Bayesian Model Averaging of Selective Naive Bayes Classifiers

Model averaging consists in combining the prediction of an ensemble of classifiers in order to reduce the prediction error. This section reminds the principles of Bayesian model averaging and applies this averaging scheme to the selective naive Bayes classifier.

4.1 Bayesian Model Averaging

The Bayesian model averaging (BMA) method (Hoeting et al., 1999) aims at accounting for the model uncertainty. Whereas the MAP approach retrieves the most probable model given the data, the BMA approach exploits every model in the model space, weighted by their posterior probability. This approach relies on the definition of a prior distribution on the models, on an efficient computation technique to estimate the model posterior probabilities and on an effective method to sample the posterior distribution. Apart from these technical difficulties, the BMA approach is an appealing technique, with strong theoretical results concerning the optimality of its long-run performance, as shown by Raftery and Zheng (2003).

The BMA approach has been applied to the naive Bayes classifier by Dash and Cooper (2002). Apart from the differences in the weighting scheme, their method (DC) differs from ours mainly on the initial assumptions. The DC method does not manage the numeric variables and assumes multinomial distributions with Dirichlet priors for the categorical variables, which requires the choice of hyper-parameters for each variable. Structure modularity of the Bayesian network is also assumed: each selection of a variable is independent from the others. The DC approach estimates the full data distribution (explanatory and class variables), whereas we focus on the class conditional probabilities. Once the prior hyper-parameters are fixed, the DC method allows to compute an exact model averaging, whereas we rely on an heuristic to estimate the averaged model. Compared to the DC method, our method is not restricted to categorical attributes and does not need any hyper-parameter.

4.2 From Bayesian Model Averaging to Expectation

For a given variable of interest Δ, the BMA approach averages the predictions of all the models weighted by their posterior probability.

$$P(\Delta|D) = \sum_m P(\Delta|M_m, D) P(M_m|D).$$

This formula can be written, using only the prior probabilities and the likelihood of the models.

$$P(\Delta|D) = \frac{\sum_m P(\Delta|M_m, D) P(M_m) P(D|M_m)}{\sum_m P(M_m) P(D|M_m)}.$$

Let $f(M_m, D) = P(\Delta|M_m, D)$ and $f(D) = P(\Delta|D)$. Using these notations, the BMA formula can be interpreted as the expectation of function f for the posterior distribution of the models

$$E(f) = \sum_m f(M_m, D) P(M_m|D).$$

We propose to extend the BMA approach in the case where f is not restricted to be a probability function.

4.3 Expectation of the Class Conditional Information

The selective naive Bayes classifier provides an estimation of the class conditional probabilities. These estimated probabilities are the natural candidates for averaging. For a given model M_m defined by the variable selection $\{a_{mk}\}$, we have

$$f(M_m, D) = \frac{P(Y) \prod_{k=1}^{K} P(X_k|Y)^{a_{mk}}}{P(X)}. \tag{5}$$

Let $I(M_m, D) = -\log f(M_m, D)$ be the class conditional information. Whereas the expectation of f relates to a (weighted) arithmetic mean of the class conditional probabilities, the expectation of I relates to a (weighted) geometric mean of these probabilities. This puts more emphasis on the magnitude of the estimated probabilities. Taking the negative log of (5), we obtain

$$I(M_m, D) = I(Y) - I(X) + \sum_{k=1}^{K} a_{mk} I(X_k|Y). \tag{6}$$

We are looking for the expectation of this conditional information

$$\begin{aligned} E(I) &= \frac{\sum_m I(M_n, D) P(M_m|D)}{\sum_m P(M_m|D)} \\ &= I(Y) - I(X) + \frac{\sum_m P(M_m|D) \sum_{k=1}^{K} a_{mk} I(X_k|Y)}{\sum_m P(M_m|D)} \\ &= I(Y) - I(X) + \sum_{k=1}^{K} I(X_k|Y) \frac{\sum_m a_{mk} P(M_m|D)}{\sum_m P(M_m|D)}. \end{aligned}$$

Let $b_k = \frac{\sum_m a_{mk} P(M_m|D)}{\sum_m P(M_m|D)}$. We have $b_k \in [0,1]$.

The b_k coefficients are computed using (4), on the basis of the prior probabilities and of the likelihood of the models. Using these coefficients, the expectation of the conditional information is

$$E(I) = I(Y) - I(X) + \sum_{k=1}^{K} b_k I(X_k|Y). \tag{7}$$

The averaged model thus provides the following estimation for the class conditional probabilities:

$$P(Y|X) = \frac{P(Y)\prod_{k=1}^{K} P(X_k|Y)^{b_k}}{P(X)}.$$

It is noteworthy that the expectation of the conditional information in (7) is similar to the conditional information estimated by each individual model in (6). The weighting scheme on the models reduces to a weighting scheme on the variables. When the MAP model is selected, the variables have a weight of 1 when selected and 0 otherwise: this is a "hard selection" of the variables. When the above averaging scheme is applied, each variable has a $[0,1]$ weight, which can be interpreted as a "soft selection".

4.4 An Efficient Algorithm for Model Averaging

We have previously introduced a model averaging method which relies on the expectation of the class conditional information. The calculation of this expectation requires the evaluation of all the variable selection models, which is not computationally feasible as soon as the number of variables goes beyond about 20. This expectation can heuristically be evaluated by sampling the posterior distribution of the models and accounting only for the sampled models in the weighting scheme.

We propose to reuse the MS(FFWBW) search heuristic to perform this sampling. This heuristic is effective for finding high probability models and searching in their neighborhood. The repetition of the search from several random starting points (in the multi-start meta-heuristics) brings diversity and allows to escape local optima. We use the whole set of models evaluated during the search to estimate the expectation.

Although this sampling strategy is biased by the search heuristic, it has the advantage of being simple and computationally tractable. The overhead in the time complexity of the learning algorithm is negligible, since the only need is to collect the posterior probability of the models and to compute the weights in the averaging formula. Concerning the deployment of the averaged model, the overhead is also negligible, since the initial naive Bayes estimation of the class conditional probabilities is just extended with variable weights.

5. Evaluation on an Illustrative Example

This section describes the waveform data set, introduces three evaluation criteria and illustrates the behavior of each variant of the selective naive Bayes classifier.

5.1 The Waveform Data Set

The waveform data set introduced by Breiman et al. (1984) contains 5000 instances, 21 continuous variables and 3 equidistributed class values. Each instance is defined as a linear combination of two out of the three triangular waveforms pictured in Figure 1, with randomly generated coefficients and Gaussian noise. Figure 2 plots 10 random instances from each class.

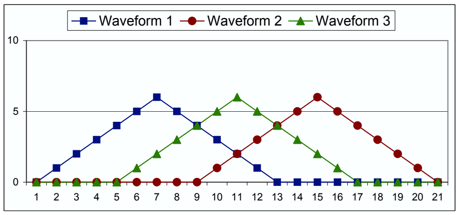

Figure 1: Basic waveforms used to generated the 21 input variables of the waveform data set.

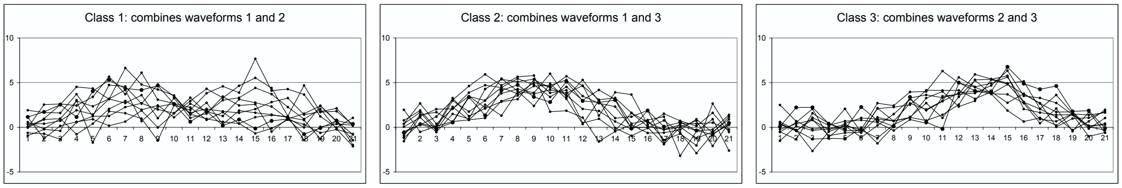

Figure 2: Waveform data.

Learning on the waveform data set is generally considered a difficult task in pattern recognition, with reported accuracy of 86.8% using a Bayes optimal classifier. The input variables are correlated, which violates the naive Bayes assumption. Selecting the best subset of variables compliant with the naive Bayes assumption is a challenging problem.

5.2 The Evaluation Criteria

We evaluate three criteria of increasing complexity: accuracy (ACC), area under the ROC curve (AUC) and informational loss function (ILF).

The ACC criterion evaluates the accuracy of the prediction, no matter whether its conditional probability is 51% or 100%.

The AUC criterion (see Fawcett, 2003) evaluates the ranking of the class conditional probabilities. In a two-classes problem, the AUC is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Extending the AUC criterion to multi-class problems is not a trivial task and has lead to computationally intensive methods (see for example Deng et al., 2006). In our experiments, we use the approach of Provost and Domingos (2001) to calculate the multi-class AUC, by computing each one-against-the-others two-classes AUC and weighting them by the class prior probabilities $P(\lambda_j)$. Although this version of multi-class AUC is sensitive to the class distribution, it is easy to compute, which motivates our choice.

The ILF criterion (see Witten and Frank, 2000) evaluates the probabilistic prediction owing to the negative log of the predicted class conditional probabilities

$$-\log P_m(Y = y^{(n)}|X = x^{(n)}).$$

The empirical mean of the ILF criterion is equal to

$$\overline{ILF(M_m)} = \frac{1}{N}\sum_{n=1}^{N} -\log P_m(Y = y^{(n)}|X = x^{(n)}).$$

The predicted class conditional probabilities in the ILF criterion are given by Equation (2) for the naive Bayes classifier and by Equation (3) for the selective naive Bayes classifier.

Let $M_{\emptyset}$ be the "null" model, with no variable selected. The null model estimates the class conditional probabilities by their prior probabilities, ignoring all the explanatory variables. For the null model $M_{\emptyset}$, we obtain

$$\begin{aligned}\overline{ILF(M_{\emptyset})} &= \frac{1}{N}\sum_{n=1}^{N} -\log P(Y = y^{(n)}) \\ &= -\sum_{j=1}^{J} P(\lambda_j)\log P(\lambda_j) \\ &= H(Y),\end{aligned}$$

where $H(Y)$ is the entropy of Shannon (1948) of the class variable.

We introduce a compression rate to normalize the ILF criterion using

$$\begin{aligned}CR(M_m) &= 1 - \overline{ILF(M_m)}\,/\,\overline{ILF(M_{\emptyset})} \\ &= 1 - \overline{ILF(M_m)}\,/\,H(Y).\end{aligned}$$

The normalized CR criterion is mainly ranged between 0 (prediction not better than the basic prediction of the class priors) and 1 (prediction of the true class probabilities in case of perfectly separable classes). It can be negative when the predicted probabilities are worse than the basic prior predictions.

5.3 Evaluation on the Waveform Data Set

We use 70% of the waveform data set to train the classifiers and 30% to test them. These evaluations are merely illustrative; extensive experiments are reported in Section 7.

In the case of the waveform data set, the MODL preprocessing method determines that 2 variables (1^{st} and 21^{st}) are irrelevant, and the naive Bayes classifier uses all the 19 remaining variables. We evaluate four variants of selective naive Bayes methods. The SNB(ACC) method of Langley and Sage (1994) optimizes the train accuracy and the SNB(AUC) method of Boullé (2006a) optimizes the area under the ROC curve on the train data set. The SNB(MAP) method introduced in Section 3 selects the most probable subset of variables compliant with the naive Bayes assumption, and the SNB(BMA) method described in Section 4 averages the selective naive Bayes classifiers weighted by their posterior probability. In this experiment, we evaluate exhaustively the half a million models related to the 2^{19} possible variable subsets. This allows us to focus on the variable selection criterion and to avoid the potential bias of the optimization algorithms.

The selected subsets of variables are pictured in Figure 3. The SNB(ACC) method selects 12 variables and the SNB(AUC) 18 variables. The SNB(MAP) which focuses on a subset of variables compliant with the naive Bayes assumption selects only 8 variables, which turns out to be a subset of the variables selected by the alternative methods.

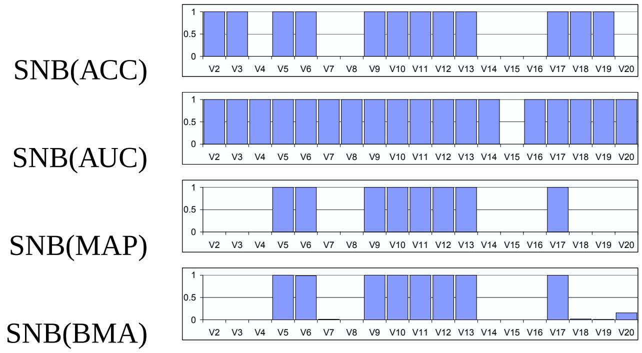

Figure 3: Variables selected by the selective naive Bayes classifiers for the waveform data set.

The predictive performance for the ACC, AUC and ILF criteria are reported in Figure 4. In multi-criteria analysis, a solution *dominates* (or is *non-inferior* to) another one if it is better for all criteria. A solution that cannot be dominated is *Pareto* optimal: any improvement of one of the criteria causes a deterioration on another criterion (see Pareto, 1906). The *Pareto surface* is the set of all the Pareto optimal solutions.

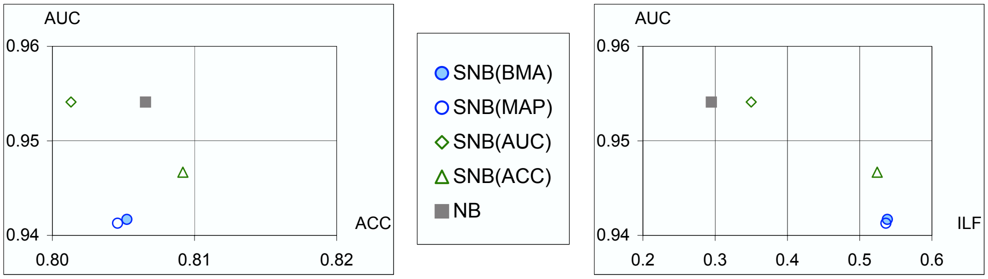

Figure 4: Evaluation of the predictive performance of selective naive Bayes classifiers on the waveform data set.

The SNB(ACC) method is slightly better than the NB method on the ACC criterion. Owing to its small subset of selected variables, it manages to reduce the redundancy between the variables and to significantly improve the estimation of the class conditional probabilities, as reported by its ILF evaluation. The SNB(AUC) method gets the same AUC performance as the NB method with one variable less. The SNB(MAP) and SNB(BMA) methods almost directly optimizes the ILF criterion on the train data set, with a regularization term related to the model prior. They get the best ILF evaluation on the test set, but are dominated by the NB and SNB(ACC) methods on the two other criteria, as shown in Figure 4. Almost all the methods are Pareto optimal: none of them is the best on the three evaluated criteria.

Compared to the other variable selection methods, the SNB(MAP) truly focuses on complying with the naive independence assumption. This results in a much smaller subset of variables and a better prediction of the class conditional probabilities, at the expense of a decay on the other criteria.

The SNB(BMA) method exploits a model averaging approach which results in soft variable selection. Figure 3 shows the weights of each variable. Surprisingly, the selected variables are almost the same as the 8 variables selected by the SNB(MAP) method. Compared to the hard variable selection scheme, the soft variable selection exhibits mainly one minor change: a new variable (V20) is selected with a small weight of 0.15. The other modifications of the variable weights are insignificant: two variables (V6 and V17) decrease their weight from 1.0 to 0.99 and three variables (V7, V18 and V19) appear with a tiny weight of 0.01.

Since the variable selection is almost the same as in the MAP approach, the model averaging approach does not bring any significant improvement in the evaluation results, as shown in Figure 4.

6. Compression Model Averaging of Selective Naive Bayes Classifiers

This section analyzes the limits of Bayesian model averaging and proposes a new weighting scheme that better exploits the posterior distributions of the models.

6.1 Understanding the Limits of Bayesian Model Averaging

We use again the waveform data set to explain why the Bayesian model averaging method fails to outperform the MAP method.

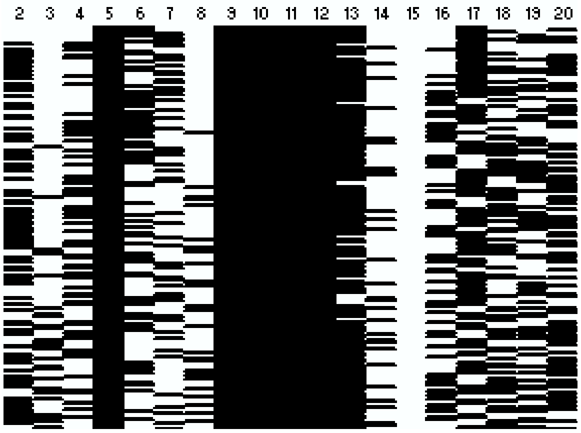

Figure 5: Index of the selected variables in the 200 most probable selective naive Bayes models for the waveform data set. Each line represents a model, where the variables are in black color when selected.

The variable selection problem consists in finding the most probable subset of variables compliant with the naive Bayes assumption among about half a million (2^{19}) potential subsets. In order to study the posterior distribution of the models, all these subsets are evaluated. The MAP model selects 8 variables (V5, V6, V9, V10, V11, V12, V13, V17). A close look at the posterior distribution shows that most of the good models (in the top 50%) contain around 10 variables. Figure 5 displays the selected variables in the top 200 models (0.05%). Five variables (V5, V9, V10, V11, V12) among the 8 MAP variables are always selected, and the other models exploit a diversity of subsets of variables. The potential benefit of model averaging is to account for all these models, with higher weights for the most probable models.

However, the posterior distribution is very sharp everywhere, not only around the MAP. Variable V18 is first selected in the 3^{rd} model, which is about 40 times less probable than the MAP model. Variable V4 is first selected in the 10^{th} model, about 4000 times less probable than the MAP model. Figure 6 displays the repartition function of the posterior probabilities, using a log scale. Using this logarithmic transformation, the posterior distribution is flattened and can be visualized. The MAP model is 10^{1033} times more probable than the minimum a posteriori model, which selects no variable.

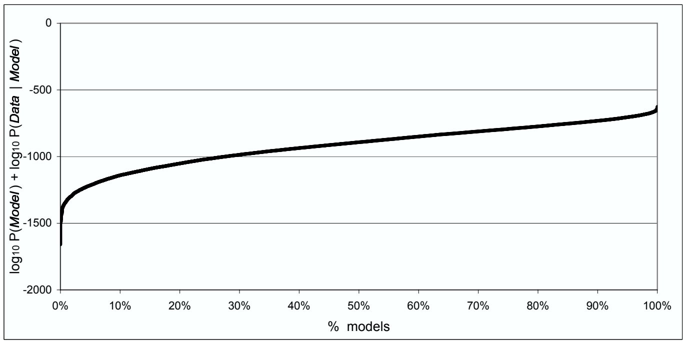

Figure 6: Repartition function of the posterior probabilities of half a million variable selection models evaluated for the waveform data set, sorted by increasing posterior probability. For example, the 10% models on the left represent the models having the lowest posterior probability.

In the waveform example, averaging using the posterior probabilities to weight the models is almost the same as selecting the MAP model (which itself is hard to find with a heuristic search) and model averaging is almost useless. In theory, BMA is optimal (see Raftery and Zheng, 2003), but this optimality result assumes that the true distribution of the data belongs to the space of models. In the case of the selective naive Bayes classifier, this assumption is violated on most real data sets and BMA fails to build effective model averaging.

6.2 Model Averaging with Compression Coefficients

When the posterior distribution is sharply peaked around the MAP, averaging is almost the same as selecting the MAP model. These peaked posterior distributions are more and more likely to happen when the number of instances rises, since a few tens of instances better classified by a model are sufficient to increase its likelihood by several orders of magnitude. Therefore, the algorithmic overhead is not valuable if averaging turns out to be the same as selecting the MAP.

In order to have a theoretical insight on the relation between MAP and BMA, let us analyze again the model selection criterion (4). It is closely related to the ILF criterion described in Section 5.2, according to

$$l(M_m) + l(D|M_m) = -\log P(M_m) + N\,\overline{ILF(M_m)}.$$

For the the “null” model $M_\emptyset$, with no variable selected, we have:

$$l(M_\emptyset)+l(D|M_\emptyset)=-\log P(M_\emptyset)+NH(Y).$$

The posterior probability $P(M_m|D)$ of a model M_m relative to that of the null model is

$$\frac{P(M_m|D)}{P(M_\emptyset|D)}=\frac{P(M_m)}{P(M_\emptyset)}\left(\frac{H(Y)}{\overline{\overline{ILF(M_m)}}}\right)^N. \tag{8}$$

Equation (8) shows that the posterior probability of the models is exponentially peaked when N goes to infinity. Small improvements in the estimation of the conditional entropy brings very large differences in the posterior probability of the models, which explains why Bayesian model averaging is asymptotically equivalent to selecting the MAP model.

We propose an alternative weighting scheme, whose objective is to better account for the set of all models. Let us precisely define the compression coefficient $c(M_m,D)$ of a model. The model selection criterion $l(M_m)+l(D|M_m)$ defined in Equation (4) represents the quantity of information required to encode the model plus the class values given the model. The code length of the null model $M_\emptyset$ can be interpreted as the quantity of information necessary to describe the classes, when no explanatory data is used to induce the model.

Each model M_m can potentially exploit the explanatory data to better “compress” the class conditional information. The ratio of the code length of a model to that of the null model stands for a relative gain in compression efficiency. We define the compression coefficient $c(M_m,D)$ of a model as follows:

$$c(M_m,D)=1-\frac{l(M_m)+l(D|M_m)}{l(M_\emptyset)+l(D|M_\emptyset)}.$$

The compression coefficient is 0 for the null model, is maximal when the true class conditional probabilities are correctly estimated and tends to 1 in case of separable classes. This coefficient can be negative for models which provide an estimation worse than that of the null model.

In our heuristic attempt to better account for all the models, we replace the posterior probabilities by their related compression coefficient in the weighting scheme.

Let us focus again on the variable weights b_k introduced in Section 4 in our first model averaging method. Dividing the posterior probabilities by those of the null model, we get

$$b_k=\frac{\sum_m a_{mk}\frac{P(M_m|D)}{P(M_\emptyset|D)}}{\sum_m \frac{P(M_m|D)}{P(M_\emptyset|D)}}.$$

We introduce new c_k coefficients by taking the log of the probability ratios and normalizing by the code length of the null model. We obtain

$$c_k=\frac{\sum_m a_{mk}c(M_m,D)}{\sum_m c(M_m,D)}.$$

Mainly, the principle of this new heuristic weighting scheme consists in smoothing the exponentially peaked posterior probability distribution of Equation (8) with the log function.

In the implementation, we ignore the “bad” models and consider the positive compression coefficients only. We evaluate the compression based model averaging (CMA) model using the model averaging algorithm introduced in Section 4.4.

6.3 Evaluation on the Waveform Data Set

We use the protocol introduced in Section 5 to evaluate the SNB(CMA) compression model averaging method on the waveform data set, with an exhaustive evaluation of all the models to avoid the potential bias of the optimization algorithms.

Figure 7 shows the weights of each variable resulting from the soft variable selection of the SNB(CMA) compression model averaging method. Contrary to the SNB(BMA) method, the averaging has a significant impact on variable selection.

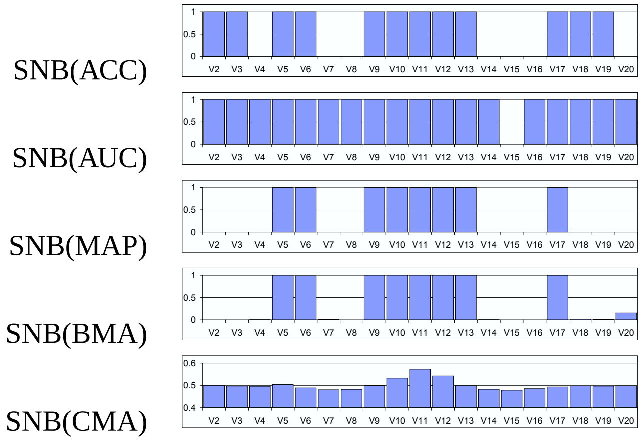

Figure 7: Variables selected by the SNB(CMA) method and the alternative selective naive Bayes classifiers for the waveform data set.

Instead of "hard selecting" about half of the variables as in the SNB(MAP) method, the SNB(CMA) method selects all the variable with weights around 0.5. Interestingly, the variable selection pattern is similar to that of the alternative variable selection methods, in a smoothed version. A central group of variables is emphasized around variable V11, between two less important groups of variables around variables V5 and V17.

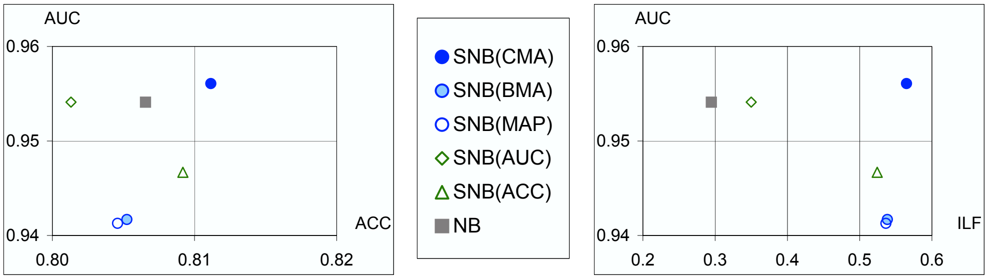

Figure 8: Evaluation of the predictive performance of selective naive Bayes classifiers on the waveform data set.

In the waveform data set, all the variables are informative, but the most probable subsets of variables compliant with the naive Bayes assumption select only half of the variables. In other words, whereas many good SNB classifiers are available, none of them is able to account for all the information contained in the variables. Since the BMA model is almost the same as the MAP model, it fails to perform better than one single classifier . Our CMA approach averages complementary subsets of variables and exploits more information than the BMA approach. This smoothed variable selection results in improved performance, as shown in Figure 8. The SNB(CMA) method is the best one: it dominates all the other methods on the three evaluated criteria.

7. Experiments

This section presents an experimental evaluation of the performance of the selective naive Bayes methods described in the previous sections.

7.1 Experimental Setup

The experiments aim at comparing the performance of model averaging methods versus the MAP method, the standard selective naive Bayes (SNB) and naive Bayes (NB) methods. All the classifiers except the last one exploit the same MODL preprocessing, allowing a fair comparison. The evaluated methods are:

- No variable selection
  - NB(EF): NB with 10 bins equal frequency discretization and no value grouping,
  - NB: NB with MODL preprocessing,
- Variable selection
  - SNB(ACC): optimization of the accuracy,
  - SNB(AUC): optimization of the area under the ROC curve,
  - SNB(MAP): MAP SNB model,
- Variable selection and model averaging
  - SNB(BMA): Bayesian model averaging,
  - SNB(CMA)[2] : compression-based model averaging.

The three last SNB classifiers (SNB(MAP), SNB(BMA) and SNB(CMA)) represent our contribution in this paper. All the SNB classifiers are optimized with the same MS(FFWBW) search heuristic, except the SNB(ACC), based on the forward selection greedy heuristic. The DC method (Dash and Cooper, 2002), similar to the SNB(BMA) approach, was not evaluated since it is restricted to categorical attributes.

The evaluated criteria are the same as for the waveform data set: accuracy (ACC), area under the ROC curve (AUC) and informational loss function (ILF) (with its compression rate (CR) normalization).

[2] The method is implemented in a tool available as a shareware at http://www.francetelecom.com/en/group/rd/offer/software/technologies/middlewares/khiops.html.

Name	Instances	Numerical variables	Categorical variables	Classes	Majority accuracy
Abalone	4177	7	1	28	16.5
Adult	48842	7	8	2	76.1
Australian	690	6	8	2	55.5
Breast	699	10	0	2	65.5
Crx	690	6	9	2	55.5
German	1000	24	0	2	70.0
Glass	214	9	0	6	35.5
Heart	270	10	3	2	55.6
Hepatitis	155	6	13	2	79.4
HorseColic	368	7	20	2	63.0
Hypothyroid	3163	7	18	2	95.2
Ionosphere	351	34	0	2	64.1
Iris	150	4	0	3	33.3
LED	1000	7	0	10	11.4
LED17	10000	24	0	10	10.7
Letter	20000	16	0	26	04.1
Mushroom	8416	0	22	2	53.3
PenDigits	7494	16	0	10	10.4
Pima	768	8	0	2	65.1
Satimage	6435	36	0	6	23.8
Segmentation	2310	19	0	7	14.3
SickEuthyroid	3163	7	18	2	90.7
Sonar	208	60	0	2	53.4
Spam	4307	57	0	2	64.7
Thyroid	7200	21	0	3	92.6
TicTacToe	958	0	9	2	65.3
Vehicle	846	18	0	4	25.8
Waveform	5000	21	0	3	33.9
Wine	178	13	0	3	39.9
Yeast	1484	8	1	10	31.2

Table 1: UCI Data Sets

We conduct the experiments on two collections of data sets: 30 data sets from the repository at University of California at Irvine (Blake and Merz, 1996) and 10 data sets from the NIPS 2003 feature selection challenge (Guyon et al., 2006a) and the IJCNN 2006 performance prediction challenge (Guyon et al., 2006c). A summary of some properties of these data sets is given in Table 1 for the UCI data sets and in Table 2 for the challenge data sets. We use stratified 10-fold cross validation to evaluate the criteria. A two-tailed Student test at the 5% confidence level is performed in order to evaluate the significant wins or losses of the SNB(CMA) method versus each other method.

7.2 Results

We collect and average the three criteria owing to the stratified 10-fold cross validation, for the seven evaluated methods on the forty data sets. The results are presented in Table 3 for the UCI data

Name	Instances	Numerical variables	Categorical variables	Classes	Majority accuracy
Arcene	200	10000	0	2	56.0
Dexter	600	20000	0	2	50.0
Dorothea	1150	100000	0	2	90.3
Gisette	7000	5000	0	2	50.0
Madelon	2600	500	0	2	50.0
Ada	4147	48	0	2	75.2
Gina	3153	970	0	2	50.8
Hiva	3845	1617	0	2	96.5
Nova	1754	16969	0	2	71.6
Sylva	13086	216	0	2	93.8

Table 2: Challenge Data Sets

sets and in Table 4 for the challenge data sets. They are summarized across the data sets using the mean, the number of wins and losses (W/L) for the SNB(CMA) method and the average rank, for each of the three evaluation criteria.

Method		ACC			AUC			CR	
	Mean	W/L	Rank	Mean	W/L	Rank	Mean	W/L	Rank
SNB(CMA)	0.824		2.2	0.920		1.9	0.577		2.2
SNB(BMA)	0.817	9/2	3.7	0.916	11/0	3.3	0.559	12/6	2.6
SNB(MAP)	0.813	11/1	4.5	0.913	17/1	4.4	0.549	15/6	3.6
SNB(AUC)	0.820	8/0	3.3	0.918	10/2	3.1	0.532	17/4	4.4
SNB(ACC)	0.817	5/1	3.5	0.910	14/0	4.5	0.536	13/2	4.5
NB	0.814	11/0	4.0	0.913	16/1	4.6	0.476	19/2	5.3
NB(EF)	0.796	15/1	4.6	0.911	13/3	4.8	0.401	15/2	5.3

Table 3: Evaluation of the methods on the UCI data sets

The three ways of aggregating the results (mean, W/L and rank) are consistent, and we choose to display the mean of each criterion to ease the interpretation. Figure 9 summarizes the results for the UCI data sets and Figure 10 for the challenge data sets.

The results of the two NB methods are reported mainly as a sanity check. The MODL preprocessing in the NB classifier exhibits better performance than the equal frequency discretization method in the NB(EF) classifier.

The experiments confirm the benefit of selecting the variables, using the standard selection methods SNB(ACC) and SNB(AUC). These two methods achieve comparable results, with an emphasis on their respective optimized criterion. They significantly improve the results of the NB methods, especially for the estimation of class conditional probabilities measured by the CR criterion. It is noteworthy that the NB and NB(EF) classifiers obtain poor CR results. Their mean CR

Method		ACC			AUC			CR	
	Mean	W/L	Rank	Mean	W/L	Rank	Mean	W/L	Rank
SNB(CMA)	0.883		1.9	0.904		1.0	0.510		1.0
SNB(BMA)	0.872	3/0	3.5	0.882	6/0	2.9	0.446	9/0	2.3
SNB(MAP)	0.865	4/0	4.5	0.863	6/0	5.1	0.425	9/0	3.7
SNB(AUC)	0.872	6/0	3.6	0.888	7/0	2.7	0.331	10/0	4.1
SNB(ACC)	0.875	3/2	2.9	0.869	8/0	4.8	0.365	9/0	4.3
NB	0.841	7/0	4.9	0.846	9/0	5.3	-0.321	9/0	5.9
NB(EF)	0.823	9/0	6.6	0.833	9/0	6.2	-0.423	10/0	6.7

Table 4: Evaluation of the methods on the challenge data sets

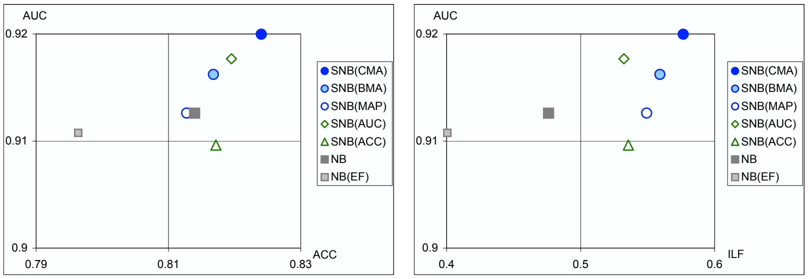

Figure 9: Mean of the ACC, AUC and CR evaluation criteria on the 30 UCI data sets.

result is less than 0 in the case of the challenge data sets, which means that their estimation of the class conditional probabilities is worse than that of the null model (which selects no variable).

The three regularized methods SNB(MAP), SNB(BMA) and SNB(CMA) focus on the estimation of the class conditional probabilities, which are evaluated using the compression rate criterion. They clearly outperform the other methods on this criterion, especially for the challenge data sets where the improvement amounts to about 50%. However, the SNB(MAP) method is not better than the two standard SNB methods for the accuracy and AUC criteria. The MAP method increases the bias of the models by penalizing the complex models, leading to a decayed fit of the data.

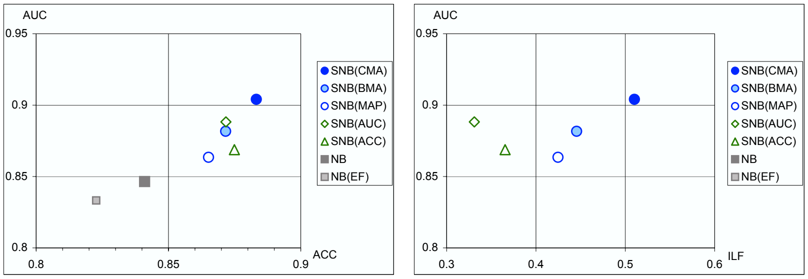

Figure 10: Mean of the ACC, AUC and CR evaluation criteria on the 10 challenge data sets.

The model averaging approach exploited in SNB(BMA) method offers only slight enhancements compared to the SNB(MAP) method. This confirms the analysis drawn from the waveform case study.

The compression-based averaging method SNB(CMA) clearly dominates all the other methods on all the criteria. On average, the number of significant wins is about 10 times the number of significant losses, and amounts to more than half of the 40 data sets. On the 10 challenge data sets, having very large numbers of variables, the SNB(CMA) method always gets the best results on the AUC and CR criteria, and almost always on the accuracy criterion. The domination of the SNB(CMA) method increases with the complexity of the criteria: it is noteworthy for accuracy (ACC), important for the ordering of the class conditional probabilities (AUC) and very large for the prediction of the class conditional probabilities (CR). This shows that the regularized and averaged naive Bayes classifier becomes effective for conditional probability estimation, whereas the standard naive Bayes classifier is usually considered to be poor at estimating these probabilities.

To summarize, this experiment demonstrates that variable selection is useful to improve the classification accuracy of the naive Bayes classifier. The MAP selection approach presented in Section 3 allows to find a selective naive Bayes classifier which is as compliant as possible with the naive Bayes assumption. Although this has little impact on the classification accuracy, this greatly improves the estimation of the class conditional probabilities.

Model averaging aims at improving the predictive performance at the expense of models which are more difficult to understand and to deploy. From this point of view, the experiment indicates that Bayesian model averaging is not much useful, since it does not significantly outperform the MAP model. On the opposite, our compression model averaging scheme introduced in Section 6 takes benefit from the full posterior distribution of the models and obtains superior results for all the evaluated criteria.

8. Evaluation on the Performance Prediction Challenge

This section reports the results obtained by our compression-based averaging method on the performance prediction challenge of Guyon et al. (2006c).

8.1 The Performance Prediction Challenge

The purpose of the performance prediction challenge is “to stimulate research and reveal the state-of-the art in model selection”. Each method is evaluated according to its predictive performance and to its ability to guess its performance. The performance is assessed using the balanced error rate (BER) criterion to account for skewed distributions. The BER guess error is evaluated as the absolute value of the difference between the test BER and the predicted BER. A test score is computed as a combination of the test BER and the BER guess error to rank the participants.

Five data sets are used in the challenge (the five last data sets in Table 2). The ada data set comes from the marketing domain, the gina data set from handwriting recognition, the hiva data set from drug discovery, the nova data set from text classification and the sylva data set from ecology. The test sets used to assess the performance are 10 times larger than the train data sets.

8.2 Details of the Submissions

All our entries are based on the compression-based averaging of the selective naive Bayes classifier SNB(CMA).

The method computes the posterior probabilities of the classes, which is convenient when the accuracy criterion or the area under the ROC curve is evaluated. In a two-classes problem, any instance whose class posterior probability is beyond a threshold $\tau = 0.5$ is classified as positive, and otherwise as negative. For the challenge, the BER criterion is the main criterion, and it is no longer optimal to predict the most probable class. In order to improve the BER criterion, we adjust the decision threshold τ in a post-optimization step. We sort the train instances by decreasing class posterior probabilities, which determines N possible values of the threshold τ. We then loop on the instances, and for each τ, we compute the confusion matrix between the prediction outcome and the actual class value. We keep the threshold that maximizes the BER of the related confusion matrix. Post-optimizing the BER criterion requires $0(N \log N)$ computation time to sort the instances and $\mathrm{O}(N)$ to compute the N possible confusion matrices, since evaluating each successive τ involves the move of only one instance in the confusion matrix.

For the challenge, we perform several trials of feature construction in order to evaluate the computational and statistical scalability of the method, and to leverage the naive Bayes assumption:

- 10k F(2D): 10 000 features constructed for each data set, each one is the sum of two randomly selected initial features,
- 100k F(2D): 100 000 features constructed (sums of two features),
- 10k F(3D): 10 000 features constructed (sums of three features).

The performance prediction guess is computed using a stratified tenfold cross-validation on the train data set.

8.3 Results

The challenge started Friday September 30, 2005, and ended Monday March 1, 2006. About 145 entrants participated to the challenge and submitted more than 4000 “development entries”. A total of 28 participants competed for the final ranking by providing valid challenge entries (results on train, validation, and test sets for all five tasks of the challenge). Each participant was allowed to submit at most 5 entries.

In the challenge, we rank 7^{th} out of 28, according to the average rank computed by the organizers. On 2 of the 5 five data sets (ada and sylva), our best entry ranks 1^{st}, as shown in Table 5. The AUC criterion, which evaluates the ranking of the class posterior probabilities, indicates high performance for our method, which ranks 3^{rd} on this criterion.

The detailed results of our entries are presented in Figure 11, together with all the final entries of the 28 finalists. The analysis of Guyon et al. (2006c) reveals that the top ranking entries exploit a variety of methods: ensembles of decision trees, support vector machines (SVM) kernel methods, Bayesian neural networks, ensembles of linear methods. On three out of the five data sets (gina, hiva and nova), the data set winner exploits a SVM kernel method. This type of the method is the most frequently used by the challenge participants, but their performance shows a lot of variance, so they need human expertise to adjust their parameters. On the contrary, ensembles of decision trees, like the method of the challenge winner, perform consistently well on all the data sets. Overall, the

Data Set	Our best entry				The challenge best entry			
	Test BER	BER Guess	Guess Error	Test Score	Test BER	BER Guess	Guess Error	Test Score
Ada	0.1723	0.1650	0.0073	0.1793	0.1723	0.1650	0.0073	0.1793
Gina	0.0733	0.0770	0.0037	0.0767	0.0288	0.0305	0.0017	0.0302
Hiva	0.3080	0.3170	0.0090	0.3146	0.2757	0.2692	0.0065	0.2797
Nova	0.0776	0.0860	0.0084	0.0858	0.0445	0.0436	0.0009	0.0448
Sylva	0.0061	0.0060	0.0001	0.0062	0.0061	0.0060	0.0001	0.0062
Overall	0.1307	0.1306	0.0096	0.1399	0.1090	0.1040	0.0079	0.1165

Table 5: Results of our best entry on the performance prediction challenge data sets.

top five ranked methods get an average test BER of 11%. Our method gets an average test BER of 13% and is ranked only 11^{th} on the BER criterion, even though it obtains very good results on the ada and sylva data sets.

The main limitation of our SNB(CMA) method comes from the naive Bayes assumption. Our method fails to correctly approximate the true class conditional distribution when the representation space of the data set does not contain any competitive subset of variables compliant with the naive Bayes assumption. For example, the gina data set consists of a set of image pixels, where the classification problem is to predict the parity of a number. On the initial representation of the gina data set, our test BER is only 13%, far from the best result which is about 3%. However, when the constructed features allow to “partially” circumvent the naive Bayes assumption, the method succeeds in significantly improving its performance, from 13% down to 7%. According to the challenge organizers, the hiva and nova data sets are also highly non-linear, which explains our poor BER results. For example, the nova data set is a text classification problem with approximately 17000 variables in a bag-of-words representation. In the case of this sparse data set, adding randomly constructed features is useless and results mainly in duplicating the variables. This explains why all our nova entries obtained the same BER results of 8%, far from the best result which is about 4%.

It is noteworthy that our method is very robust and ranks 4^{th} on average on the guess error criterion. Although we use all the train data to select our model without reserving validation data, our method is not prone to overfitting. In our feature construction schemes, we expand the size of the initial representation space of the data sets by a one hundred factor, which turns variable selection into a challenging problem. However, adding many variables never decreases the performance of our method. This means that our method correctly account for many useless and redundant variables, and is able to benefit from the potentially informative constructed variables, like in the gina data set for example.

Our method is evaluated with data sets having almost one billion values (up to 100 000 constructed features). Figure 12 reports the training time for all our submissions in the challenge. Our method is highly scalable and resistant to noisy or redundant features: it is able to quickly process about 100 000 constructed features without decreasing the predictive performance.

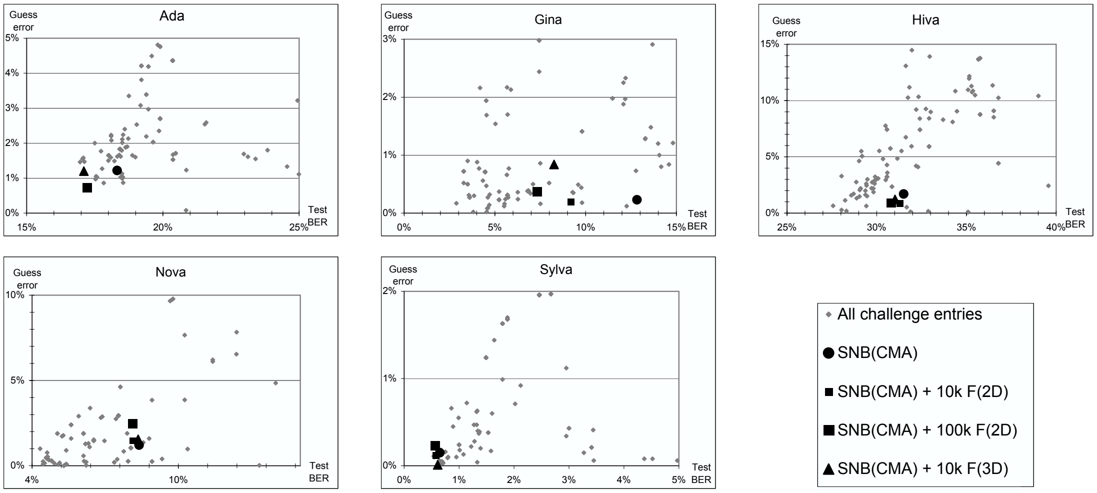

Figure 11: Detailed results of all of our entries on the performance prediction challenge data sets.

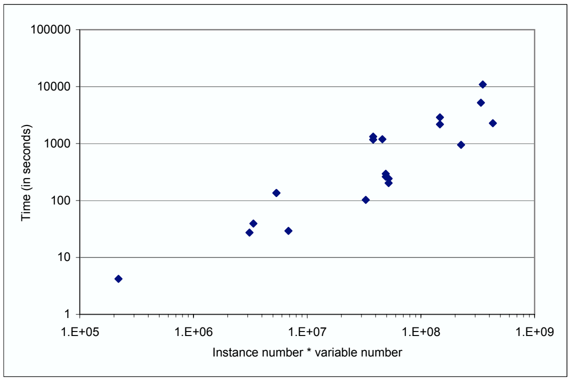

Figure 12: Training times for the SNB(CMA) classifier for all our entries in the performance prediction challenge.

9. Conclusion

The naive Bayes classifier is a popular method that is often highly effective on real data sets and is competitive with or even sometimes outperforms much more sophisticated classifiers. This paper confirms the potential benefit of variable selection to obtain still better performance.

We have proposed a MAP approach to select the best subset of variables compliant with the naive Bayes assumption and introduced an efficient search algorithm which time complexity is $O(KN\log(KN))$, where K is the number of variables and N the number of instances. We have also showed empirically that Bayesian model averaging is not much useful, since it does not perform significantly better that the MAP model.

On the basis of experimental and theoretical evidence that indicates that the posterior distribution of the models is exponentially peaked, we have shown that choosing a logarithmic smoothing of the posterior distribution makes sense. We have empirically demonstrated that the resulting compression-based model averaging scheme clearly outperforms the Bayesian model averaging scheme. This is encouraging and suggests that further research could be done to design a still more effective averaging scheme with more grounded foundations.

Our method consistently improves the performance of the naive Bayes classifier, but is outperformed by more sophisticated methods when the naive Bayes assumption is too harmful. In future work, we plan to exploit multivariate preprocessing methods in order to circumvent the naive Bayes assumption. On the basis of a set of univariate and multivariate conditional density estimators, our goal is to build a classifier that better approximates the true conditional density. In this setting, we think that compression-based model averaging might still be superior to Bayesian model averaging to account for the whole posterior distribution of the models.

Acknowledgments

I am grateful to the editor and the anonymous reviewers for their useful comments.

References

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1996. http://www.ics.uci.edu/mlearn/MLRepository.html.

M. Boullé. *Feature Extraction: Foundations And Applications*, chapter 25, pages 499–507. Springer, 2006a.

M. Boullé. Regularization and averaging of the selective naive Bayes classifier. In *International Joint Conference on Neural Networks*, pages 2989–2997, 2006b.

M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005a.

M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006c.

M. Boullé. A grouping method for categorical attributes having very large number of values. In P. Perner and A. Imiya, editors, *Proceedings of the Fourth International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *LNAI*, pages 228–242. Springer verlag, 2005b.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. California: Wadsworth International, 1984.

D. Dash and G.F. Cooper. Exact model averaging with naive Bayesian classifiers. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 91–98, 2002.

K. Deng, C. Bourke, S. Scott, and N.V. Vinodchandran. New algorithms for optimizing multi-class classifiers via ROC surfaces. In *Proceedings of the ICML 2006 Workshop on ROC Analysis in Machine Learning*, pages 17–24, 2006.

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

J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202. Morgan Kaufmann, San Francisco, CA, 1995.

T. Fawcett. ROC graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, 2003.

I. Guyon, S. Gunn, A. Ben Hur, and G. Dror. *Feature Extraction: Foundations And Applications*, chapter 9, pages 237–263. Springer, 2006a. Design and Analysis of the NIPS2003 Challenge.

I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction: Foundations And Applications*. Springer, 2006b.

I. Guyon, A.R. Saffari, G. Dror, and J.M. Bumann. Performance prediction challenge. In *International Joint Conference on Neural Networks*, pages 2958–2965, 2006c.

D.J. Hand and K. Yu. Idiot bayes ? not so stupid after all? *International Statistical Review*, 69(3): 385–399, 2001.

J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.

R. Kohavi and G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

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

P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

H. Liu, F. Hussain, C.L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 4(6):393–423, 2002.

T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

V. Pareto. *Manuale di Economia Politica*. Piccola Biblioteca Scientifica, Milan, 1906. Translated into English by Ann S. Schwier (1971), Manual of Political Economy, MacMillan, London.

F. Provost and P. Domingos. Well-trained pets: Improving probability estimation trees. Technical Report CeDER #IS-00-04, New York University, 2001.

F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–553, 1998.

A.E. Raftery and Y. Zheng. Long-run performance of Bayesian model averaging. Technical Report 433, Department of Statistics, University of Washington, 2003.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

C.P. Robert. *The Bayesian Choice: A Decision-Theoretic Motivation*. Springer-Verlag, New York, 1997.

C.E. Shannon. A mathematical theory of communication. Technical report, Bell systems technical journal, 1948.

I.H. Witten and E. Frank. *Data Mining*. Morgan Kaufmann, 2000.

Y. Yang and G. Webb. A comparative study of discretization methods for naive-Bayes classifiers. In *Proceedings of the Pacific Rim Knowledge Acquisition Workshop*, pages 159–173, 2002.