Regularization and Averaging of the Selective Naïve Bayes classifier

Marc Boullé

Abstract—**The Naïve Bayes classifier has proved to be very effective on many real data applications. Its performances usually benefit 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 new regularization technique to select the most probable subset of variables and propose a new model averaging method. The weighting scheme on the models reduces to a weighting scheme on the variables, and finally results in a Naïve Bayes with "soft variable selection". Extensive experimental results show that the averaged regularized classifier outperforms the initial Selective Naïve Bayes classifier.**

I. Introduction

THE Naïve Bayes modeling approach 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 [10, 18, 22]. Experiments demonstrate that even a simple Equal Width discretization with 10 bins brings superior performances compared to the assumption using a Gaussian distribution. Using the Bayes optimal MODL discretization method [4] to estimate the conditional probabilities has proved to be very efficient in detecting irrelevant variables [5]. Similar improvements can be achieved in the case of categorical variable, using the Bayes optimal value grouping method presented in [2] and extended in [3].

The naïve independence assumption can harm the performances when violated. In order to better deal with highly correlated variables, the Selective Naïve Bayes approach [17] uses a greedy forward search to select the variables. The accuracy is evaluated directly on the training set, and the variables are selected as long as they do not degrade the accuracy. One problem with this approach is that it does not prevent the selection of irrelevant variables having no effect on the accuracy, or even of redundant variables having either an insignificant or no effect on the accuracy. In [5], the area under the ROC curve [11] is used as a selection criterion and exhibits a better predictive performance than the accuracy criterion with fewer variables.

Although the Selective Naïve Bayes approach performs quite well on datasets with a reasonable number of variables, it does not scale on very large datasets 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 subject 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 new criterion is optimized owing to a search heuristic with super-linear algorithmic complexity in the number of instances and variables. We also present a new model averaging method, inspired from the Bayesian Model Averaging approach [15]. We show that averaging the model turns into averaging the contribution of the variables in the case of the Selective Bayes Classifier. Finally we proceed with extensive experiments to evaluate our method.

The remainder of the paper is organized as follows. Section II presents the regularization technique, section III the model averaging technique. Section IV browses related work. Section V proceeds with extensive experimental evaluations. Appendix summarizes the method and its results on the Performance Prediction Challenge [14].

II. Regularization

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

A. Introduction

The Naïve Bayes classifier is a very robust algorithm. It can hardly overfit the data, since no hypothesis space is explored during the learning process. The Selective Naïve Bayes classifier reduces the strong bias of the naïve 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 Naïve Bayes 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 (area under the ROC curve for example). These slight improvements during the training step can be detrimental to the predictive performance in test.

We propose to tackle this overfitting problem by relying on a Bayesian approach, where the best model is found by maximizing the probability $P(Model|Data)$ of the model

M. Boullé is with France Telecom R&D, 2, avenue Pierre Marzin, 22307 Lannion Cedex, France (e-mail: marc.boulle@lfrancetelecom.com).

given the data. Using the Bayes rule and since the probability $P(Data)$ is constant while varying the model, this is equivalent to maximizing $P(Model)P(Data/Model)$. In the following, we describe how we compute the likelihood of the models and propose a prior distribution for variable selection.

B. Assumptions and Notation

Let $X=(X_1, X_2,\ldots, X_K)$ be the vector of the K explanatory variables and Y the class variable. Let $y_1, y_2,\ldots, y_J$ be the J class values of Y.

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

Let $M=\{M_m\}$ be the set of all the potential Selective Naïve 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 $P(y_j)$ be the prior probabilities of the class values, and $P(X_k/y_j)$ the conditional probability distributions of the explanatory variables given the class values.

We assume that the prior probabilities $P(y_j)$ and the conditional probability distributions $P(X_k/y_j)$ are known. The purpose of the method is to select the best subset of variables for Naïve Bayes classification.

In the experimental section, the $P(y_j)$ are estimated by counting and the $P(X_k/y_j)$ are computed using the contingency tables, resulting from the preprocessing of the explanatory variables. This preprocessing is performed using the MODL discretization method [4] for the numeric variables and the MODL grouping method [3] for the categorical variables. The conditional probabilities are estimated using a m-estimate $(support+m*p)/(coverage+m)$ with $m=J/N$ and $p=1/J$, in order to avoid zero probabilities.

The MODL preprocessing methods are based on a Bayesian approach. A space of discretization (or grouping) models is defined, and a prior distribution on this model space is proposed. This leads to a Bayes optimal evaluation criterion of discretization (or grouping) models.

C. Likelihood of Models

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

$$P(y_j/X)=\frac{P(y_j)P(X/y_j)}{P(X)}. \quad (1)$$

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

$$P(y_j/X)=\frac{P(y_j)\prod_{k=1}^{K}P(X_k/y_j)}{P(X)}. \quad (2)$$

For a given model M_m, the class conditional probability estimation $P_m(y_j/X)$ turns into

$$P_m(y_j/X)=\frac{P(y_j)\prod_{k=1}^{K}a_{mk}P(X_k/y_j)}{P(X)}, \quad (3)$$

$$P_m(y_j/X)=\frac{P(y_j)\prod_{k=1}^{K}a_{mk}P(X_k/y_j)}{\sum_{j=1}^{J}P(y_j)\prod_{k=1}^{K}a_{mk}P(X_k/y_j)}. \quad (4)$$

Equation (4) 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 $P_m(y^{(n)}/X=x^{(n)})$. The likelihood of the model is obtained by computing the product of these quantities on the whole dataset. The negative log-likelihood of the model is given by:

$$-\log(P(D/M_m))=\sum_{n=1}^{N}-\log(P_m(y^{(n)}/X=x^{(n)})). \quad (5)$$

This quantity turns out to be the sum over the dataset of the Information Loss Function [21]. This quantity is a popular criterion for the evaluation of probabilistic prediction. It is minimized when the true probabilities are predicted.

D. 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 $C(K, K_m)$ 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 $C(K+K_m-1, K_m)$.

Taking the negative log of this prior, we get the following code length for the variable selection models

$$-\log(P(M_m))=\log(K+1)+\log(C(K+K_m-1, K_m)). \quad (6)$$

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

E. Posterior Distribution of the Models

The posterior probability of a model is evaluated as the product of the prior and the likelihood. This is equivalent to a MDL approach [20], where the code length of the model plus the data given the model has to be minimized:

$$\log(K+1)+\log\left(C\left(K+K_m-1,K_m\right)\right) - \sum_{n=1}^{N}\log\left(P_m\left(y^{(n)}/X=x^{(n)}\right)\right) \quad (7)$$

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, 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.

F. An Efficient Search Heuristic

Many heuristics have been used for variable selection. 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.

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. Each search step requires O(K) evaluations. The whole process converges very quickly, so that it still requires O(K) evaluations in practice. Each evaluation of a Selective Naïve Bayes model requires O(KN) steps, mainly to evaluate all the class conditional probabilities. Using the additivity of these probabilities with respect to the addition or deletion of variables, the total time complexity of the FFWBW heuristic can be reduced down to O(KN).

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 O($KN\log(KN)$).

Algorithm MS(FFWBW)

- Multi-start: repeat $\log_2(KN)$ times
  - Start with an empty subset of variables
  - Fast Forward Backward Selection
    - Initialize an empty subset of variables
    - Repeat until no improvement
      - Randomly reorder the variables
      - Fast Forward Selection
      - Randomly reorder the variables
      - Fast Backward Selection
  - Update the best subset of variables if improved
- Return the best subset of variables

III. Model Averaging

Model averaging consists in combining the prediction of an ensemble of classifiers in order to reduce the predictive error. This section introduces a new model averaging method applied to the Selective Naïve Bayes classifier.

A. From Bayesian Model Averaging to Expectation

Most inductive methods ignore the uncertainty in model selection and are over-confident about their predictive performances. The Bayesian Model Averaging (BMA) approach [15] provides a consistent method to accounting for model uncertainty, by weighting them by their posterior probability. For a given variable of interest Δ, this weighting scheme is computed according to the following formula:

$$P(\Delta/D)=\sum_m P(\Delta/M_m,D)P(M_m/D). \quad (8)$$

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

$$P(\Delta/D)=\frac{\sum_m P(\Delta/M_m,D)P(M_m)P(D/M_m)}{\sum_m P(M_m)P(D/M_m)}. \quad (9)$$

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

$$E(f)=\sum_m f(M_m,D)P(M_m/D). \quad (10)$$

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

B. Expectation of the class conditional information

The Naïve 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

$$f(M_m,D)=\frac{P(Y)\prod_{k=1}^{K}a_{mk}P(X_k/Y)}{P(X)}. \quad (11)$$

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 (11), we obtain

$$I(M_m, D) = I(Y) - I(X) + \sum_{k=1}^{K} a_{mk} I(X_k/Y) . \quad (12)$$

We are looking for the expectation of this conditional information

$$E(I) = \frac{\sum_m I(M_m, D) P(M_m/D)}{\sum_m P(M_m/D)}, \quad (13)$$

$$E(I) = I(Y) - I(X) + \frac{\sum_m P(M_m/D) \sum_{k=1}^{K} a_{mk} I(X_k/Y)}{\sum_m P(M_m/D)} , \quad (14)$$

$$E(I) = I(Y) - I(X) + \sum_{k=1}^{K} I(X_k/Y) \frac{\sum_m a_{mk} P(M_m/D)}{\sum_m P(M_m/D)} . \quad (15)$$

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 (7), on the basis of the prior probabilities and of the likelihood of the models. Using these coefficients, the expectation of the conditional information is

$$E(I) = I(Y) - I(X) + \sum_{k=1}^{K} b_k I(X_k/Y) . \quad (16)$$

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

$$P(Y/X) = \frac{P(Y) \prod_{k=1}^{K} P(X_k/Y)^{b_k}}{P(X)} . \quad (17)$$

It is noteworthy that the expectation of the conditional information in (16) is similar to the conditional information estimated by each individual model in (12). 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 is applied, each variable has a [0, 1] weight, which can be interpreted as a "soft selection".

C. Expectation with Compression Coefficients

Using the posterior probabilities to weight the models in the averaging approach presents some practical disadvantages. 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 tenths 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.

We propose an alternative weighting scheme, whose objective is to better account for the set of all models. Let us first introduce the compression coefficient $c(M_m,D)$ of a model. The Bayesian model selection approach we use to derive the criterion (7) is equivalent to a MDL model selection approach where

$$l(M_m) + l(D/M_m) = -\log(P(M_m)) - \log(P(D/M_m)) . \quad (18)$$

Let $M_\varnothing$ 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. The code length of the null model can be interpreted as the quantity of information necessary to describe the classes, when no explanatory data is used to induce the model. Applying (4), we have

$$l(M_\varnothing) + l(D/M_\varnothing) = \log(K+1) + N\, ent(Y) \quad (19)$$

where $ent(Y) = -\sum_{j=1}^{J} P(y_j) \log(P(y_j))$.

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:

$$c(M_m, D) = 1 - \frac{l(M_m) + l(D/M_m)}{l(M_\varnothing) + l(D/M_\varnothing)} . \quad (20)$$

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 our first model averaging method. Dividing the posterior probabilities by those of the null model, we get

$$b_k = \frac{\sum_m a_{mk} \frac{P(M_m/D)}{P(M_\varnothing/D)}}{\sum_m \frac{P(M_m/D)}{P(M_\varnothing/D)}} . \quad (21)$$

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

$$c_k = \frac{\sum_m a_{mk} c(M_m, D)}{\sum_m c(M_m, D)} . \quad (22)$$

In the implementation, we ignore the "bad" models and consider the positive compression coefficients only.

Mainly, the principle of this new heuristic weighting

scheme consists in smoothing the peaked posterior probability distribution with the log function.

D. An Efficient Algorithm for Model Averaging

We have previously introduced two model averaging methods which rely on the expectation of the class conditional information (with standard probabilistic weights or with compression-based weights). 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. This sampling strategy is biased in favor of the most probable models. This has little impact, since the most probable models contribute the most to the weights.

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 Naïve Bayes estimation of the class conditional probabilities is just extended with variables weights.

IV. Related Work

The section briefly reviews three popular alternative ensemble methods: Boosting, Bagging and Bayesian Model Averaging.

A. Introduction

The predictive performances of the classifiers can be understood using the bias-variance decomposition [16]. The bias evaluates the fit capacity: classifiers with low bias such as neural networks can approximate any data distribution, whereas classifiers with strong bias such as the Naïve Bayes classifier cannot fit complex data. The variance results from the statistical uncertainty around the training set.

The model averaging methods aim at reducing the bias or the variance part of predictive error by combining the predictions of an ensemble of classifiers. They mainly differ by the way they sample the distribution of the models and by their weighting scheme.

B. Boosting

The Boosting averaging method [12] exploits a weight distribution of the instances. Initially, all the instances have the same weight, and a classifier is trained on the initial training set. The instances that are incorrectly classified are given a higher weight, and a new training set is sampled from the weight-updated dataset. This process is repeated a given number of iterations, and the averaged classifier is built using weights based on the predictive performances of the individual classifiers. The Boosting method is theoretically founded to reduce the training bias, but without guarantee against overfitting.

C. Bagging

The Bagging (Boostrap Aggregating) averaging method [7] exploits a set of data samples obtained owing to a bootstrap process (N instances are randomly selected from the training set, with replacement). The classifier is trained on each bootstrap dataset, and the averaged classifier gives the same weight to all the trained models. The bootstrap process is a way of sampling the model space around reasonably good and equiprobable models. It is theoretically founded to reduce the variance part of the predictive error.

The random selection of data samples has been extended to variables in the Random Forests classifier [8]: each node of classification tree is trained on a new randomly sampled subset of variables. A similar approach is used in [23], where a sequence of Naïve Bayes classifiers is iteratively trained on randomly selected variables. The probability of selecting each variable is increased or decreased according to the performance of the previously trained classifier. By default, the number of training iterations is equal to the number of variables, and each classifier has the same weight.

D. Bayesian Model Averaging

The Bayesian Model Averaging (BMA) method [15] 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 to 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 performances [19].

The BMA approach has been applied to the Naïve Bayes classifier in [9]. 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 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.

V. EXPERIMENTS

This section presents an experimental evaluation of the performances of the Selective Naïve Bayes inducing methods described in the previous sections. We first comment on the rationale behind the compression-based model averaging approach before proceeding with extensive experiments.

A. The Waveform example

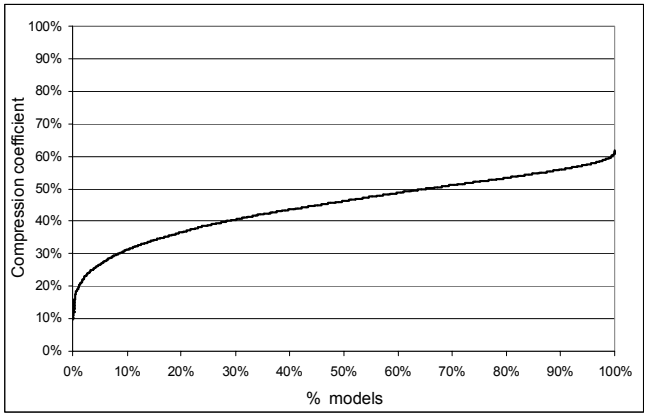

Fig. 1. Repartition function of the compression coefficients (normalized posterior probabilities) of half a million variable selection models evaluated for the Waveform dataset, sorted by increasing compression coefficient. For example, the 10% models on the left represent the models having the lowest compression coefficient.

The Waveform dataset [6] contains 5000 instances and 21 numeric variables, with 3 classes of waves. We use 70% of this dataset to train a Selective Naïve Bayes classifier, using the Bayes regularization introduced in section II. The MODL preprocessing determines that 2 variables (1st and 21th) are irrelevant. The variable selection problem consists in finding the most probable subset of variables 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 (5, 6, 9, 10, 11, 12, 13, 17). The posterior distribution is very sharp everywhere, not only around the MAP. Variable 18 is first selected in the 3rd model, which is about 40 times less probable than the MAP model. Variable 4 is first selected in the 10th model, about 4000 times less probable than the MAP model. Figure 1 displays the repartition function of the posterior probabilities, using a normalized log scale (compression coefficients). 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 is the null one. This huge range of probabilities is projected on a [0%, 62%] range of compression coefficients.

A closer look at the posterior distribution shows that most of the good models (in the top 50%) contain around 10 variables. Figure 2 displays the selected variables in the top 200 models (0.05%). Five variables (5, 9, 10, 11, 12) among the 8 MAP variables are always selected, and the other models exploits 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.

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). The compression-based model averaging exploits a flattened distribution of the weights (see figure 1) and thus enables to account for a large number of models. This averaging approach is evaluated in the next section.

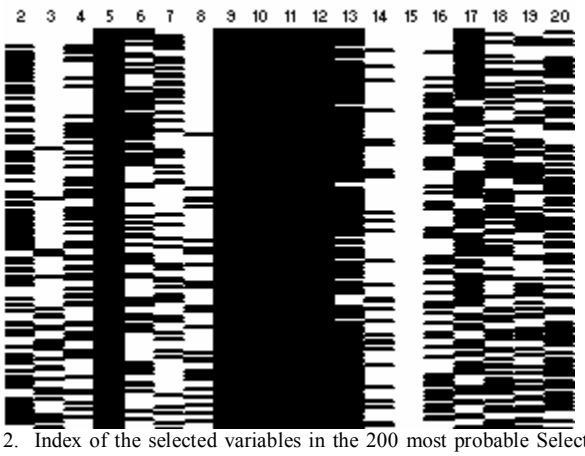

Fig. 2. Index of the selected variables in the 200 most probable Selective Naïve Bayes models for the Waveform dataset. Each line represents a model, where the variables are in black color when selected.

B. Experimental Setup

The experiments aim at comparing the performances of model averaging methods versus the MAP method, the standard Selective Naïve Bayes (SNB) and Naïve Bayes (NB) methods. All the classifiers except the last one exploit the same MODL preprocessing. The evaluated methods are:

- SNB(CMA): model averaging using compression-based weights in the expectation formula,
- SNB(MA): model averaging using expectation,
- SNB(MAP): MAP SNB model,
- SNB(AUC): optimization of the area under the ROC curve,
- SNB(ACC): optimization of the accuracy,
- NB: NB with MODL preprocessing,
- NB(EF): NB with 10 bins Equal Frequency discretization and no value grouping.

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 [9], similar to the SNB(MA) approach, was not evaluated since it is restricted to categorical attributes.

We evaluate three criteria of increasing complexity: accuracy (ACC), area under the ROC curve (AUC) and informational loss function (ILF). The ACC criterion focuses on the most probable class (which is fair to evaluate

class probabilities, but not very discriminating in case of highly unbalanced datasets), the AUC criterion focusses on the ranking of the class probabilities and the ILF criterion on the class probabilities. We normalize the ILF using the compression rate CR=1-ILF/Entropy (similar to (20), without the prior regularization penalty). The entropy has the same value as the ILF when the prior class probabilities are predicted on every instance. 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.

TABLE I
UCI DATASETS

Name	Instances	Numeric 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

We conduct the experiments on two collections of datasets: 30 datasets from the repository at University of California at Irvine [1] and 10 datasets from the NIPS 2003 Feature Selection Challenge [13] and the WCCI 2006 Performance Prediction Challenge [14]. A summary of some properties of these datasets is given in table I for the UCI datasets and in table II for the Challenge datasets. 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.

TABLE II
CHALLENGE DATASETS

Name	Instances	Numeric 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%

C. Results

We collect and average the three criteria owing to the stratified 10-fold cross validation, for the seven evaluated methods on the forty datasets. We summarize these results using the mean of each criterion, the number of significant wins or losses of the SNB(CMA) method and the average rank of each method. The results are presented in table III for the UCI datasets and in table IV for the Challenge datasets.

TABLE III
EVALUATION OF THE METHODS ON THE UCI DATASETS

Method	ACC			AUC			CR		
	M	W/L	R	M	W/L	R	M	W/L	R
SNB(CMA)	.824		2.2	.920		1.9	.577		2.2
SNB(MA)	.817	9/2	3.7	.916	11/0	3.3	.559	12/6	2.6
SNB(MAP)	.813	11/1	4.5	.913	17/1	4.4	.549	15/6	3.6
SNB(AUC)	.820	8/0	3.3	.918	10/2	3.1	.532	17/4	4.4
SNB(ACC)	.817	5/1	3.5	.910	14/0	4.5	.536	13/2	4.5
NB	.814	11/0	4.0	.913	16/1	4.6	.476	19/2	5.3
NB(EF)	.796	15/1	4.6	.911	13/3	4.8	.401	15/2	5.3

The evaluated criteria are the accuracy (ACC), the area under the ROC curve (AUC) and the compression rate (CR). The results are summarized across the datasets using the mean (M), the number of wins and losses for the SNB(CMA) method (W/L) and the average rank (R).

TABLE IV
EVALUATION OF THE METHODS ON THE CHALLENGE DATASETS

Method	ACC			AUC			CR		
	M	W/L	R	M	W/L	R	M	W/L	R
SNB(CMA)	.883		1.9	.904		1.0	.510		1.0
SNB(MA)	.872	3/0	3.5	.882	6/0	2.9	.446	9/0	2.3
SNB(MAP)	.865	4/0	4.5	.863	6/0	5.1	.425	9/0	3.7
SNB(AUC)	.872	6/0	3.6	.888	7/0	2.7	.331	10/0	4.1
SNB(ACC)	.875	3/2	2.9	.869	8/0	4.8	.365	9/0	4.3
NB	.841	7/0	4.9	.846	9/0	5.3	-.321	9/0	5.9
NB(EF)	.823	9/0	6.6	.833	9/0	6.2	-.423	10/0	6.7

The evaluated criteria are the accuracy (ACC), the area under the ROC curve (AUC) and the compression rate (CR). The results are summarized across the datasets using the mean (M), the number of wins and losses for the SNB(CMA) method (W/L) and the average rank (R).

The results of the two NB methods are reported mainly as a sanity check. The MODL preprocessing in the NB classifier exhibits better performances than the Equal Frequency discretization method in the NB(EF) classifier. All the SNB methods exploit the same MODL

preprocessing, allowing a fair comparison.

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 result of the NB methods, especially for the estimation of class conditional probabilities.

The three regularized methods SNB(MAP), SNB(MA) 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. 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.

The model averaging approach exploited in SNB(MA) 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) strongly 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 datasets. On the 10 challenge datasets, having very large numbers of variables, the SNB(CMA) methods always gets the best results on the AUC and CR criteria, and almost always on the accuracy criterion.

The domination of the SNB(CMA) 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 Naïve Bayes becomes effective for conditional probability estimation, whereas the standard Naïve Bayes is usually considered to be poor at estimating these probabilities.

VI. Conclusion

The Naïve Bayes classifier is a popular method that is often highly effective on real datasets 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 performances. We have proposed a new regularization method, founded on a Bayesian approach, to select the best subset of variables. We have introduced a new model averaging method as well, using compression-based weights. Extensive experiments on many datasets demonstrate the effectiveness of the new method, which considerably enhances the predictive performances of the raw Naïve Bayes classifier, especially for the estimation of the class conditional probabilities.

Appendix

This appendix summarizes the method and its results on the Performance Prediction Challenge [14].

Title: Regularized and Averaged Selective Naïve Bayes Classifier

Name, address, email: Marc Boullé,
France Telecom R&D, 2, avenue Pierre Marzin, 22307 Lannion cedex – France
marc.boulle@francetelecom.com

Acronym of my best entry: SNB(CMA) + 10k F(3D) tv

References:

M. Boullé, "Regularization and Averaging of the Selective Naïve Bayes classifier", *International Joint Conference on Neural Networks*, 2006.
M. Boullé, "MODL: a Bayes Optimal Discretization Method for Continuous Attributes", *Machine Learning*, to be published.

Method:

Our method is based on the Naive Bayes assumption.

All the input features are preprocessed using the Bayes optimal MODL discretization method.

We use a Bayesian approach to compromise between the number of selected features and the performance of the Selective Naïve Bayes classifier: this provides a regularized feature selection criterion. The feature selection search is performed using alternate forward selection and backward elimination searches on randomly ordered feature sets: this provides a fast search heuristic, with super-linear time complexity with respect to the number of instances and features.

Finally, our method introduces a variant of feature selection: feature "soft" selection. Whereas feature "hard" selection gives a "Boolean" weight to the features according to whether they selected or not, our method gives a continuous weight between 0 and 1 to each feature. This weighing schema of the features comes from a new classifier averaging method, derived from Bayesian Model Averaging.

The method computes the posterior probabilities of the classes, which is convenient when the classical accuracy criterion or the area under the ROC curve is evaluated. For the challenge, the Balanced Error Rate (BER) criterion is the main criterion. In order to improve the BER criterion, we adjusted the decision threshold in a post-optimization step. We still predict the class having the highest posterior probability, but we artificially adjust the class prior probabilities in order to optimize the BER criterion on the train dataset.

For the challenge, several trials of feature construction have been performed in order to evaluate the computational and statistical scalability of the method, and to naively attempt to escape the naïve Bayes assumption:

- 10k F(2D): 10 000 features constructed for each dataset, 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.

Results:

In the challenge, we rank 7th as a group and our best entry is 26th, according to the average rank computed by the organizers. On 2 of the 5 five datasets (ADA and SYLVA), our best entry ranks 1st, as shown in table V.

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.

Its main limitation comes from the Naïve Bayes assumption. However, when the constructed features allow to "partially" break the naïve Bayes assumption, the method succeeds in significantly improve its performances. This is the case for example for the GINA dataset, which does not fit well the naïve Bayes assumption: adding randomly constructed features allows to improve the BER from 12.83% down to 7.30%.

The AUC criterion, which evaluates the ranking of the class posterior probabilities, indicates high performances for our method.

Code: Our implementation was done in C++.

Keywords: Discretization, Bayesianism, Naïve Bayes, Wrapper, Regularization, Model Averaging

REFERENCES

[1] C.L Blake, C.J Merz, UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, UC. Irvine, Department of Information and Computer Science, 1998.

[2] M. Boullé, "A Bayes Optimal Approach for Partitioning the Values of Categorical Attributes". *Journal of Machine Learning Research 6*, 2005a, pp 1431-1452.

[3] M. Boullé, " A Grouping Method for Categorical Attributes Having Very Large Number of Values", P. Perner and A. Imiya (Eds.), *Machine Learning and Data Mining in Pattern Recognition*, Springer Verlag, lnai 3587, 2005b, pp 228-242.

[4] M. Boullé, "MODL: a Bayes Optimal Discretization Method for Continuous Attributes", *Machine Learning*, to be published.

[5] M. Boullé, "An Enhanced Selective Naïve Bayes Method with Optimal Discretization", *Feature extraction, foundations and Applications*, II(25), to be published.

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

[7] L. Breiman, "Bagging predictors", *Machine Learning*, 24(2), 1996, pp 123-140.

[8] L. Breiman, "Random forests", *Machine Learning*, 45(1), 2001, pp 5-32.

[9] D. Dash and G.F. Cooper, "Exact model averaging with naïve Bayesian classifiers", *Proceeding of the Nineteenth International Conference on Machine Learning*, 2002, pp 91-98.

[10] J. Dougherty, R. Kohavi, M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features". *Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann., 1995, pp 194-202.

[11] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers", Technical Report HPL-2003-4, HP Laboratories, 2003.

[12] Y. Freund and R.E. Shapire, "Experiments with a New Boosting Algorithm", *Proceeding of the Thirteenth International Conference on Machine Learning*, 1996, pp 148-156.

[13] I. Guyon, "Design of experiments of the NIPS 2003 variable selection benchmark", http://www.nipsfsc.ecs.soton.ac.uk/papers/Datasets.pdf, 2003.

[14] I. Guyon, "Model Selection Workshop and Performance Prediction Challenge", IEEE World Congress on Computational Intelligence, http://clopinet.com/isabelle/Projects/modelselect/#challenge, 2006.

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

[16] R. Kohavi and D.H. Wolpert, "Bias Plus Variance Decomposition for Zero-One Loss Function", *Proceeding of the Thirteenth International Conference on Machine Learning*, 1996, pp 275-283.

[17] P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers", *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1994, pp 399-406.

[18] H. Liu, F. Hussain, C.L. Tan, M. Dash., "Discretization: An Enabling Technique", *Data Mining and Knowledge Discovery*, 6(4), 2002, pp 393-423.

[19] A.E. Raftery and Y. Zheng "Long-Run Performance of Bayesian Model Averaging", Technical Report no. 433, Department of Statistics, University of Washington., 2003.

[20] J. Rissanen, "Modeling by shortest data description", *Automatica*, 14, 1978, pp 465-471.

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

[22] Y. Yang and G.I. Webb, "A comparative study of discretization methods for naïve-bayes classifiers", *Proceeding of the Pacific Rim Knowledge Acquisition Workshop*, 2002.

[23] Z. Zheng, "Naïve Bayesian Classifier Committees", *Proceedings of the ECML'98*, Berlin: Springer Verlag, 1998, pp 196-207.

TABLE V
RESULTS OF OUR BEST ENTRY ON THE PERFORMANCE PREDICTION CHALLENGE DATASETS

Dataset	Our best entry					The challenge best entry				
	Test AUC	Test BER	Ber Guess	Guess Error	Test Score	Test AUC	Test BER	Ber Guess	Guess Error	Test score
ADA	0.9149	0.1723	0.1650	0.0073	0.1793 (1)	0.9149	0.1723	0.1650	0.0073	0.1793
GINA	0.9772	0.0733	0.0770	0.0037	0.0767	0.9712	0.0288	0.0305	0.0017	0.0302
HIVA	0.7542	0.3080	0.3170	0.0090	0.3146	0.7671	0.2757	0.2692	0.0065	0.2797
NOVA	0.9736	0.0776	0.0860	0.0084	0.0858	0.9914	0.0445	0.0436	0.0009	0.0448
SYLVA	0.9991	0.0061	0.0060	0.0001	0.0062 (1)	0.9991	0.0061	0.0060	0.0001	0.0062
Overall	0.9242	0.1307	0.1306	0.0096	0.1399 (26.4)	0.8910	0.1090	0.1040	0.0079	0.1165 (6.2)