Forward stagewise naïve Bayes

Diego Vidaurre · Concha Bielza · Pedro Larrañaga

Abstract The naïve Bayes approach is a simple but often satisfactory method for supervised classification. In this paper, we focus on the naïve Bayes model and propose the application of regularization techniques to learn a naïve Bayes classifier. The main contribution of the paper is a stagewise version of the selective naïve Bayes, which can be considered a regularized version of the naïve Bayes model. We call it forward stagewise naïve Bayes. For comparison's sake, we also introduce an explicitly regularized formulation of the naïve Bayes model, where conditional independence (absence of arcs) is promoted via an L_1/L_2-group penalty on the parameters that define the conditional probability distributions. Although already published in the literature, this idea has only been applied for continuous predictors. We extend this formulation to discrete predictors and propose a modification that yields an adaptive penalization. We show that, whereas the L_1/L_2 group penalty formulation only discards irrelevant predictors, the forward stagewise naïve Bayes can discard both irrelevant and redundant predictors, which are known to be harmful for the naïve Bayes classifier. Both approaches, however, usually improve the classical naïve Bayes model's accuracy.

Keywords Naïve bayes · Forward stagewise naïve bayes · Regularized naïve bayes · Redundant predictors

D. Vidaurre · C. Bielza · P. Larrañaga (✉)
Computational Intelligence Group,
Departamento de Inteligencia Artificial,
Universidad Politécnica de Madrid, Madrid, Spain
e-mail: pedro.larranaga@fi.upm.es

D. Vidaurre
e-mail: diego.vidaurre@fi.upm.es

C. Bielza
e-mail: mcbielza@fi.upm.es

1 Introduction

Bayesian network classifiers [7] are a popular supervised classification paradigm. A well-known Bayesian network classifier is the naïve Bayes [14], a simple Bayesian network classifier that assumes that the predictors or variables are independent given each class value. Despite its simplicity and strong assumptions, the naïve Bayes classifier has been proven to work satisfactorily in many domains [4,11]. Typically, the parameters of the naïve Bayes model are found by maximizing the joint likelihood of the model.

The naïve Bayes model's accuracy, however, declines in the presence of noisy predictors. A noisy predictor can be a predictor that either carries no useful information for the classification (irrelevant) or is strongly dependent on another predictor (redundant). Redundancy is particularly harmful, because the predictor information has double the influence than it should.

For variable selection purposes, it is common to use filtering approaches, which perform variable selection disregarding the classifier, or (greedy) wrapper algorithms, which simultaneously introduce variables into the model and iteratively estimate the parameters. We focus on the wrapper paradigm. The (stepwise) selective naïve Bayes [13] is a popular example of greedy wrapper algorithm.

Regularization techniques introduce additional information, usually to solve an ill-posed problem or to avoid overfitting. Also, by imposing certain restrictions, regularization trades off a little bias against a larger reduction in variance. L_1-regularization [15], which imposes an L_1-penalty on the parameters, is also useful for variable selection, because it drives some parameters to exactly zero.

An example of regularization within the naïve Bayes model is the L_1/L_2-regularized naïve Bayes, taken by van Gerven and Heskes [9], which applies optimization

techniques to minimize the negative log-likelihood function of the data given the model plus an L_1/L_2-group penalty on the model complexity. This penalty encourages some predictors to be discarded. While they apply this idea only to the continuous predictor case, we extend it to deal with discrete predictors. Also, we introduce an adaptive penalty [19] that further improves the method's performance.

The main contribution of this paper, however, is a stagewise version of the selective naïve Bayes that is particularly useful when there are predictors that are relevant but, to some extent, redundant. At each iteration, instead of adding an "entire" predictor to the model, the parameters of the selected predictor are updated just a little. This method is inspired by the forward stagewise selection method for linear regression [17], which is also related to boosting and can be considered a form of regularization. We call this method forward stagewise naïve Bayes.

The remainder of the paper is organized as follows. Section 2 defines the notation and presents the basic naïve Bayes approach. Section 3 introduces some methods related to naïve Bayes, including selective naïve Bayes and the L_1/L_2-regularized naïve Bayes. Section 4 describes the effect of noisy (irrelevant or redundant) predictors. Section 5 introduces the forward stagewise naïve Bayes method. Section 6 discusses model selection. Section 7 outlines the set of experiments used to test the algorithms. Finally, Sect. 8 presents the conclusions and future work.

2 Notation and classical naïve Bayes

Let $\{X_1, \ldots, X_p\}$ be the set of p predictors and Y the class variable. Let $\boldsymbol{D} = \{(x_{r1}, \ldots, x_{rp}, y_r), r = 1, \ldots, N\}$ be the labeled data set containing N instances. $\boldsymbol{X}$ denotes the $N \times p$ predictor data matrix, with elements $x_{ri}, r \in \{1, \ldots, N\}, i \in \{1, \ldots, p\}$, and $\boldsymbol{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ denotes the vector of responses. We assume that the class variable, Y, may take values $j \in \{1, \ldots, J\}$. The objective is to learn a classifier from $\boldsymbol{D}$ so as to predict the class value for incoming data points just given by predictor values.

We assume that predictors are either discrete or continuous, although generalizations for combining the two are extremely straightforward.

When the inputs are discrete, we assume that each predictor X_i has M_i possible states. Assuming that the predictors are conditionally independent given the class variable, we denote their conditional probability table (CPT) as an $M_i \times J$ matrix $\boldsymbol{\Theta}_i$. Each element θ_{ikj} of $\boldsymbol{\Theta}_i, j \in \{1, \ldots, J\}, k \in \{1, \ldots, M_i\}$, is the probability of the predictor X_i taking its kth state given the jth class variable state, i.e., $\theta_{ikj} = P(X_i = k | Y = j; \boldsymbol{\Theta}_i)$.

We assume that, when the inputs are continuous, predictors follow a Gaussian distribution within each class value. We denote as $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ the vectors whose elements are, for each state of Y, the expectation and standard deviation of X_i, respectively, i.e., $X_i | Y = j \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}), j \in \{1, \ldots, J\}$. We denote the conditional density function for predictor X_i, given that $Y = j$, as $f(x_i | j; \mu_{ij}, \sigma_{ij})$.

Let $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_p\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_p\}$ and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_p\}$. Likewise, we denote the whole set of predictor parameters as $\boldsymbol{\Omega} = \{\boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_p\}$, where $\boldsymbol{\Omega}_i$ generically denotes either $\boldsymbol{\Theta}_i$ or $\{\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i\}$. Also, we denote class prior probabilities as $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$. Considering the predictors to be conditionally independent given the class, the full likelihood function for the naïve Bayes (NB) [14] model is defined as

$$L(\boldsymbol{D}; \boldsymbol{\Omega}) = \prod_{r=1}^{N} \left[\pi_{y_r} \prod_{i=1}^{p} \psi(X_i = x_{ri} | Y = y_r, \boldsymbol{\Omega}_i) \right], \quad (1)$$

where function $\psi(\cdot)$ computes the contribution of each predictor to the full likelihood. The likelihood is thus decomposable and can be computed separately for each predictor. We now define the contribution of each predictor to the full likelihood.

Let $\boldsymbol{W}(i)$ be an $N \times M_i$ indicator matrix for discrete predictor X_i. For the rth instance, the elements of the indicator matrix are defined as $w(i)_{rk} = 1$ if $x_{ri} = k$ and $w(i)_{rk} = 0$ if $x_{ri} \neq k$. Similarly, $\boldsymbol{S}$ is defined as the $N \times J$ indicator matrix for class variable Y. Hence, the contribution of a discrete predictor X_i and instance r to the full likelihood is

$$\begin{aligned} \psi(X_i = x_{ri} | Y = y_r, \boldsymbol{\Omega}_i) &= P(X_i = x_{ri} | Y = y_r, \boldsymbol{\Theta}_i) \\ &= \boldsymbol{w}(i)_{r\cdot} \boldsymbol{\Theta}_i \boldsymbol{s}_{r\cdot}^{\mathrm{T}}, \end{aligned} \quad (2)$$

where $\boldsymbol{w}(i)_{r\cdot}$ is the rth row vector of $\boldsymbol{W}(i)$ and $\boldsymbol{s}_{r\cdot}$ is the rth row vector of $\boldsymbol{S}$. Hence, $\boldsymbol{w}(i)_{r\cdot}$ and $\boldsymbol{s}_{r\cdot}$ are selecting the appropriate conditional probability for the rth instance from $\boldsymbol{\Theta}_i$.

On the other hand, the contribution of a continuous predictor X_i and instance r to the full likelihood is defined as

$$\begin{aligned} &\psi(X_i = x_{ri} | Y = y_r, \boldsymbol{\Omega}_i) = f(x_{ri} | y_r; \mu_{iy_r}, \sigma_{iy_r}) \\ &= \frac{1}{\sqrt{2\pi}\sigma_{iy_r}} \exp - \frac{(x_{ri} - \mu_{iy_r})^2}{2\sigma_{iy_r}^2}. \end{aligned} \quad (3)$$

Let $\boldsymbol{\Omega}_i^{(0)}$ be the parameters of predictor X_i such that they are exactly equal for all class values, that is, either $\theta_{ikj}^{(0)}$ in the discrete case or $\{\mu_{ij}^{(0)}, \sigma_{ij}^{(0)}\}$ in the continuous case are equal for all $j \in \{1, \ldots, J\}$. This is equivalent to removing predictor X_i from the model.

To estimate a NB model, we compute the maximum likelihood estimation (MLE) of the parameters, denoted as $\hat{\boldsymbol{\Theta}}_i^{(1)}$, $\hat{\boldsymbol{\mu}}_i^{(1)}$, $\hat{\boldsymbol{\sigma}}_i^{(1)}$ and $\hat{\boldsymbol{\pi}}$, as

$$\hat{\theta}_{ikj}^{(1)} = \frac{N_{ijk}}{N_j},$$
$$\hat{\mu}_{ij}^{(1)} = \frac{\sum_{r;y_r=j} x_{ri}}{N_j},$$
$$\hat{\sigma}_{ij}^{(1)} = \sqrt{\frac{\sum_{r;y_r=j}(x_{ri}-\hat{\mu}_{ij})^2}{N_j}},$$
$$\hat{\pi}_j = \frac{N_j}{N},$$

where N_{ijk} is the number of instances in the training data set, where predictor X_i takes the value k and Y takes the value j, and N_j is the number of instances where Y takes the value j.

The NB formulation for the probability of the class given the (continuous or discrete) predictors is

$$P(Y=j|X_1=k_1,\ldots,X_p=k_p,\hat{\boldsymbol{\Omega}}^{(1)},\hat{\boldsymbol{\pi}}) \propto \hat{\pi}_j \prod_{i=1}^{p} \psi(X_i=k_i|Y=j,\hat{\boldsymbol{\Omega}}_i^{(1)}) = \phi_j. \quad (4)$$

Thus, given vector $\boldsymbol{\phi}=(\phi_1,\ldots,\phi_J)$, whose components are computed with Eq. (4), the actual classification is performed by

$$\hat{j} = \text{maxpos}(\boldsymbol{\phi}), \quad (5)$$

where, maxpos$(\cdot)$ returns the position of the maximum element of the vector argument. Ties can be broken at random. Note that, although $\boldsymbol{\phi}$ depends on the input data configuration, it is omitted from the notation for simplicity sake.

3 Methods related to naïve Bayes

In this section, we introduce some existing methods related to NB: the selective naïve Bayes [13], the weighted naïve Bayes [6] and the L_1/L_2-regularized naïve Bayes [9]. Also, we generalize the L_1/L_2-regularized naïve Bayes to handle both discrete and continuous predictors and propose a simple improvement on this method.

3.1 Existing methods

The selective naïve Bayes (SNB) model [13] is a popular greedy, wrapper, stepwise algorithm for obtaining a NB model and performing variable selection. The SNB approach obeys Eq. (4) and, hence, makes use of the MLE. However, it is applied over only a subset of predictors. A forward greedy search finds this subset of predictors, where predictors are included in the model as long as the prediction accuracy (over training data) keeps increasing. Langley and Sage [13] also introduce a backwards search strategy, but they conclude that forward search is often more advantageous. On this ground, we use forward search in this paper.

The weighted naïve Bayes (WNB) model [6] includes all the predictors, which it weights according to their relevance for the classification. It is conceived only for discrete predictors. Weights are computed as

$$w_i = \sqrt{\sum_{j=1}^{J}\sum_{k=1}^{M_i}\Big[P(Y=j|X_i=k)-P(Y=j)\Big]^2}, \quad (6)$$

so that the resulting model is

$$P(Y=j|X_1=k_1,\ldots,X_p=k_p,\boldsymbol{\Omega}) \propto \hat{\pi}_j \prod_{i=1}^{p} \times \psi(X_i=k_i|Y=j,\boldsymbol{\Omega}_i)^{w_i} = \phi_j. \quad (7)$$

The classification rule is the same as for NB (Eq. (5)).

Using regularization techniques, the L_1/L_2-regularized naïve Bayes approach (L_1/L_2-NB) [9], designed for continuous predictors, is formulated as the optimization problem

$$\begin{aligned} &\text{argmin}_{\boldsymbol{\mu},\boldsymbol{\sigma}} \quad -\log L(\boldsymbol{D};\boldsymbol{\mu},\boldsymbol{\sigma}) \\ &+\lambda\sum_{i=1}^{p}\sqrt{\sum_{j=1}^{J}(\mu_{ij}-\hat{\mu}_{ij}^{(0)})^2+\sum_{j=1}^{J}(\sigma_{ij}-\hat{\sigma}_{ij}^{(0)})^2}, \\ &\text{s.t.} -\sigma_{ij} < 0 \qquad \forall i,j, \end{aligned} \quad (8)$$

where $L(\boldsymbol{D};\boldsymbol{\mu},\boldsymbol{\sigma})$ is defined in Eqs. (1) and (3) and λ is some regularization parameter. This optimization problem has Jp inequality constraints.

This way, the set of parameters of each single predictor (inside the square root) forms a group. This penalty is hence a group lasso-type penalty or L_1/L_2-penalty [18], which is able to discard entire groups. Therefore, all the parameters $\{\mu_{ij},\sigma_{ij}\}$ of some predictors will be prompted to be equal to $\{\hat{\mu}_{ij}^{(0)},\hat{\sigma}_{ij}^{(0)}\}$, so that such predictors will be effectively excluded.

Note that this optimization problem is convex. First, it is well known that the Gaussian likelihood defined in Eq. (3) is log-concave and hence the negative log-likelihood is convex [3]. This can be easily proved by taking the Hessian, which is positive semidefinite and thus proves convexity. Second, the L_1/L_2-penalty defined in (8) is just a sum of L_2-penalties. Since the L_2-norm function is convex, it is the sum of L_2-norms. The sum of two convex functions is convex. Finally, the inequality constraint functions are just nonnegativity constraints. Therefore, problem (8) is convex and is, in fact, denoted in the standard form. Although the entire objective function is non-smooth (non-differentiable), it is composed of a smooth loss function and a block-separable penalty and, hence, the problem can be solved by unconstrained (block) coordinate gradient descent optimization [16]. The constraint can be subsumed into the penalty term by setting it to ∞ when $\sigma_{ij} < 0$ for any pair (i,j).

3.2 Generalized L_1/L_2-regularized naïve Bayes

Now, we extend the L_1/L_2-NB formulation to deal with the discrete predictor case and propose an adaptive formulation of the problem for achieving better predictions.

We formulate the optimization problem for discrete predictors as

$$\begin{aligned}
&\text{argmin}_{\boldsymbol{\Theta}} -\log L(\boldsymbol{D};\boldsymbol{\Theta}) + \lambda \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{J}\sum_{k=1}^{M_i} (\theta_{ikj} - \hat{\theta}_{ijk}^{(0)})^2},\\
&\text{s.t.} \quad \mathbf{1}_i^t \boldsymbol{\theta}_{ij} - 1 = 0 \quad \forall i, j,\\
&-\theta_{ikj} < 0 \quad \forall i, j, k,\\
&\theta_{ikj} - 1 < 0 \quad \forall i, j, k, \qquad (9)
\end{aligned}$$

where, the loss function $L(\boldsymbol{D};\boldsymbol{\Theta})$ is defined in Eqs. (1) and (2), $\boldsymbol{\theta}_{ij}$ is the jth column of $\boldsymbol{\Theta}_i$ and $\mathbf{1}_i$ is a column vector with M_i ones. Therefore, there are J_p equality and $\sum_{i=1}^{p} JM_i$ inequality constraints (each pair of inequality constraints can be subsumed in one open box constraint $\theta_{ikj} \in (0,1)$).

This problem is also convex and is denoted in the standard form. Since the expression in Eq. (2) is linear on $\boldsymbol{\Theta}$, it is clear that the negative log-likelihood is convex and differentiable. The penalty in the loss function is also convex (but non-differentiable), and thus the entire loss function is convex. Both the equality and inequality constraint functions are affine. Even if we mixed both continuous and discrete predictors, the problem would still be convex. However, with the equality constraints, we cannot follow the gradient descent direction, so that (block) coordinate gradient descent optimization is not directly applicable. Instead, we take a simple approximation: starting with initial values $\Theta_i^{(0)}$, $i = 1, \ldots, p$, we update Θ_i toward $\Theta_i^{(1)}$ at each iteration, while the others predictors are held fixed, until the objective function in Eq. 9 reaches a minimum. This is a just a line-search.

A possible improvement on this approach is to use an adaptive penalty, which will hopefully improve the accuracy of the estimator. In L_1-penalized linear regression [15], for example, such penalties reduce the bias and lead to a consistent estimation [19]. The innovation is to penalize each predictor variable according to its importance. Each variable penalty is thus scaled by $1/|\beta_i^{(1)}|$, where $\beta_i^{(1)}$ is (in the $N > p$ case) the ordinary least squares regression coefficient or MLE. Note that $|\beta_i^{(1)}|$ is just the absolute upper bound of this coefficient in the regularized problem, i.e., the upper bound of the penalty for this variable.

We can apply an analogous idea to the L_1/L_2-NB formulation by computing weights $\boldsymbol{w} = (w_1, \ldots, w_p)$, for the discrete and continuous predictor cases, respectively, as

$$w_i = \sqrt{\sum_{j=1}^{J}\sum_{k=1}^{M_i} (\hat{\theta}_{ijk}^{(1)} - \hat{\theta}_{ijk}^{(0)})^2},$$

$$w_i = \sqrt{\sum_{j=1}^{J} (\hat{\mu_{ij}}^{(1)} - \hat{\mu_{ij}}^{(0)})^2 + \sum_{j=1}^{J} (\hat{\sigma_{ij}}^{(1)} - \hat{\sigma_{ij}}^{(0)})^2},$$

so that loss functions in (9) and (8) become, respectively,

$$\begin{aligned}
&-\log L(\boldsymbol{D};\boldsymbol{\mu},\boldsymbol{\sigma}) + \lambda \sum_{i=1}^{p} w_i \times \sqrt{\sum_{j=1}^{J}\sum_{k=1}^{M_i} (\theta_{ikj} - \hat{\theta}_{ijk}^{(0)})^2},\\
&-\log L(\boldsymbol{D};\boldsymbol{\Theta})\\
&+\lambda \sum_{i=1}^{p} w_i \sqrt{\sum_{j=1}^{J} (\mu_{ij} - \hat{\mu}_{ij}^{(0)})^2 + \sum_{j=1}^{J} (\sigma_{ij} - \hat{\sigma}_{ij}^{(0)})^2}.
\end{aligned}$$

Note that each w_i is a tight upper bound of the penalty for predictor X_i, and here we have the parallelism with the adaptive penalty for linear regression. We call this approach adaptive L_1-regularized naïve Bayes (aL_1/L_2-NB).

4 Noisy predictors

In this section, we define irrelevance and redundancy and remark on some ideas that motivate the approach introduced in Sect. 5.

We show that it is sometimes beneficial to use a point of compromise between $\hat{\boldsymbol{\Omega}}^{(0)}$ and $\hat{\boldsymbol{\Omega}}^{(1)}$ instead of the MLE like SNB does. Also, we discuss why the L_1/L_2-NB approach (including the adaptive version) can discard only irrelevant predictors and not redundant predictors.

First, we define the redundancy and irrelevance concepts and briefly discuss their effect on the NB model. We define a predictor as *noisy* if it is irrelevant for the class variable or is redundant to another predictor. Similar definitions of irrelevance and redundancy can be found, for example, in Kohavi and John [12] and Langley and Sage [13].

A discrete predictor X_i is *irrelevant* for Y if

$$\begin{aligned}
&P(Y = j | X_i = k) = P(Y = j), \quad \forall k \in \{1, \ldots, M_i\},\\
&\forall j \in \{1, \ldots, J\},
\end{aligned}$$

so that the value of X_i does not give any information about the value of Y. Equivalently, we can say that the within-class parameters of predictor X_i are equal for all class values. The definition for a continuous predictor is analogous.

On the other hand, two predictors X_{i_1} and X_{i_2} are *redundant* when there is a dependency between them.

Let $H(\cdot)$ represent the entropy function. Two predictors X_{i_1} and X_{i_2} are fully redundant when

$$H(X_{i_1} | X_{i_2}) = H(X_{i_2} | X_{i_1}) = 0. \qquad (10)$$

On the other hand, they are completely independent when

$$H(X_{i_1} | X_{i_2}) = H(X_{i_1}), \quad H(X_{i_2} | X_{i_1}) = H(X_{i_2}). \qquad (11)$$

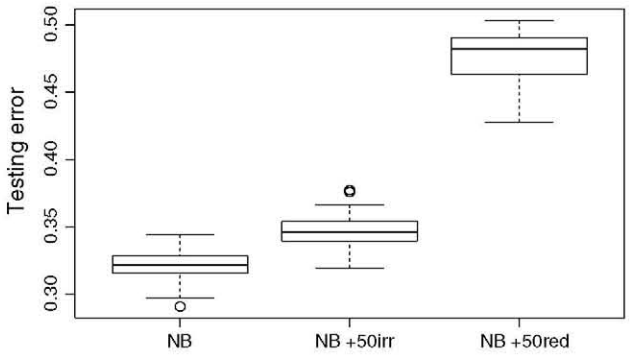

Fig. 1 Boxplots for the testing errors of NB without noisy variables (*left*), NB with 50 irrelevant predictors and NB with 50 redundant predictors

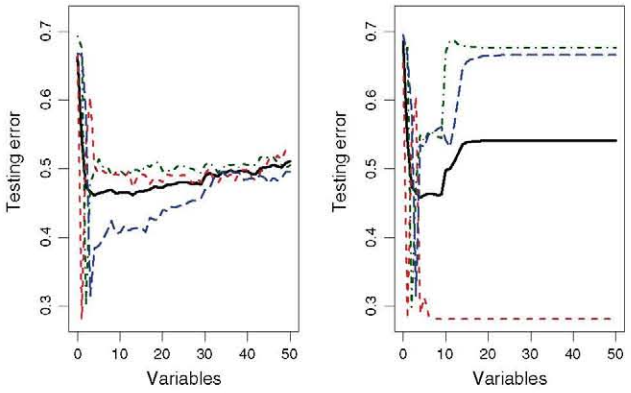

Fig. 2 Evolution of the testing error for an increasing number of irrelevant (*left*) and redundant (*right*) predictors. The first three predictors are non-noisy

Note that these conditions are just extremes of a continuum. In real-world data, predictors are rarely fully redundant or completely independent. Instead, they typically are somewhere between these two extreme conditions.

When $N \to \infty$ and p is finite (i.e., the complete information case), irrelevant variables do not increment the expected error of a NB classifier because $\hat{\boldsymbol{\Omega}}^{(1)} = \hat{\boldsymbol{\Omega}}^{(0)}$ holds exactly. In the realistic case, when N is finite, we only have $\hat{\boldsymbol{\Omega}}^{(1)} \simeq \hat{\boldsymbol{\Omega}}^{(0)}$. In the presence of many irrelevant predictors, these small differences accumulate and can finally bias the actual decision and degrade the classification accuracy.

It is well known, however, that, for NB, redundant predictors have a more harmful effect than irrelevant predictors [5]. Figure 1 shows testing errors of NB models obtained from three different types of discrete synthetic data sets. The first type has three non-noisy predictors, $\{X_1, X_2, X_3\}$, that are generated from the following probabilities

$$\boldsymbol{\Theta}_1 = \begin{pmatrix} 0.80 & 0.33 & 0.33 \\ 0.10 & 0.33 & 0.33 \\ 0.10 & 0.33 & 0.33 \end{pmatrix},$$

$$\boldsymbol{\Theta}_2 = \boldsymbol{\Theta}_3 = \begin{pmatrix} 0.33 & 0.30 & 0.30 \\ 0.33 & 0.10 & 0.60 \\ 0.33 & 0.60 & 0.10 \end{pmatrix}, \tag{12}$$

so that predictor X_1 discriminates between the first and the other two class values, and predictors X_2 and X_3 mainly discriminate between the second and third class values; $\boldsymbol{\pi}$ is defined as being equal for all three class values. The other two types have, in addition, 50 irrelevant discrete predictors and 50 (fully) redundant discrete predictors, respectively. The class can take three values, each with the same frequency. We have conducted 100 experiments, generating training data sets with $N = 1{,}000$ instances and test data sets with $N_{\text{te}} = 3{,}000$ instances. Notice that both kinds of noisy predictors, but especially the redundant ones, decrease accuracy.

Using the same data, Fig. 2 illustrates, for one experiment, the evolution of the testing error for an increasing number of predictors. The X-axis represents the number of predictors in the model. The first three added predictors (leftmost part of the graphs) are relevant, and the others, up to 50, are irrelevant (left hand graph) or redundant (right hand graph). Predictors are redundant with regard to the first non-noisy predictor. The solid line represents the error computed on the complete testing data set, whereas the other lines represent the error for each of the three class values. The black line represents the mean of the other three lines. We find that the class value that is best discriminated by the first predictor (short-dashed line) decreases the error in the presence of redundant predictors, but the other class values are no longer distinguishable. Irrelevant predictors, on the other hand, produce a more uniform and moderate increment of the error.

Ideally, SNB only adds predictors that reduce the classification error to the model. Hence, it will discard both redundant and irrelevant predictors, and retain those variables that are relevant but not redundant. However, as mentioned above, relevance and redundancy are not absolute concepts. What will SNB do with a set of relevant but non-fully redundant predictors? Let us suppose that there are two predictors, X_{i_1} and X_{i_2}, that are (non-fully) redundant, and each carries valuable information. In this paper, we claim that a NB model that balances the contribution of these predictors may be better than a classic NB model that either excludes or fully includes them, like SNB does.

We use an example to illustrate this point. Let us first define

$$\begin{aligned} \boldsymbol{\Theta}_i^{(\alpha_i)} &= \alpha_i \boldsymbol{\Theta}_i^{(1)} + (1-\alpha_i)\boldsymbol{\Theta}_i^{(0)}, \\ \boldsymbol{\mu}_i^{(\alpha_i)} &= \alpha_i \boldsymbol{\mu}_i^{(1)} + (1-\alpha_i)\boldsymbol{\mu}_i^{(0)}, \\ \boldsymbol{\sigma}_i^{(\alpha_i)} &= \alpha_i \boldsymbol{\sigma}_i^{(1)} + (1-\alpha_i)\boldsymbol{\sigma}_i^{(0)}, \end{aligned} \tag{13}$$

where, $\alpha_i \in [0, 1]$. Hence, $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i)}$ is a linear combination of $\hat{\boldsymbol{\Omega}}_i^{(0)}$ and $\hat{\boldsymbol{\Omega}}_i^{(1)}$, where α_i refers to predictor X_i. Within this notation, we can say that SNB only considers values

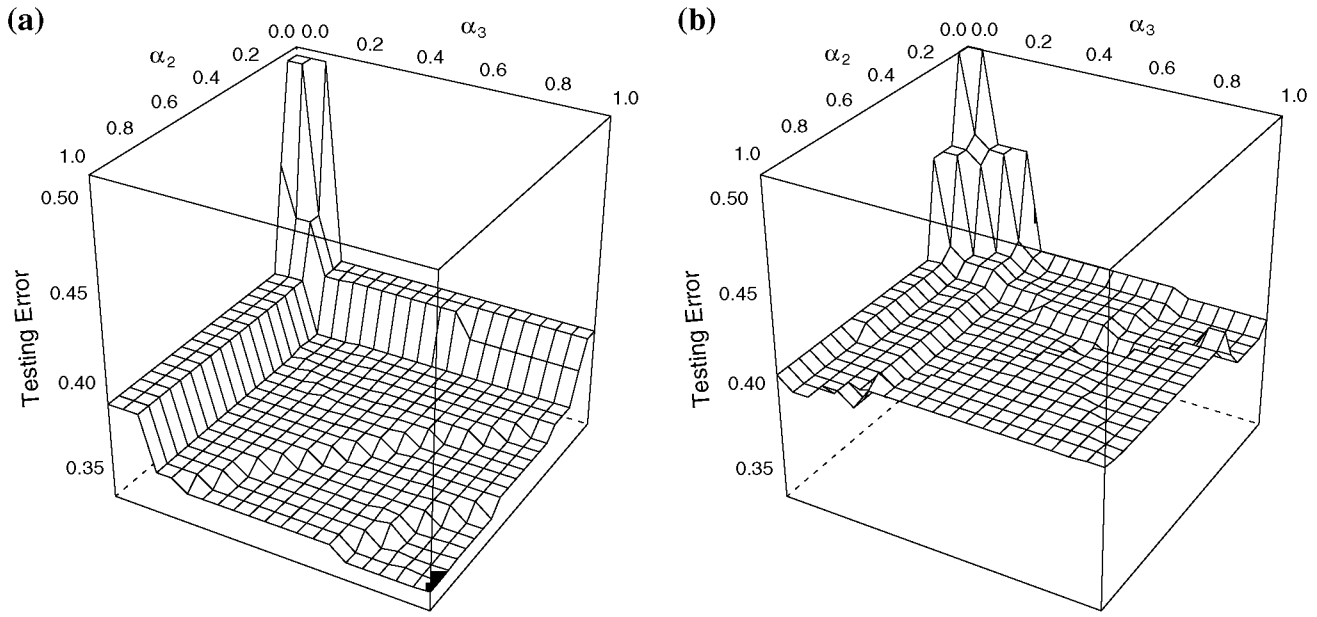

Fig. 3 **a** Testing error when discrete predictors X_2 and X_3 are not made redundant, **b** testing error when predictors are somewhat redundant

$\alpha_i \in \{0, 1\}$ (exclusion or inclusion, respectively, of predictor X_i).

Now, we consider a training data set with $N = 1{,}000$ instances and a testing data set with $N_{\text{te}} = 3{,}000$ instances, with three predictors whose CPTs are given in (12). Now, we consider making X_2 and X_3 redundant by setting $x_{r2} = x_{r3}$ for some proportion of the data instances.

Let us consider NB models with parameters $\hat{\boldsymbol{\Theta}}_1^{(1)}$, $\hat{\boldsymbol{\Theta}}_2^{(\alpha_2)}$ and $\hat{\boldsymbol{\Theta}}_3^{(\alpha_3)}$. For a grid of values $\alpha_2, \alpha_3 \in [0, 1]$, Fig. 3a shows testing errors when X_2 and X_3 are not made redundant, that is, if we have not set $x_{r2} = x_{r3}$ at any time.

Figure 3b shows testing errors when X_2 and X_3 are somewhat redundant, that is, after setting $x_{r2} = x_{r3}$ for some proportion of the data instances.

We find that, when X_2 and X_3 are independent, the minimum error is achieved when α_2, α_3 are equal to 1, i.e., when $\hat{\boldsymbol{\Theta}}_2 = \hat{\boldsymbol{\Theta}}_2^{(1)}$ and $\hat{\boldsymbol{\Theta}}_3 = \hat{\boldsymbol{\Theta}}_3^{(1)}$. On the other hand, when there is some dependence between X_2 and X_3, and X_1 is already part of the model, the best model is somewhere in $0 < \alpha_2, \alpha_3 < 1$.

Figure 4 illustrates the same scenario for continuous predictors. Figure 4a shows testing errors when X_2 and X_3 are independent, and Fig. 4b shows testing errors when X_2 and X_3 are somewhat redundant.

Although the effect is less obvious than in the discrete case, the conclusion is analogous.

These examples illustrate that it may be worthwhile finding a tradeoff between the MLE ($\hat{\boldsymbol{\Omega}}_i^{(1)}$) and the parameters that remove the predictor from the model ($\hat{\boldsymbol{\Omega}}_i^{(0)}$). This is the main motivation for proposing the forward stagewise naïve Bayes approach.

Finally, we note that, although L_1/L_2-NB is a natural choice for applying regularization to the NB model, it discards only irrelevant and not redundant predictors. It discards irrelevant predictors because, since $\hat{\boldsymbol{\Omega}}_i^{(0)}$ is not very different from $\hat{\boldsymbol{\Omega}}_i^{(1)}$ in this case, they make only a small contribution to the loss function in optimization problems (8) and (9). Note, however, that setting $\hat{\boldsymbol{\Omega}}_i = \hat{\boldsymbol{\Omega}}_i^{(0)}$ amounts to removing this predictor from the NB model, but it does not lead to the exclusion of the predictor from the loss function calculation in the optimization problem. In other words, according to this formulation, all predictors participate in the loss function (Eq. (1)), even when they can be simplified from the classification rule (Eqs. (4) and (5)). Therefore, if, for example, two predictors are fully redundant but separately relevant, the L_1/L_2-NB (or aL_1/L_2-NB) approach will add them both to the model, because, according to the log-likelihood formulation, both have a relevant impact on the loss function, no matter what the state of the other is. In other words, the inclusion of one predictor does not change the effect of the other on the loss function. In general terms, any algorithm that solves optimization problems (8) or (9) will select either both predictors or neither.

5 Forward stagewise naïve Bayes

We now introduce a more cautious version of the SNB approach, the forward stagewise naïve Bayes (fsNB). Like SNB, fsNB is a greedy algorithm but, instead of moving a set of parameters from $\hat{\boldsymbol{\Omega}}_i^{(0)}$ to $\hat{\boldsymbol{\Omega}}_i^{(1)}$ at each iteration, it takes small steps from $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i)}$ to $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i+\epsilon)}$, where $\epsilon > 0$ is some small

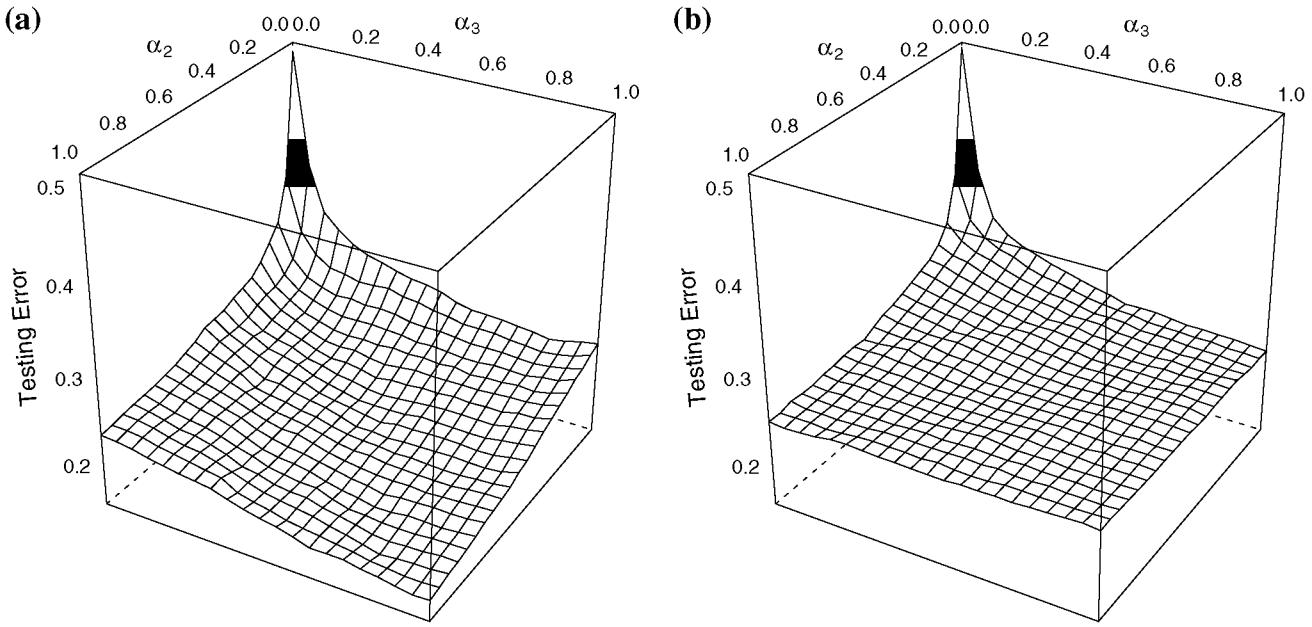

Fig. 4 **a** Testing error when continuous predictors X_2 and X_3 are not made redundant, **b** testing error when predictors are somewhat redundant

constant and α_i determines the current parameters of predictor X_i (see Eq. (13)). We can informally say that fsNB is to SNB what stagewise regression is to stepwise regression [17].

The rationale of this approach is to deal with the situation discussed in Sect. 4, i.e., when there are partially redundant variables that each carry separate information. By giving a balanced estimation of their parameters, we expect to retain the valuable information while minimizing the harmful effect of redundancy.

Concerning the greedy strategy, there is one important matter to address. At each iteration, we need to evaluate each predictor so as to decide which is going to be adjusted. There are two simple strategies for finding which predictor is most worth updating. Let us suppose that the parameters of predictor X_i are $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i)}$. The first strategy is to evaluate predictor X_i by checking $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i+\epsilon)}$. The second strategy is to check $\hat{\boldsymbol{\Omega}}_i^{(1)}$. Whatever we do, the predictor that leads to the greatest error decrement will be updated by ϵ (and the others are unchanged). Neither approach is problem-free. In the first case, it is often not possible to decide how important a predictor is by just looking at some small increment ϵ.

In the second case, if we look at the complete update of the parameters of the predictor, the contribution of other predictors with low $\alpha_{i'}$ ($i' \neq i$) could become negligible. Even when predictor X_i is important, the model accuracy may decrease considerably if the contribution of other important variables (almost) disappears.

Figure 5 illustrates this situation for two predictors, one relevant (left) and one irrelevant (right). It shows, at some early step of the algorithm, the evolution of the training and testing errors when we increase α_i for each predictor. Note that, in order to select the relevant rather than the irrelevant predictor, we have to look at a point between $\hat{\boldsymbol{\Omega}}^{(\alpha_i+\epsilon)}$ and $\hat{\boldsymbol{\Omega}}^{(1)}$, where the training (and testing error) is most decreased.

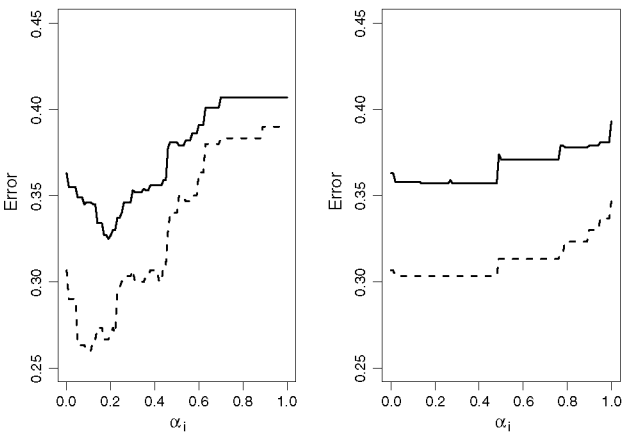

Fig. 5 Training error (*dashed line*) and testing error (*solid line*) across the evolution of two variables, one relevant (*left*) and one irrelevant (*right*)

To do this, we consider some further steps ν at each iteration, i.e., for each predictor, we check the error for $\hat{\boldsymbol{\Omega}}_i^{\alpha_i+\epsilon}$, $\hat{\boldsymbol{\Omega}}_i^{\alpha_i+2\epsilon}$, $\ldots$, $\hat{\boldsymbol{\Omega}}_i^{\alpha_i+t\epsilon}$, $\ldots$, $\hat{\boldsymbol{\Omega}}_i^{\alpha_i+\nu\epsilon}$. This way, at each iteration, we select the optimal values $\{i, t\}$, and update the parameters accordingly. Parameters ϵ and ν define how detailed is the search at each step and may have an impact in the computational efficiency of the algorithm. Reasonable variations of them, however, does not greatly change the algorithm accuracy.

Algorithm 1 Forward stagewise naïve Bayes (fsNB)

Initialize $\alpha_i = 0,\ \forall i \in \{1, \ldots, p\}$, so that $\hat{\boldsymbol{\Omega}}_i^{(\alpha_i)} = \hat{\boldsymbol{\Omega}}_i^{(0)}$
while $\alpha_i \neq 1, \forall i \in \{1, \ldots, p\}$, **do**
  $error^* = \infty$
  for $i \in \{1, \ldots, p\}$ such that $\alpha_i \neq 1$ **do**
    for $t \in \{1, \ldots v\}$ **do**
      Compute $\hat{\boldsymbol{\Omega}}_i^{+} = \hat{\boldsymbol{\Omega}}_i^{(\alpha_i + t\epsilon)}$
      $\phi_j^{(r)} = \pi_j \prod_{i'=1}^{p} \psi(X_{i'} = x_{ri'} | Y = j, \hat{\boldsymbol{\Omega}}_{i'}^{+})$, for $r \in \{1, \ldots, N\}$, $j \in \{1, \ldots, J\}$
      $error = 1/N \sum_{r=1}^{N} I\big(\text{maxpos}(\boldsymbol{\phi}^{(r)}), y_r\big)$
      if $error \leq error^*$ **then**
        $error^* = error$
        $i^* = i$
        $t^* = t$
      end if
    end for
  end for
  $\alpha_{i^*} = \alpha_{i^*} + t^* \epsilon$
end while

Algorithm 1 details the fsNB method in pseudocode format. The main part consists of two nested loops that look for the best pair $\{i, t\}$ at each iteration. Like SNB, the fitting criterion is the training error. The function $I(\cdot, \cdot)$ is an indicator function that outputs 1 if its arguments are equal and 0 if otherwise.

To minimize the computational cost, we can stop the procedure early if the training error has not improved during a certain number of iterations. We have observed that the minimum testing error is very rarely found after the training error comes to a standstill, which makes this strategy promising.

6 Model selection

Both the L_1/L_2-NB (using a grid of λ values) and the fsNB approaches generate a potentially large set of models, from which a final model needs to be selected. We can use a validation subset of the data set (if data are abundant), K-fold cross-validation, or some penalized criterion, which is typically the training loss plus some estimation of the optimism of the training loss rate. In this paper, we use the AIC statistic [1]:

$$\text{AIC} = Q(\boldsymbol{D}, \hat{\boldsymbol{\Omega}}) + \frac{2}{N} d,$$

where the loss function is the mean cross-entropy or deviance

$$Q(\boldsymbol{D}, \hat{\boldsymbol{\Omega}}) = \frac{1}{N} \sum_{r=1}^{N} -2 \log(Y = y_r | X_1 = x_{r1}, \ldots, X_p = x_{rp}, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\pi}}),$$

and d represents the degrees of freedom of the model, which we compute as

$$d = \sum_{i=1}^{p} I(\alpha_i > 0), \tag{14}$$

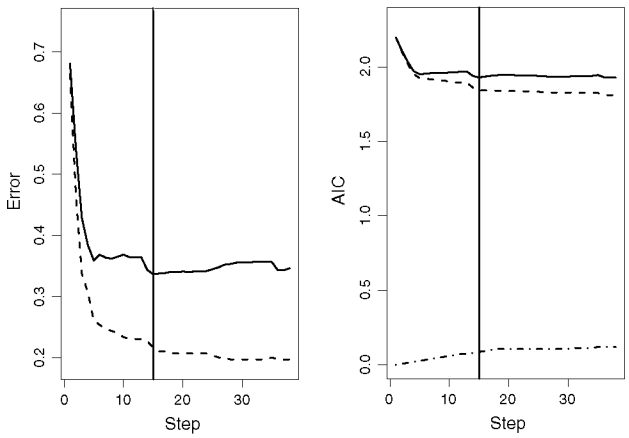

Fig. 6 *Left* Training error (*dashed line*) and testing error (*solid line*). Minimum testing error step is highlighted with a vertical line. *Right* AIC statistic (*solid line*), loss function (*dashed line*) and AIC penalty term (*dashed-dotted line*). The step with the lowest AIC statistic value is *highlighted* with a *vertical line*.

where $I(\cdot)$ outputs 1 if the argument is true and 0 if otherwise. Since a NB model is linear and $0 < d \leq p$, this is a reasonable estimation.

For L_1/L_2-NB, a possible natural choice, instead of Eq. (14), for computing d, in the discrete and continuous case, respectively, would be

$$d = \sum_{i=1}^{p} \frac{1}{J M_i} \sum_{j=1}^{J} \sum_{k=1}^{M_i} \frac{\theta_{ikj} - \hat{\theta}_{ijk}^{(0)}}{\hat{\theta}_{ijk}^{(1)} - \hat{\theta}_{ijk}^{(0)}},$$

$$d = \sum_{i=1}^{p} \frac{1}{J} \sum_{j=1}^{J} \left(\frac{\mu_{ij} - \hat{\mu}_{ij}^{(0)}}{\hat{\mu}_{ij}^{(1)} - \hat{\mu}_{ij}^{(0)}} + \frac{\sigma_{ij} - \hat{\sigma}_{ij}^{(0)}}{\hat{\sigma}_{ij}^{(1)} - \hat{\sigma}_{ij}^{(0)}} \right).$$

For fsNB, this would simplify to

$$d = \sum_{i=1}^{p} \alpha_i.$$

We have found, however, that the results are better using Eq. (14). Therefore, in this paper, we compute d using Eq. (14).

Figure 6 shows, for some generated data set with three relevant variables and twelve irrelevant variables, training and testing errors (left) and the AIC statistic, loss function and AIC penalty term (right) for a sequence of NB models generated by fsNB. Note that, in this example, the best model is nearly the same for the test data as for AIC.

7 Experiments

So far, we have presented some examples to illustrate the claims. In this section, we perform a more systematic

evaluation of the methods. We test the methods first on some synthetic data sets and then on some data sets derived from the *Diabetes* data set, taken from the UCI repository[1].

7.1 Synthetic data sets

We now run the algorithms on a number of synthetic training/test data sets, generated from several scenarios. Each data set has $p = 20$ predictors, which can be discrete ($M_i = 3$) or continuous. Training data sets have $N = 300$ instances and test data sets have $N_{\text{te}} = 3{,}000$ instances. There are $J = 3$ class values.

Within each data set, there are $p_1 = 3$ non-noisy predictors, $p_2 = 7$ non-fully redundant predictors, which nevertheless carry some information, and $p_3 = 10$ totally noisy predictors, which may be irrelevant or redundant to any of the p_1 non-noisy predictors. We call these three groups, respectively, V_1, V_2 and V_3. Hence, $p = p_1 + p_2 + p_3$.

For each experiment, we randomly generate the "true" parameters that produce the data as follows. In the discrete case, for each predictor in V_1, we sample

$$\boldsymbol{\theta}_{ij} \sim \text{Dir}(\boldsymbol{c}),$$

where $\boldsymbol{\theta}_{ij}$ is the jth column of $\boldsymbol{\Theta}_i$ and $\text{Dir}(\boldsymbol{c})$ is a Dirichlet distribution with the vector of shape parameters $\boldsymbol{c}$, whose components are all equal except one, which is different for each $j \in \{1, \ldots, J\}$.

Within each data set, all predictors in V_2 have the same CPT, which is similarly generated from a Dirichlet distribution. Each predictor in V_2 is slightly redundant to the preceding and following predictor, i.e., $H(X_i|X_{i-1}) < H(X_i)$ and $H(X_i|X_{i+1}) < H(X_i)$ for $i \in \{p_1 + 2, p_1 + p_2 - 1\}$ (assuming that predictors in V_2 are preceded by predictors in V_1 and followed by predictors in V_3). This redundancy is achieved by setting the value of the predictor X_i to be equal to either X_{i-1} or X_{i+1} with a probability equal to 0.5.

If predictors in V_3 are irrelevant, they have the parameters of a multinomial distribution, and they are generated from a Dirichlet distribution with equal hyperparameters. In other words, the CPT columns of each predictor in V_3 are all equal. If predictors in V_3 are redundant, the parameters are generated as for irrelevant predictors. In this case, however, each predictor has a very low conditional entropy given some randomly selected predictor from V_1. This is achieved by setting the value of the predictor in V_3 to be equal to the predictor in V_1 with a probability equal to 0.9. Note that, once the corresponding predictor in V_1 has been added, this predictor does not carry any additional useful information at all.

In the continuous case, predictors are generated from Gaussian distributions. For each predictor in V_1, we sample

$$\mu_{ij} \sim \text{Unif}(-2, 2), \quad \sigma_{ij} = 0.75,$$

where $\text{Unif}(-2, 2)$ is the uniform distribution between -2 and 2.

As in the discrete case, all predictors in V_2 have the same parameters. Again, let X_i be equal to either X_{i-1} or X_{i+1} with a probability equal to 0.5.

If predictors in V_3 are irrelevant, we have $\mu_{ij} = m_i$, for all $j \in \{1, \ldots, C\}$. The value m_i is generated from a uniform distribution in the interval $(-2, 2)$. If predictors in V_3 are redundant, parameters are generated similarly, but, for each data instance, each predictor in V_3 is bound, with a probability equal to 0.9, to have the same value as some predictor in V_1, plus some small noise.

Finally, we set $\boldsymbol{\pi} = (1/3, 1/3, 1/3)$ in all cases. Hence, we have four different scenarios, which are the four possible combinations of discrete/continuous predictors and irrelevant/redundant predictors within V_3.

We generate 100 different data sets from each scenario using the Bayes rule (taking into account the mentioned redundancies). Table 1 shows, for each data set type, the means and standard deviations of the testing misclassification error, number of selected variables and number of (fully) noisy-selected variables, for NB, SNB, WNB, aL_1/L_2-NB and fsNB. We have run fsNB with parameters $\epsilon = 0.025$, $\nu = 20$, which we have empirically observed to be a good choice in general. Also, we use early stopping (see Sect. 5). For comparison's sake, we have also run NB on a subset of predictors, selected by (prefiltering) correlation-based feature selection [10]. We denote this approach as CFS + NB.

We find that there are two clearly different scenarios. First, when the noisy predictors are irrelevant, the methods that do not select variables (NB and WNB) perform best. This is certainly expectable, because, as discussed in Sect. 4, NB is relatively robust to irrelevant predictors, and there are not enough to significantly reduce accuracy. Note, however, that, in the discrete case at least, fsNB is closer to NB and WNB than the other wrapper selective methods and also than CFS + NB.

Second, when the noisy predictors are redundant, fsNB beats the others. CFS + NB also works fine and turns out to be the most accurate method in the continuous case. The differences between fsNB and SNB are probably due to the fsNB's balanced estimation of parameters of the predictors in V_2. The number of selected predictors is not very different for fsNB and SNB in this case. CFS + NB clearly selects more predictors than the wrapper approaches. Finally, note that, excepting for the continuous with irrelevant noise variables data set, aL_1/L_2-NB does not excel. Although L_1/L_2-NB is not shown in Table 1, aL_1/L_2-NB is slightly better than its non-adaptive counterpart. Regarding computational cost, SNB

[1] http://archive.ics.uci.edu/ml.

Table 1 Mean testing misclassification error (top), mean number of selected variables (middle) and mean number of (fully) noisy selected variables (bottom) for each synthetic data set type and each method

Data set type	NB	SNB	WNB	aL_1/L_2-NB	fsNB	CFS + NB
Misclassification error						
DI	0.076 (±0.03)	0.080 (±0.03)	**0.075** (±0.03)	0.079 (±0.06)	0.077 (±0.02)	0.166 (±0.18)
CI	**0.082** (±0.04)	0.083 (±0.05)	**0.082** (±0.04)	0.083 (±0.05)	0.083 (±0.05)	0.087 (±0.02)
DR	0.152 (±0.07)	0.076 (±0.02)	0.131 (±0.08)	0.171 (±0.10)	**0.070** (±0.03)	0.082 (±0.06)
CR	0.132 (±0.05)	0.097 (±0.03)	0.132 (±0.05)	0.158 (±0.10)	0.090 (±0.03)	**0.083** (±0.08)
Number of selected variables						
DI	–	**6.2** (±1.4)	–	6.5 (±2.1)	6.3 (±1.7)	10.8 (±0.8)
CI	–	6.0 (±1.5)	–	6.0 (±1.2)	**5.3** (±1.8)	10.0 (±0.1)
DR	–	5.7 (±1.5)	–	10.2 (±2.1)	**5.6** (±1.7)	10.4 (±0.6)
CR	–	**5.5*** (±1.3)	–	11.8 (±1.9)	6.6 (±2.3)	10.8 (±1.0)
Number of noisy selected variables						
DI	–	0.6 (±0.8)	–	**0.2** (±0.7)	0.3 (±0.9)	0.8 (±0.9)
CI	–	0.8 (±0.8)	–	**0.1** (±0.3)	**0.1** (±0.2)	0.4 (±0.5)
DR	–	0.4 (±0.7)	–	5.1 (±0.8)	**0.2** (±0.5)	0.6 (±0.6)
CR	–	**0.8** (±0.4)	–	5.9 (±0.8)	1.1 (±0.7)	1.0 (±1.0)

Data set types are discrete with irrelevant noise variables (DI), continuous with irrelevant noise variables (CI), discrete with redundant noise variables (DR) and continuous with redundant noise variables (CR). The best result for each row is highlighted in bold. NB and WNB have been omitted from the variable selection report because they do not perform variable selection

Table 2 Mean 10-CV cross-validated misclassification error (top) and number of selected variables (bottom) for each data set derived from the *Diabetes* data set and each method

Data set	NB	SNB	WNB	aL_1/L_2-NB	fsNB	CFS + NB
Misclassification error						
$\boldsymbol{\pi} = (1/4, 3/4)$	0.28 (±0.06)	0.23 (±0.05)	0.28 (±0.06)	0.25 (±0.07)	**0.21** (±0.06)	0.26 (±0.10)
$\boldsymbol{\pi} = (1/2, 1/2)$	0.28 (±0.07)	0.27 (±0.06)	0.28 (±0.07)	0.27 (±0.08)	**0.26** (±0.07)	0.27 (±0.15)
$\boldsymbol{\pi} = (3/4, 1/4)$	0.20 (±0.04)	0.18 (±0.05)	0.20 (±0.04)	0.19 (±0.05)	**0.16** (±0.06)	0.18 (±0.15)
Number of selected predictors						
$\boldsymbol{\pi} = (1/4, 3/4)$	–	**1.7** (±0.48)	–	4.4 (±3.80)	2.2 (±0.42)	3.0 (±0.42)
$\boldsymbol{\pi} = (1/2, 1/2)$	–	**3.3** (±0.48)	–	8.3 (±0.48)	4.4 (±0.95)	4.2 (±0.51)
$\boldsymbol{\pi} = (3/4, 1/4)$	–	3.3 (±0.67)	–	8.4 (±0.84)	**2.4** (±0.51)	4.9 (±0.78)

The best result for each row is highlighted in bold. NB and WNB have been omitted from the variable selection report because they do not perform variable selection

and fsNB take, respectively, 125.10 and 6732.25 evaluations on average. The computational cost is similar for all data sets.

7.2 *Diabetes* data sets

We now carry out some experiments with real data. We use the *Diabetes* data set, which has $N = 442$ instances and $p = 10$ continuous predictors. Although the response is continuous, we generate data sets for binary classification by means of the rule

$$y_r = \begin{cases} 0 & \text{if } \tilde{y}_r < \tau, \\ 1 & \text{if } \tilde{y}_r \geq \tau. \end{cases}$$

where $\tilde{y}_r$ is the continuous response and τ is some real constant. We generate three different data sets by setting τ to be equal to the first three quartiles. Therefore, for each data set, we have, respectively, $\boldsymbol{\pi} = (1/4, 3/4)$, $\boldsymbol{\pi} = (1/2, 1/2)$ and $\boldsymbol{\pi} = (3/4, 1/4)$.

Table 2 illustrates the results obtained from 10-fold cross-validation, which include testing misclassification error and number of selected variables. As before, the tested methods are NB, SNB, WNB, aL_1/L_2-NB, fsNB and CFS + NB. We have run fsNB with parameters $\epsilon = 0.025$, $\nu = 20$, using early stopping.

Note that fsNB is the most accurate, followed by SNB and CFS + NB.

Table 3 Mean 10-CV cross-validated misclassification error (top) and number of selected variables (bottom) for each subject in the *Starplus* data set and each method

Subject	NB	SNB	WNB	aL_1/L_2-NB	fsNB	CFS + NB
Misclassification error						
04799	0.47 (±0.08)	0.45 (±0.04)	0.47 (±0.08)	0.52 (±0.07)	**0.41** (±0.26)	0.50 (±0.23)
05675	0.44 (±0.07)	**0.43** (±0.06)	0.44 (±0.07)	0.51 (±0.06)	0.50 (±0.11)	0.46 (±0.19)
04820	0.44 (±0.07)	0.43 (±0.06)	0.44 (±0.07)	0.55 (±0.03)	0.37 (±0.34)	**0.34** (±0.21)
05680	0.45 (±0.05)	0.44 (±0.06)	0.45 (±0.05)	0.57 (±0.04)	**0.35** (±0.26)	0.48 (±0.16)
04847	0.36 (±0.06)	**0.33** (±0.05)	0.36 (±0.06)	0.57 (±0.06)	0.35 (±0.06)	0.44 (±0.18)
05710	0.40 (±0.07)	0.45 (±0.06)	0.40 (±0.07)	0.55 (±0.02)	**0.36** (±0.26)	0.48 (±0.32)
Number of selected predictors						
04799	–	3.8 (±1.51)	–	**0.2** (±0.02)	1.0 (±0.77)	1.1 (±0.81)
05675	–	3.5 (±1.32)	–	**0.3** (±0.01)	0.9 (±0.30)	1.0 (±0.66)
04820	–	4.9 (±1.31)	–	**0.1** (±0.01)	1.3 (±1.04)	0.9 (±1.01)
05680	–	5.5 (±2.50)	–	**0.3** (±0.02)	1.2 (±0.79)	1.2 (±0.36)
04847	–	4.2 (±1.51)	–	**0.1** (±0.02)	1.8 (±0.03)	2.1 (±0.77)
05710	–	5.9 (±1.82)	–	**0.2** (±0.03)	1.5 (±0.91)	2.0 (±0.52)

The best result for each row is highlighted in bold. NB and WNB have been omitted from the variable selection report because they do not perform variable selection

Note that aL_1/L_2-NB is always worse than fsNB and SNB, which is a possible sign of certain redundancy among the predictors (that aL_1/L_2-NB is not purging). In these data sets, WNB obtains very similar results to NB. None of the methods, however, is very accurate when $\boldsymbol{\pi} = (1/4, 3/4)$. In this case, the simple "most frequent class" rule obtains an accuracy similar to NB (0.28), which is not greatly improved by any method. On the other hand, the number of selected predictors is reasonable for SNB, fsNB and CFS + NB, and higher for aL_1/L_2-NB. The L_1/L_2-NB approach (not shown) achieves similar results to aL_1/L_2-NB, for both accuracy and selected variables. The mean number of evaluations for SNB is 30.7, whereas fsNB needs 905.4 evaluations on average.

7.3 Neuroscience fMRI data

In this section, we report results on functional magnetic resonance imaging (fMRI) data, the *StarPlus* data set[2], collected at Carnegie Mellon University.

Experiments are conducted on six subjects and forty trials per subject. For each trial, the subject is shown a picture for 4 s and a sentence for 4 s. The objective is to discriminate between these two mental states: "picture" or "sentence". Each data item matches a unique 3-dimensional image. Images are captured every 0.5 s. Hence, each trial has 16 useful images. In brief, there are six data sets, one per subject, and they all have $n = 40 \times 16 = 640$ data items. On the other hand, each image has a number of voxels, split into 25 localized regions of interest (ROIs). In this paper, instead of considering each individual voxel, we will use the mean activation of voxels at each ROI. Therefore, our data set has $p = 25$ covariates.

Table 3 shows the results obtained from 10-fold cross-validation. Algorithms and parameter configuration are the same than in Sects. 7.1 and 7.2.

In this example, fsNB beats the other wrapper algorithms in four out of six subjects, whereas SNB is the best wrapper method for the other two subjects. CFS + NB performs better than fsNB and SNB in one of the subjects.

On the other hand, fsNB selects fewer predictors than SNB in all cases. The number of selected predictors is not very different from CFS + NB.

The performance of aL_1/L_2-NB is poor, and the model selection procedure often prefers the model with no predictors. Note that, in general, none of the approaches behave particularly well. We conjecture that this is because the data have a very nonlinear nature.

The mean number of evaluations for SNB is 126.2, whereas fsNB needs 9,311.1 evaluations on average.

7.4 Comparison across data sets

Finally, we perform an overall analysis of the methods that includes the results obtained from all the data sets described above. To do so, we follow the guidelines outlined by [8], performing all pairwise comparisons among the classifiers to detect (statistically) significant differences between each pair. In particular, we use the [2] dynamic procedure to adjust the raw p values. Table 4 shows, for each pair, these adjusted

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/.

Table 4 Adjusted p values, via the Bergmann–Hommel's dynamic procedure, for each pair of methods

Pair	Adjusted p value	Pair	Adjusted p value
fsNB versus NB	1.57E−6	SNB versus NB	0.783
fsNB versus WNB	3.35E−6	SNB versus WNB	0.783
fsNB versus aL_1/L_2-NB	3.35E−6	SNB versus aL_1/L_2-NB	0.783
fsNB versus SNB	9.86E−4	CFS + NB versus SNB	2.210
fsNB versus CFS + NB	0.030	aL_1/L_2-NB versus NB	2.424
CFS + NB versus NB	0.352	WNB versus NB	2.424
CFS + NB versus WNB	0.352	aL_1/L_2-NB versus NB	2.424
CFS + NB versus aL_1/L_2-NB	0.352		

p values. We can observe that fsNB is significantly better than all the other procedures, with a significance level of 0.05.

8 Discussion

In this paper, we have proposed a forward stagewise version of the forward stepwise SNB approach. This approach has some advantages over the usual SNB, and often beats other naïve Bayes-based algorithms, like the WNB. We have illustrated this point empirically on both synthetic and real data sets. The forward stagewise approach is computationally more expensive than SNB. Computational complexity, however, can be modulated via the ν parameter, which, with ϵ, defines the extent of the search at each step.

We have also extended the L_1/L_2-regularized naïve Bayes approach taken by van Gerven and Heskes [9] to accommodate discrete predictors. In addition, we have introduced a handy modification of this method based on adaptive penalties [19]. Unlike the fsNB, however, the L_1/L_2-regularized naïve Bayes approach does not discard redundant predictors and, hence, performs poorly when the data set contains large sets of these noisy predictors. This phenomenon has been discussed and observed in a comprehensive synthetic experimental setting. L_1/L_2-regularized naïve Bayes fares relatively well, though, when noisy predictors are irrelevant. Nonetheless, irrelevant predictors are considerably less harmful to the classification than redundant predictors.

In addition, note that, whereas it is straightforward for the fsNB approach to deal with data sets with both discrete and continuous predictors, it is not so simple for the L_1/L_2-regularized naïve Bayes method. This is because the continuous and discrete penalties scale differently. Besides discretizing the continuous predictors, we have two choices to address this issue. First, we can use two separate regularization parameters for each type of penalty, which is an expensive solution if they have to be estimated. Second, we can somehow scale the continuous predictors to make the penalties scale similarly. This is an approximate and rather tricky solution, and we do not expect the results to be good.

Also, the WNB approach cannot be used with continuous predictors unless they are discretized beforehand. In summary, flexibility is another advantage of the proposed fsNB approach.

Future work could focus on the possibility of converting the fsNB approach into a boosting method, where all intermediate models collaborate to output a final prediction. Plugging more complex Bayesian classifiers into this framework is also on the agenda. Of course, the algorithm structure accepts other distributions than multinomial and Gaussian.

Acknowledgments Research partially supported by the Spanish Ministry of Science and Innovation, projects TIN2010-20900-C04-04, Consolider Ingenio 2010-CSD2007-00018 and Cajal Blue Brain.

References

1. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**, 716–723 (1974)
2. Bergmann, G., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: Multiple Hypotheses Testing, pp. 100–115. Springer, Berlin (1988)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Domingos, P., Pazzani, M.: Beyond independence: conditions for the optimality of the simple Bayesian classifier. Mach. Learn. **29**, 103–130 (1997)
5. Drugan, M.M., Wiering, M.A.: Feature selection for Bayesian network classifiers using the MDL-FS score. Int. J. Approx. Reason. **51**, 695–717 (2010)
6. Ferreira, J.T.A.S., Denison, D.G.T., Hand, D.J.: Data mining with products of trees. In: Advances in Intelligent Data Analysis, Lecture Notes in Computer Science, vol. 2189, pp. 167–176. Springer, Berlin (2001)
7. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**, 131–163 (1997)
8. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J. Mach. Learn. Res. **9**, 2677–2694 (2008)
9. van Gerven, M., Heskes, T.: L_1/L_p regularization of differences. Tech. Rep. ICIS-R08009, Radboud University Nijmegen (2008)
10. Hall, M.: Induction of selective Bayesian classifiers. In: Proceedings of the 17th International Conference on Machine Learning, pp. 359–366 (2000)

11. Hand, D.J., Yu, K.: Idiot's Bayes—not so stupid after all. Int. Stat. Rev. **69**, 385–398 (2001)
12. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **29**, 273–324 (1996)
13. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, pp. 399–406 (1994)
14. Minsky, M.: Steps toward artificial intelligence. In: Computers and Thought, pp. 406–450. McGraw-Hill, New York (1961)
15. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Royal Stat. Soc. Ser. B **58**, 267–288 (1996)
16. Tseng, P.: Convergence of block coordinate descent method for nondifferentiable minimation. J. Optim. Theory Appl. **109**, 475–494 (2001)
17. Weisberg, S.: Applied Linear Regression. Wiley, New York (1980)
18. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. Royal Stat. Soc. Ser. B **70**, 53–71 (2006)
19. Zou, H.: The adaptive Lasso and its oracle properties. J. Am. Stat. Assoc. **101**, 1418–1429 (2006)