International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

SELECTIVE NAIVE BAYES FOR REGRESSION BASED ON MIXTURES OF TRUNCATED EXPONENTIALS*

MARÍA MORALES, CARMELO RODRÍGUEZ and ANTONIO SALMERÓN†

Department of Statistics and Applied Mathematics
University of Almería
Carrera de Sacramento s/n
E-04120 Almería (Spain)
{*maria.morales,crt,antonio.salmeron*}*@ual.es*

Received (received date)
Revised (revised date)

Naive Bayes models have been successfully used in classification problems where the class variable is discrete. These models have also been applied to regression or prediction problems, i.e. classification problems where the class variable is continuous, but usually under the assumption that the joint distribution of the feature variables and the class is multivariate Gaussian. In this paper we are interested in regression problems where some of the feature variables are discrete while the others are continuous. We propose a Naive Bayes predictor based on the approximation of the joint distribution by a Mixture of Truncated Exponentials (MTE). We have followed a filter-wrapper procedure for selecting the variables to be used in the construction of the model. This scheme is based on the mutual information between each of the candidate variables and the class. Since the mutual information can not be computed exactly for the MTE distribution, we introduce an unbiased estimator of it, based on Monte Carlo methods. We test the performance of the proposed model in artificial and real-world datasets.

Keywords: Bayesian networks, mixtures of truncated exponentials, naive Bayes models, probabilistic prediction, regression.

1. Introduction

The problem of *classification* consists of determining the class to which an individual belongs given that some features about that individual are known. In other words, classification means to predict the value of a *class* variable given the value of some other *feature* variables. Naive Bayes models have been successfully employed in classification problems where the class variable is discrete.[7] A naive Bayes model is a particular class of Bayesian network. A Bayesian network is a decomposition of a joint distribution as a product of conditionals, according to the independence

*This work has been supported by the Spanish Ministry of Education and Science, through project TIN2004-06204-C03-01 and by FEDER funds.
†Author for correspondence.

assumptions induced by the structure of a directed acyclic graph in which each vertex corresponds to one of the variables in the distribution,[14] and attached to each node there is a conditional distribution for it given its parents. The naive Bayes structure is obtained as a graph with the class variable as root whose only arcs are those that aim from the class variable to each one of the features.

When the class variable is continuous, the problem of determining the value of the class for a given configuration of values of the feature variables is called *regression* or *prediction* rather than classification. Naive Bayes models have been applied to regression problems under the assumption that the joint distribution of the feature variables and the class is multivariate Gaussian.[9] It implies that the marginals for all the variables in the model should be Normal as well, and therefore the model is not valid for discrete variables.

When the normality assumption is not fulfilled, the problem of regression with naive Bayes models has been approached using kernel densities to model the conditional distributions in the Bayesian network.[6] However, in both models (Gaussian and kernel-based), the results are poor compared to the performance of the M5' algorithm.[27] Furthermore, the use of kernels introduces a high complexity in the model (there is one term in the density for each sample point), which can be problematic especially if the regression model is a part of a larger Bayesian network, because standard algorithms for carrying out the computations in Bayesian networks are not directly valid for kernels.

In this paper we are interested in regression problems where some of the feature variables are discrete or qualitative while the others are continuous. We propose a Naive Bayes predictor based on the approximation of the joint distribution by a Mixture of Truncated Exponentials (MTE). The MTE model has been proposed in the context of Bayesian networks as a solution to the presence of discrete and continuous variables simultaneously,[13] and can perform well as an exact model as well as an approximation of other probability distributions.[2,3]

The rest of the paper is organised as follows. In section 2 we review the necessary concepts of Bayesian networks and explain how they can be used for regression. The MTE model is introduced in section 3. Section 4 is devoted to the comparison of the MTE model with other approaches to handle discrete and continuous variables simultaneously in Bayesian networks. Afterwards, we propose the naive Bayes regression models based on MTEs in section 5, and a variable selection scheme for it in section 6. The performance of the proposed model is experimentally tested in section 7. The paper ends with conclusions in section 8.

2. Bayesian networks and regression

Consider a problem defined by a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. A *Bayesian network* is a directed acyclic graph where each variable is assigned to one node, which has associated a conditional distribution given its parents.[10,14] An arc linking two variables indicates the existence of probabilistic dependence between them.

An important feature of Bayesian networks is that the joint distribution over $\mathbf{X}$ factorises according to the d-separation criterion as follows:[14]

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|pa(x_i)) \ , \tag{1}$$

where $Pa(X_i)$ denotes the set of parents of variable X_i and $pa(x_i)$ is a configuration of values of them. Figure 1 shows a Bayesian network which encodes the distribution

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_5|x_3)p(x_4|x_2, x_3) \ .$$

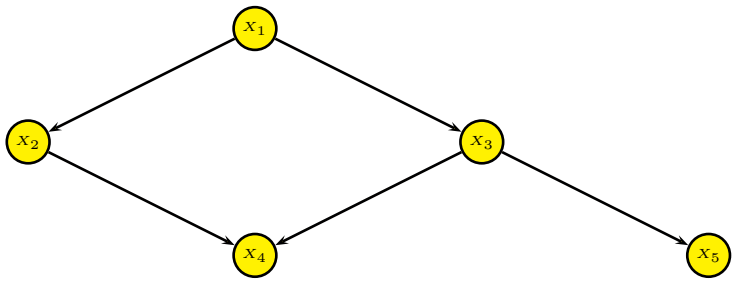

Fig. 1. A sample Bayesian network.

A Bayesian network can be used for classification purposes if it consists of a *class* variable, C, and a set of *feature* variables $X_1, \ldots, X_n$, so that an individual with observed features $x_1, \ldots, x_n$ will be classified as a member of class c^* obtained as

$$c^* = \arg\max_{c \in \Omega_C} p(c|x_1, \ldots, x_n) \ , \tag{2}$$

where Ω_C denotes the support of variable C. Similarly, a Bayesian network can be used for regression, i.e, when C is continuous. However, in this case the goal is to compute the posterior distribution of the class variable given the observed features $x_1, \ldots, x_n$, and once this distribution is computed, a numerical prediction can be given using some characteristic of the distribution, as the mean or the median.

Note that $p(c|x_1, \ldots, x_n)$ is proportional to $p(c) \times p(x_1, \ldots, x_n|c)$, and therefore solving the regression problem would require specifying an n dimensional distribution for $X_1, \ldots, X_n$ given the class. Using the factorisation determined by the Bayesian network, this problem is simplified. The extreme case is the so-called

naive Bayes structure,[4,7] where all the feature variables are considered independent given the class. An example of naive Bayes structure can be seen in Fig. 2.

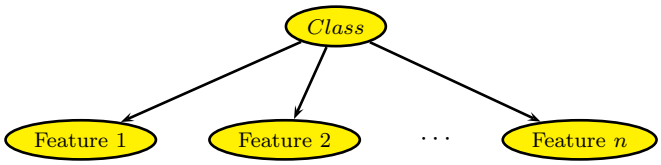

Fig. 2. Structure of a Naive Bayes classifier/predictor.

The independence assumption behind naive Bayes models is somehow compensated by the reduction on the number of parameters to be estimated from data, since in this case, it holds that

$$p(c|x_1,\ldots,x_n) = p(c)\prod_{i=1}^{n} p(x_i|c) \ , \tag{3}$$

which means that, instead of one n-dimensional conditional distribution, n one-dimensional conditional distributions are estimated.

3. The MTE model

Throughout this paper, random variables will be denoted by capital letters, and their values by lowercase letters. In the multi-dimensional case, boldfaced characters will be used. The domain of the variable $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. The MTE model is defined by its corresponding potential and density as follows:[13]

Definition 1. (MTE potential) Let $\mathbf{X}$ be a mixed n-dimensional random vector. Let $\mathbf{Y} = (Y_1,\ldots,Y_d)$ and $\mathbf{Z} = (Z_1,\ldots,Z_c)$ be the discrete and continuous parts of $\mathbf{X}$, respectively, with $c + d = n$. We say that a function $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ is a *Mixture of Truncated Exponentials potential (MTE potential)* if for each fixed value $\mathbf{y} \in \Omega_{\mathbf{Y}}$ of the discrete variables $\mathbf{Y}$, the potential over the continuous variables $\mathbf{Z}$ is defined as:

$$f(\mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{\sum_{j=1}^{c} b_i^{(j)} z_j\right\} \tag{4}$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$, where a_i, $i = 0,\ldots,m$ and $b_i^{(j)}$, $i = 1,\ldots,m$, $j = 1,\ldots,c$ are real numbers. We also say that f is an MTE potential if there is a partition $D_1,\ldots,D_k$ of $\Omega_{\mathbf{Z}}$ into hypercubes and in each D_i, f is defined as in Eq. (4).

Example 1. The function ϕ defined as

$$\phi(z_1, z_2) = \begin{cases} 2 + e^{3z_1+z_2} + e^{z_1+z_2} & \text{if } 0 < z_1 \leq 1,\ 0 < z_2 < 2 \\ 1 + e^{z_1+z_2} & \text{if } 0 < z_1 \leq 1,\ 2 \leq z_2 < 3 \\ 0.25 + e^{2z_1+z_2} & \text{if } 1 < z_1 < 2,\ 0 < z_2 < 2 \\ 0.5 + 5e^{z_1+2z_2} & \text{if } 1 < z_1 < 2,\ 2 \leq z_2 < 3 \end{cases}$$

is an MTE potential since all of its parts are MTE potentials.

Definition 2. (MTE density) An MTE potential f is an *MTE density* if

$$\sum_{\mathbf{y} \in \Omega_{\mathbf{Y}}} \int_{\Omega_{\mathbf{Z}}} f(\mathbf{y}, \mathbf{z}) d\mathbf{z} = 1 \ .$$

A *conditional MTE density* can be specified by dividing the domain of the conditioning variables and specifying an MTE density for the conditioned variable for each configuration of splits of the conditioning variables. Moral et al.[13] propose a data structure to represent MTE potentials based on a mixed tree data structure. The formal definition of a mixed tree is as follows:

Definition 3. (Mixed tree) We say that a tree $\mathcal{T}$ is a *mixed tree* if it meets the following conditions:

i. Every internal node represents a random variable (either discrete or continuous).
ii. Every arc outgoing from a continuous variable Z is labeled with an interval of values of Z, so that the domain of Z is the union of the intervals corresponding to the arcs emanating from Z.
iii. Every discrete variable has a number of outgoing arcs equal to its number of states.
iv. Each leaf node contains an MTE potential defined on variables in the path from the root to that leaf.

Mixed trees can represent MTE potentials defined by parts. Each branch in the tree determines one sub-region of the space where the potential is defined, and the function stored in the leaf of a branch is the definition of the potential in the corresponding sub-region.

Example 2. Consider a regression model with continuous class variable X, and with two features Y and Z, where Y is continuous and Z is discrete. One example of conditional densities for this regression model is given by the following expressions:

$$f(x) = \begin{cases} 1.16 - 1.12e^{-0.02x} & \text{if } 0.4 \leq x < 4 \ , \\ 0.9e^{-0.35x} & \text{if } 4 \leq x < 19 \ . \end{cases} \tag{5}$$

$$f(y|x) = \begin{cases} 1.26 - 1.15e^{0.006y} & \text{if } 0.4 \leq x < 5,\ 0 \leq y < 13\ , \\ 1.18 - 1.16e^{0.0002y} & \text{if } 0.4 \leq x < 5,\ 13 \leq y < 43\ , \\ 0.07 - 0.03e^{-0.4y} + 0.0001e^{0.0004y} & \text{if } 5 \leq x < 19,\ 0 \leq y < 5\ , \\ -0.99 + 1.03e^{0.001y} & \text{if } 5 \leq x < 19,\ 5 \leq y < 43\ . \end{cases} \tag{6}$$

$$f(z|x) = \begin{cases} 0.3 & \text{if } z = 0,\ 0.4 \leq x < 5\ , \\ 0.7 & \text{if } z = 1,\ 0.4 \leq x < 5\ , \\ 0.6 & \text{if } z = 0,\ 5 \leq x < 19\ , \\ 0.4 & \text{if } z = 1,\ 5 \leq x < 19\ . \end{cases} \tag{7}$$

The representation of the densities in Eq. (5) and (7) can be seen in Figures 3 and 4 respectively. We do not show the representation of the potential in Eq. (6) as it is analogous to the others.

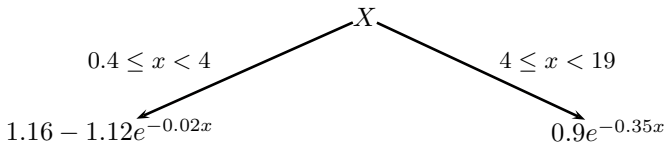

Fig. 3. A mixed probability tree representing the density in Eq. (5).

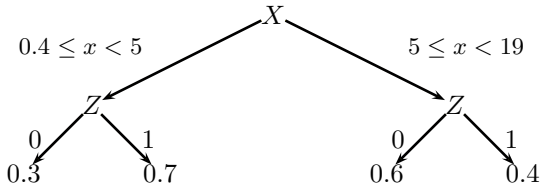

Fig. 4. A mixed probability tree representing the density in Eq. (7).

4. Comparison with other approaches

As we mentioned in the introduction, we are concerned with problems where continuous and discrete (or even qualitative) variables appear simultaneously. It is possible

to construct, from a modeling perspective, Bayesian networks where this situation appears. However, as far as we know there are not many solutions where efficient reasoning and learning algorithms have been successfully developed (see the works on MTEs[13,22] for a deeper discussion).

The most general solution consists of discretising the continuous variables, so that the regression problem is transformed into a regular classification problem for qualitative attributes. The drawback of this approach is that it is a rather rough approximation, and therefore the accuracy of the obtained models is very domain-dependent.

A well known model for which efficient learning and inference algorithms exist, and that does not require any approximation, is the so-called *conditional Gaussian* (CG) model,[11,12] where the joint distribution over the discrete variables is multinomial, and for each configuration of the discrete variables, the joint distribution over the continuous ones is multivariate normal. An important limitation of the CG model is that networks where discrete variables have continuous parents are not allowed, and therefore, networks with structures like the naive Bayes, where the class is continuous, are not possible.

If no discrete variables have continuous parents, the Gaussian model is perfectly valid and has been applied in several settings, including classification.[8,9,16] In those works, the conditional distribution for a continuous variable given its continuous parents is modeled as a normal distribution where the mean is a linear function of the parents (this is called the *conditional linear Gaussian* (CLG) model).

A comparison of the MTE model versus discretisation and the CLG model for continuous densities can be found in the paper by Cobb et al.[2] where it is shown that the MTE is competitive with the other ones.

An advantage of MTE models with respect to the Gaussian case is that no restriction on the structure of the network is applied, which means that the naive Bayes structure is perfectly valid. Furthermore, Cobb et al.[3] showed that the MTE distribution can approximate several well known distributions, including the normal, very accurately, and they even give the expression for transforming a normal density into an MTE.

A recent development in the field of Gaussian models was introduced by Shenoy.[25] He developed an efficient and exact algorithm for reasoning in networks where the conditional distributions are represented as mixtures of Normal densities. The problem of discrete variables with continuous parents is handled by transforming the structure of the network by means of arc reversal operations, until no discrete variable has continuous parents. The drawback of this approach is that it is only valid when we already have the network. If we were learning from data, we would have to use a different model to learn the distribution of the discrete variables with continuous parents, and then transform it, after arc reversal, into a mixture of Gaussians. So far, we are not aware of learning algorithms able to recover network structures and parameters taking this into account.

Therefore, according to this discussion, and considering the fact that there exist

efficient algorithms for learning[22] and reasoning,[21,23] our opinion is that the MTE approach reaches a compromise between generality and flexibility.

Regression problems have been approached from two perspectives using Bayesian networks. The first one uses a naive Bayes structure, where the conditional densities and the prior density of the class are modeled using Gaussian kernels.[6] This model allows the presence of discrete feature variables, since the conditional distribution is obtained as a quotient of kernel densities, using Bayes' rule. One problem of the resulting model, as pointed out by the authors,[6] is that it is not competitive with the M5' algorithm[27] for most of the test databases. Another drawback of kernels is that they are not efficient from the point of view of Bayesian networks, since they have one term for each sample point. In stand-alone regression models, it is not a big problem, but if the regression model is a part of a bigger network, it can be seriously inefficient, especially for reasoning tasks, which involve the multiplication of densities, whose size would increase exponentially. The MTE framework incorporates the good features of kernels without their efficiency problems,[18,20] as it is briefly commented in section 5.

The approach based on the Gaussian model[9], has the important restriction that it is not valid with discrete feature variables. Furthermore, the experiments reported by Gámez and Salmerón,[9] for continuous variables, show that the model is outperformed also by M5'.

The M5' algorithm[27] is an improved version of the model tree introduced by Quinlan.[17] The model tree is basically a decision tree where the leaves contain a regression model rather than a single value, and the splitting criterion uses the variance of the values in the database corresponding to each node rather than the information gain.

5. The naive Bayes regression model based on MTEs

Our proposal consists of solving the regression problem in which some feature variables are discrete and some other continuous using a regression model with naive Bayes structure, and modeling the corresponding conditional distributions as MTEs. More precisely, we will use a 5-parameter MTE for each split of the support of the variable, which means that in each split there will be 5 parameters to be estimated from data:

$$f(x) = a_0 + a_1 e^{a_2 x} + a_3 e^{a_4 x}, \quad \alpha < x < \beta \ . \tag{8}$$

The reason to use the 5-parameter MTE is that it has shown its ability to fit the most common distributions accurately, while the model complexity and the number of parameters to estimate is low.[3]

We follow the estimation procedure developed by Rumí et al.,[22] using the improvements proposed by Romero et al.,[18] which has these main steps:

(1) A Gaussian kernel density is fitted to the data.

(2) The domain of the variable is split according to changes in concavity/convexity or increase/decrease in the kernel.
(3) In each split, a 5-parameter MTE is fitted to the kernel by least squares estimation.

In this way we get a smoothed estimation of the target density, as using kernels, but considerably reducing the number of terms necessary to represent the density.[20]

Once the model is constructed, it can be used to predict the value of the class variable given that the values of the feature variables are observed. The forecasting is carried out by computing the posterior distribution of the class given the observed values for the features. A numerical prediction for the class value can be obtained from the posterior distribution, through its mean or its median. The choice of the mean or the median is problem-dependent. A situation in which the median can be more robust is when the training data contains outliers, and therefore the mean can be very biased towards the outliers. In this paper we have computed the posterior distribution using the Shenoy-Shafer algorithm[24] for probability updating in Bayesian networks, but adapted to the MTE case.[21,23] However, any other inference algorithm that does not require divisions would be valid.[13]

The expected value of a random variable X with a density defined as in Eq. (8) is computed as

$$\begin{aligned} E[X] &= \int_{-\infty}^{\infty} x f(x) dx = \int_{\alpha}^{\beta} x(a_0 + a_1 e^{a_2 x} + a_3 e^{a_4 x}) dx \\ &= a_0 \frac{\beta^2 - \alpha^2}{2} + \frac{a_1}{a_2^2}\left((a_2\beta - 1)e^{a_2\beta} - (a_2\alpha - 1)e^{a_2\alpha}\right) + \\ &\quad \frac{a_3}{a_4^2}\left((a_4\beta - 1)e^{a_4\beta} - (a_4\alpha - 1)e^{a_4\alpha}\right) \ . \end{aligned}$$

If the density is defined by parts, the expected value would be the sum of the expression above in each one of the parts.

The expression of the median, however, cannot be obtained in closed form, since the corresponding distribution function cannot be inverted. Therefore, we have decided to estimate it using the search procedure described below, which approximates the median with an error lower than 10^{-3} in terms of probability. The input parameter for the algorithm is the n-part density function, i.e.,

$$f(x) = f_i(x) \quad \alpha_i < x < \beta_i, \quad i = 1, \ldots, n \ ,$$

where each f_i is defined as in (8).

Algorithm MEDIAN
INPUT:

- A density f over interval (α_i, β_i).

OUTPUT:

- An estimation of the median of a random variable with density f, with error lower that 10^{-3} in terms of probability.

(1) $found := false$; $accum := 0.0$; $i := 0$.
(2) `While` $((found == false)$ `and` $(i \leq n))$
  (a) $m := \int_{\alpha_i}^{\beta_i} f_i(x)dx$.
  (b) `If` $(accum + m) \geq 0.5$, $found := true$.
  (c) `Else`
    - $i := i + 1$.
    - $accum := accum + m$.
(3) $max := \beta_i$; $min := \alpha_i$; $found := false$.
(4) `While` $(found == false)$
  (a) $mid := (max + min)/2$
  (b) $p := acum + \int_{min}^{mid} f_i(x)dx$
  (c) `If` $(\lfloor 0.5 * 1000 \rfloor == \lfloor p * 1000 \rfloor)$, $found := true$.
  (d) `Else`
    - `If` $(p > 0.5)$ $max := mid$
    - `Else` $min := mid$.
(5) `Return` mid.

6. Selecting the feature variables

An important issue to address in any classification or regression problem is to choose the feature variables to be included in the model. In general, it is not true that including more variables increases the accuracy of the model. It can happen that some variables are not informative for the class and therefore including them in the model provides noise to the predictor. Additionally, unnecessary variables cause an increase in the number of parameters that need to be determined from data.

There are different approaches to the problem of selecting variables in regression and classification problems:

- The *filter* approach, which in its simplest formulation consists of establishing a ranking of the variables according to some measure of relevance respect to the class variable, usually called *filter measure*. Then a threshold for the ranking is selected and variables below that threshold are discarded.

- The *wrapper* approach proceeds by constructing several models with different sets of feature variables, and finally the model with higher accuracy is selected.
- The *filter-wrapper* approach is a mixture of the former ones.[19] First of all, the variables are ordered using a filter measure and then they are incrementally included or excluded from the model according to that order, so that a variable is included whenever it increases the accuracy of the model.

We measure the accuracy of a model in this way:

(1) The database containing the information for the feature variables and the class is divided into two parts, D_l and D_t.
(2) The model is estimated using database D_l.
(3) The accuracy of the model is measured using database D_t, by measuring the root mean squared error between the actual values of the class and those ones predicted by the model for the records in database D_t. If we call $c_1, \ldots, c_m$ the values of the class for the registers in database D_t and $\hat{c}_1, \ldots, \hat{c}_m$ the corresponding estimates provided by the model, the root mean squared error is obtained as[28]

$$rmse = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(c_i - \hat{c}_i)^2} \ . \tag{9}$$

As the wrapper approach may be too costly, in this paper we have followed a filter-wrapper approach, based on the one proposed by Ruiz et al.,[19] using as filter measure the *mutual information* between each variable and the class. The mutual information has been successfully applied as filter measure in classification problems with continuous features.[15] The mutual information between two random variables X and Y is defined as

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log_2 \frac{f_{XY}(x,y)}{f_X(x)f_Y(y)}dydx \ , \tag{10}$$

where f_{XY} is the joint density for X and Y, f_X is the marginal density for X and f_Y is the marginal for Y.

In the case of MTE potentials, where each density is expressed as in Eq. (8), the integral in Eq. (10) cannot be obtained in closed form. Therefore, we have chosen to estimate the value of the mutual information. The estimation procedure that we have designed is based on the following results.

Proposition 1. *Let X and Y be two continuous random variables with densities f_X and f_Y respectively, and joint density f_{XY}. Let $f_{X|Y}$ denote the conditional density of X given Y. Let $Y_1, \ldots, Y_n$ be a sample drawn independently from distribution $f_Y(y)$. Let $X_1, \ldots, X_n$ be a sample such that each X_i, $i = 1, \ldots, n$ is drawn from*

distribution $f_{X|Y}(x|Y_i)$. Then,

$$\hat{I}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right) \tag{11}$$

is an unbiased estimator of $I(X,Y)$.

Proof. According to the way in which samples $X_1,\ldots,X_n$ and $Y_1,\ldots,Y_n$ are obtained, it follows that the joint sample of bivariate points (X_1,Y_1),..., (X_n,Y_n) is actually drawn from the distribution $f_{XY}(x,y)$. Therefore, if we denote by $E_{f_{XY}}$ the expected value with respect to density f_{XY}, we have that

$$\begin{aligned}
E[\hat{I}(X,Y)] &= E_{f_{XY}}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} E_{f_{XY}}\left[\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} E_{f_{XY}}\left[\log_2 \frac{f_{X|Y}(X_i|Y_i)}{f_X(X_i)}\right] = E_{f_{XY}}\left[\log_2 \frac{f_{X|Y}(X|Y)}{f_X(X)}\right] \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log_2 \frac{f_{X|Y}(x|y)}{f_X(x)}dydx \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log_2 \frac{f_{XY}(x,y)}{f_X(x)f_Y(y)}dydx \\
&= I(X,Y)\ .
\end{aligned}$$

□

Therefore, in order to estimate the mutual information between two variables X and Y, it is required to generate a sample of pairs (X_i,Y_i) and then apply formula (11). The way to sample from an MTE density is described in Moral et al.[13]

The consistency of estimator $\hat{I}(X,Y)$ is guaranteed by the next proposition.

Proposition 2. *Let $\hat{I}_n(X,Y)$ denote estimator $\hat{I}(X,Y)$ when it is computed from a sample of size n. The succession $\{\hat{I}_n(X,Y)\}_{n=1}^{\infty}$ is consistent.*

Proof. It is enough to show that

(i) $\lim_{n\to\infty} E[\hat{I}_n(X,Y)] = I(X,Y)$ and
(ii) $\lim_{n\to\infty} \mathrm{Var}(\hat{I}_n(X,Y)) = 0$.

The proof of (i) is trivial, since according to theorem 1, $E[\hat{I}_n(X,Y)] = I(X,Y)$ for all $n > 0$ and therefore the limit is equal to $I(X,Y)$ as well.

In order to prove (ii), we need the expression of the variance of the estimator.

$$\begin{aligned}
\mathrm{Var}(\hat{I}_n(X,Y)) &= \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n}\left(\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right)\right) \\
&= \frac{1}{n^2}\mathrm{Var}\left(\sum_{i=1}^{n}\left(\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right)\right) \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}\left(\log_2 f_{X|Y}(X_i|Y_i) - \log_2 f_X(X_i)\right) \\
&= \frac{1}{n^2}n\mathrm{Var}\left(\log_2 f_{X|Y}(X|Y) - \log_2 f_X(X)\right) \\
&= \frac{1}{n}\mathrm{Var}\left(\log_2 f_{X|Y}(X|Y) - \log_2 f_X(X)\right)
\end{aligned}$$

where $\mathrm{Var}\left(\log_2 f_{X|Y}(X|Y) - \log_2 f_X(X)\right)$ does not depend on n and is finite whenever distributions $f_{X|Y}$ and f_X are positive. Therefore, we can conclude that

$$\lim_{n\to\infty}\mathrm{Var}(\hat{I}_n(X,Y)) = 0 \ . \qquad \square$$

Notice that the consistency is guaranteed only if the sample used to estimate the mutual information is taken from the actual density f_{XY}. In our setting we are learning from data, and therefore we do not know any exact density. Instead we use MTE estimates, which rely on the behaviour of the estimation procedure of the parameters of the MTE density, which is based on least squares estimation. In any case, as consistency is a limit issue, it is not problematic, as we can rely on the estimations of the MTE densities for large samples.[22]

Now we have the necessary tools for giving a detailed algorithm for constructing the selective naive Bayes regression model. The idea of the algorithm is to start with a model containing only one feature variable, namely the one with highest mutual information with the class. Afterwards, the rest of the variables are included in the model in sequence, according to their mutual information with the class. In each step, if the included variable increases the accuracy of the model, it is kept. Otherwise, it is discarded.

Algorithm Selective naive Bayes regression model
INPUT:

- The class variable C.
- The feature variables $X_1, \ldots, X_n$.
- A database D for variables $X_1, \ldots, X_n, C$.

OUTPUT:

- The selective naive Bayes predictor for variable C.

(1) For $i := 1$ to n

- Compute $\hat{I}(X_i, C)$.

(2) Let $X_{(1)}, \ldots, X_{(n)}$ be a decreasing order of the feature variables according to $\hat{I}(X_{(i)}, C)$.
(3) Divide the database into two sets, one for learning the model (D_l) and the other for testing the accuracy of the learnt model (D_t).
(4) Construct a naive Bayes predictor, M, for variables C and $X_{(1)}$:
   (a) Estimate a marginal MTE density for C, f_C, from database D_l.
   (b) Estimate a conditional MTE density for $X_{(1)}$ given C, $f_{X_{(1)}|C}$, from database D_l.
   (c) Let $rmse(M)$ be the estimated accuracy of model M using database D_t, according to formula (9).
(5) For $i := 2$ to n
   (a) Let M_i be the naive Bayes predictor obtained from M by adding variable $X_{(i)}$, i.e., by estimating a conditional density for $X_{(i)}$ given C, $f_{X_{(i)}|C}$, from database D_l.
   (b) Let $rmse(M_i)$ be the estimated accuracy of model M_i using database D_t, according to formula (9).
   (c) If $(rmse(M_i) \leq rmse(M))$
      - $M := M_i$.
(6) Return (M)

7. Experimental evaluation

In order to test the performance of the naive Bayes regression model, we used ten databases, one of them artificial and the rest corresponding to real-world problems.

For all the databases, we compared the following models:

- **NB(mean)**: Naive Bayes regression model including all the feature variables and predicting with the mean of the posterior distribution.
- **NB(median)**: Naive Bayes regression model including all the feature variables and predicting with the median of the posterior distribution.
- **SNB(mean)**: Selective naive Bayes regression model and predicting with the mean of the posterior distribution.
- **SNB(median)**: Selective naive Bayes regression model and predicting with the median of the posterior distribution.
- **LM**: A linear regression model with variable selection, as implemented in Weka 3.4.11.[28]
- **M5'**: A model tree[27] using the implementation in Weka 3.4.11.[28]

The regression models proposed in this paper have been included in the Elvira system,[5] available at `leo.ugr.es/elvira`. We did not include in the experiments the regression model based on kernels developed by Frank et al.[6], since it is outperformed by M5'. Regarding the Gaussian regression model, we did not consider it

Table 1. Description of the databases used in the experiments.

Database	# of records	# of discr. and qualit. vars.	# of cont. vars.
`artificial1`	50	1	3
`artificial2`	50	1	2
`performance`	101	3	8
`success`	101	3	8
`students`	354	4	9
`abalone`	4176	1	8
`bodyfat`	251	0	15
`cloud`	107	2	6
`pollution`	59	0	16
`strikes`	624	1	6
`veteran`	136	4	4

for the experiments because of the presence of discrete or qualitative variables with continuous parents (see the discussion in section 4), since most of the databases considered contain discrete or qualitative variables. In any case, even if all the variables are continuous, the M5' was reported to outperform the Gaussian regression model.[9]

A description of the databases used in this section can be found in Table 1. Databases `bodyfat`, `cloud`, `pollution`, `strikes` and `veteran` are taken from the StatLib repository,[26] and `abalone` from the UCI repository.[1]

The `artificial1` dataset consists of a random sample of 50 records drawn from a Bayesian network with naive Bayes structure and MTE distributions. The aim of this network is to represent a situation which is handled in a natural way by the MTE model, while other models like the Gaussian or a linear regression model are less natural, at least a priori. In order to obtain this network, we first simulated a database with 500 records for variables X, Y, Z and W, where X follows a χ^2 distribution with 5 degrees of freedom, Y follows a negative exponential distribution with mean $1/X$, $Z = \lfloor X/2 \rfloor$, where $\lfloor \cdot \rfloor$ stands for the integer part function, and W is a random variable with Beta distribution with parameters $p = 1/X$ an $q = 1/X$. Out of that database, a naive Bayes regression model was constructed. The obtained MTE Bayesian network is depicted in Fig. 5, where each box represents the prior MTE density for each variable. It can be seen that the model does not correspond with a CG one, because there is a discrete variable with continuous parent, and also because the marginals of the continuous variables are clearly not normal. A similar procedure has been followed to generate database `artificial2`, where we first simulated a database with 500 records for variables X, Y and Z, where X follows a standard normal distribution, Y follows a uniform distribution with lower limit equal to $X - 1$ and upper limit equal to $X + 1$, and $Z = \lfloor X + 2 \rfloor$.

Databases `performance`, `success` and `students` correspond to three problems

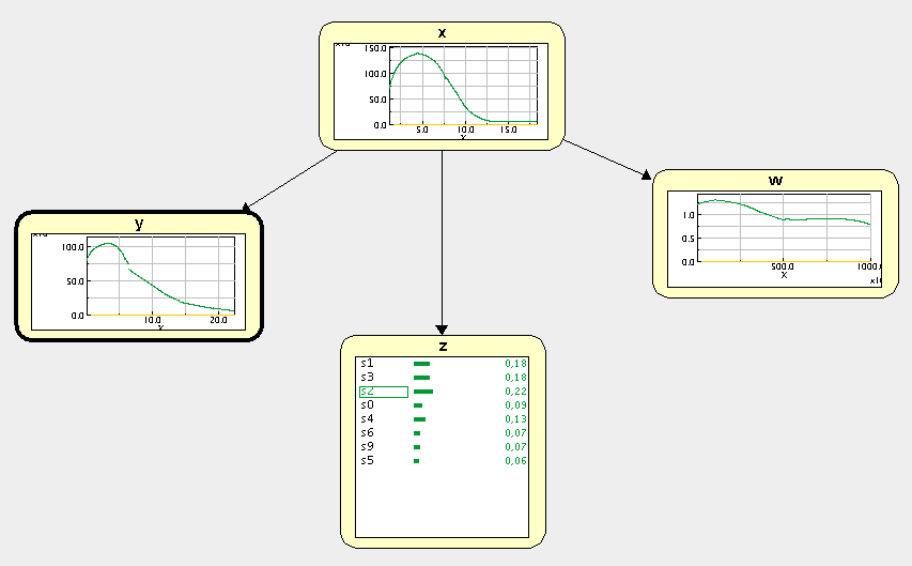

Fig. 5. An artificial MTE network.

related to higher education management. More precisely, they contain data regarding academic performance at the University of Almería (Spain).

Databases `performance` and `success` are aimed at the prediction of the *performance rate* and *success rate*, respectively, for a given degree program, and collect information about all the degree programs in the University of Almería in years 2001 to 2004.

The *performance rate* is defined as

$$pr = \frac{n_o}{n_s} \tag{12}$$

where n_s is the number of credits[1] of all the subjects selected by the students in a given year, and n_o is the number of credits actually obtained by the students at the end of the same year.

The *success rate* is defined as

$$sr = \frac{n_o}{n_e} \tag{13}$$

where n_o is as defined above and n_e is the number of credits for which the students actually showed up at the final exam.

[1]Spanish university subjects are measured in credits. One credit corresponds to ten hours of confrontation lectures.

Table 2. Performance of the different models in terms of root mean squared error. The best result for each database is marked in boldface, and the worst is underlined.

Database	NB(mean)	NB(median)	SNB(mean)	SNB(median)	LM	M5'
artificial1	1.8855	1.9372	1.7073	**1.5734**	2.5681	2.4718
artificial2	0.7234	0.6907	0.6914	**0.6625**	0.8243	0.8312
performance	0.088	0.0906	0.0813	0.0764	0.1045	**0.0651**
success	0.0432	0.0457	0.0374	0.0363	0.0467	**0.0359**
students	34.4036	35.0895	21.85	23.2301	17.9776	**17.7609**
abalone	2.7305	2.5806	2.5135	2.4609	2.2147	**2.1296**
bodyfat	4.9387	4.8694	**2.5698**	2.7193	24.8814	25.6409
cloud	0.5392	0.5382	0.4799	0.5186	0.3792	**0.3764**
pollution	42.0957	43.2251	**28.9971**	30.7210	42.7162	39.9143
strikes	473.9729	510.9138	**446.9318**	508.8195	532.666	518.6453
veteran	117.3975	115.4447	**112.9308**	114.4078	155.7577	155.8373

Table 3. Performance of the different models in terms of linear correlation coefficient. The best result for each database is marked in boldface, and the worst is underlined.

Database	NB(mean)	NB(median)	SNB(mean)	SNB(median)	LM	M5'
artificial1	0.6623	0.564	**0.798**	0.7757	0.579	0.6027
artificial2	0.7727	0.7932	0.8026	**0.8047**	0.6326	0.6217
performance	0.527	0.4851	0.6225	0.6509	0.5703	**0.8549**
success	0.5403	0.5142	0.6817	0.6864	0.5609	**0.7678**
students	0.8719	0.8595	0.9368	0.9313	0.9733	**0.9734**
abalone	0.6952	0.6981	0.6996	0.7020	0.7268	**0.7511**
bodyfat	0.7975	0.797	0.939	**0.9418**	0.4311	0.4568
cloud	0.8671	0.8641	0.8847	0.883	0.9354	**0.9364**
pollution	0.7243	0.7154	**0.8502**	0.816	0.7423	0.7641
strikes	0.5448	0.5599	0.5997	**0.6149**	0.3111	0.4233
veteran	0.5019	0.5152	0.5787	**0.5938**	0.2358	0.2354

Database `students` was used for predicting the number of students in a given subject. In this case, the database contained information about all the subjects offered at the University of Almería from years 2001 to 2004.

In order to measure the accuracy of the different models tested, we have computed, through a 10-fold cross validation process, the root mean squared error (rmse), as in Eq. (9), and the linear correlation coefficient between the predicted values of each model for the test part in each fold of the cross validation, and the exact values (in the ideal case, if the predictions matched the exact values, the linear correlation coefficient would be equal to 1). Table 2 shows the results obtained in terms of rmse and Table 3 the results in terms of linear correlation coefficient.

7.1. *Results discussion*

Out of the eleven databases, M5' gets the best performance in five cases, and in the other six the best model is the selective naive Bayes regression model. The differences are small in general, except for the case of database `bodyfat` in favour of SNB. We think that the poor behaviour of M5' in this database is due to the fact that the actual model is far from being linear (notice that the linear model in this problem is also little accurate). In order to determine whether the differences

between SNB and M5' can be considered statistically significant, we carried out a Wilcoxon signed rank test, due to the lack of normality of the samples, using as paired samples the rmse provided by each method. The result of the test provides no significant differences between both methods with a p-value of 0.2061.

It can be noticed that SNB never provides the worst result, while M5' is the worst in three cases. Regarding the linear model, it usually behaves more poorly than SNB, reaching the worst results in four tests.

Regarding the comparison between SNB and NB, the Wilcoxon signed rank test reports significant differences between them, favourable to SNB, with a p-value of 0.0004. Actually, for all the tested datasets the variable selection procedure always provides lower error. It can also be noticed that using the mean instead of the median for the numerical prediction gives more accurate estimations for the considered databases. However, the differences between both alternatives are minimal.

An added value of NB and SNB with respect to LM and M5' is that they do not only provide numerical prediction, but they also give the posterior distribution of the class variable, which allows to make other types of inferences like answering queries as *what is the probability of the number of students being between 100 and 150*. For instance, the regression model in Fig. 5 could be used for computing the probability $P\{a \leq X \leq b\}$ by calculating the area below the curve of the density of X between points a and b. Another important advantage is that the regression model represented by NB and SNB is a Bayesian network, and therefore it can be straightforwardly included as a part of a bigger Bayesian network that integrates knowledge from different sources.

8. Conclusions

In this paper we have introduced a framework for approaching regression problems with a mixture of discrete and continuous variables, based on the Bayesian network methodology using mixtures of truncated exponentials as underlying probabilistic model. We have also proposed a variable selection scheme according to the mutual information, and adopting a filter-wrapper strategy.[19]

The performance of the proposed model was tested in eleven databases, that correspond to artificial settings, practical applications related to higher education management, and standard databases commonly used as benchmark for regression problems. Out of the experimental results, the proposed model is competitive with the state-of-the-art method, the so-called M5'.

In future works, we plan to test the models in more real-world and synthetic datasets, and compare the performance of the selective naive Bayes predictor versus the Gaussian model developed by Gámez and Salmerón.[9] In order to carry out this comparison, it is necessary to solve the problem of incorporating discrete variables in a Gaussian regression model. Furthermore, more sophisticated variable selection strategies can be considered, as well as more complex network structures as the tree augmented naive Bayes (TAN) structure.[7]

Acknowledgements

We want to thank the anonymous referees for their helpful and valuable comments.

References

1. C.L. Blake and C.J. Merz. "UCI Repository of machine learning databases", www.ics.uci.edu/∼mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences. 1998.
2. B. Cobb, R. Rumí, and A. Salmerón. "Modeling conditional distributions of continuous variables in Bayesian networks". *IDA'05. Lect. Notes in Comp. Sci.* **3646** (2005) 36–45.
3. B. Cobb, P.P. Shenoy, and R. Rumí. "Approximating probability density functions with mixtures of truncated exponentials". *Stat. and Comp.* **16** (2006) 293–308.
4. R.O. Duda, P.E. Hart, and D.G. Stork. "Pattern classification". Wiley Interscience, 2001.
5. Elvira Consortium. "Elvira: An environment for creating and using probabilistic graphical models". In J.A. Gámez and A. Salmerón (eds.), *Procs. of the First European Workshop on Probabilistic Graphical Models*, Cuenca, Spain, 2002, pp. 222–230.
6. E. Frank, L. Trigg, G. Holmes, and I.H. Witten. "Technical note: Naive Bayes for regression". *Machine Learning* **41** (2000) 5–25.
7. N. Friedman, D. Geiger, and M. Goldszmidt. "Bayesian network classifiers". *Machine Learning* **29** (1997) 131–163.
8. N. Friedman, M. Goldszmidt and T.J. Lee. "Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting". In *Procs. of the Fifteenth Intnl. Conf. on Machine Learning*, 1998.
9. J.A. Gámez and A. Salmerón. "Predicción del valor genético en ovejas de raza manchega usando técnicas de aprendizaje automático". In *Actas de las VI Jornadas de Transferencia de Tecnología en Inteligencia Artificial* (Paraninfo, 2005), pp. 71–80.
10. Finn V. Jensen. "Bayesian networks and decision graphs". Springer, 2001.
11. S.L. Lauritzen and N. Wermuth. "Graphical models for associations between variables, some of which are qualitative and some quantitative". *The Annals of Statistics* **17** (1989) 31–57.
12. S.L. Lauritzen. "Propagation of probabilities, means and variances in mixed graphical association models". *Journal of the Am. Statistical Assoc.* **87** (1992) 1098–1108.
13. S. Moral, R. Rumí, and A. Salmerón. "Mixtures of truncated exponentials in hybrid Bayesian networks". *ECSQARU'01. Lect. Notes in Artif. Intel.* **2143** (2001) 135–143.
14. J. Pearl. "Probabilistic reasoning in intelligent systems". Morgan-Kaufmann (San Mateo), 1988.
15. A. Pérez, P. Larrañaga, and I. Inza. "Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes". *Int. Journal of Approximate Reasoning* **43** (2006) 1–25.
16. W.B. Poland and R.D. Shachter. "Mixtures of Gaussians and minimum relative entropy techniques for modeling continuous uncertainties". In D. Heckermann and E.H. Mamdani (eds.) *Uncertainty in Artificial Intelligence: Procs. of the Ninth Conf.*, Morgan Kaufmann, San Francisco, CA, 1993, pp. 183–190.
17. J.R. Quinlan. "Learning with continuous classes". In *Procs. of the 5th Australian Joint Conference on Artificial Intelligence*. World Scientific, Singapore, 1992, pp. 343–348.
18. V. Romero, R. Rumí, and A. Salmerón. "Learning hybrid Bayesian networks using mixtures of truncated exponentials". *Int. Journal of Approximate Reasoning* **42** (2006) 54–68.
19. R. Ruiz, J. Riquelme and J.S. Aguilar-Ruiz. "Incremental wrapper-based gene se-

lection from microarray data for cancer classification". *Pattern Recognition* **39** (2006) 2383–2392.
20. R. Rumí. "Kernel methods in Bayesian networks". In *Procs. of the 1st Int. Conf. of Mediterranean Mathematicians*, Almería, Spain, 2005.
21. R. Rumí and A. Salmerón. "Penniless propagation with mixtures of truncated exponentials". *ECSQARU'05. Lect. Notes in Comp. Sci.* **3571** (2005) 39–50.
22. R. Rumí, A. Salmerón and S. Moral. "Estimating mixtures of truncated exponentials in hybrid Bayesian networks". *Test* **15** (2006) 397–421.
23. R. Rumí and A. Salmerón. "Approximate probability propagation with mixtures of truncated exponentials". *Int. Journal of Approximate Reasoning* **45** (2007) 191–210.
24. P.P. Shenoy and G. Shafer. "Axioms for probability and belief function propagation". In R.D. Shachter, T.S. Levitt, J.F. Lemmer and L.N. Kanal (eds.). *Uncertainty in Artificial Intelligence 4*, North Holland, Amsterdam, 1990, pp. 169–198.
25. P.P. Shenoy. "Inference in hybrid Bayesian networks with mixtures of Gaussians". In R. Dechter and T. Richardson (eds.) *Uncertainty in Artificial Intelligence: Procs. of the Twenty Second Conf.*, Morgan Kaufmann, San Francisco, CA, 2006, pp. 428–436.
26. StatLib. Department of Statistics. Carnegie Mellon University. www.statlib.org. 1999.
27. Y. Wang and I.H. Witten. "Induction of model trees for predicting continuous cases". In *Procs. of the Poster Papers of the European Conf. on Machine Learning*, Prague, Czech Republic, 1997, pp. 128–137.
28. I.H. Witten and E. Frank. "Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)". Morgan Kaufmann, 2005.