PREPRINT

LEARNING BAYESIAN NETWORKS FOR REGRESSION FROM INCOMPLETE DATABASES

Cite as:

- A. Fernández, J.D. Nielsen, A. Salmerón. Learning Bayesian networks for regression from incomplete databases. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems* (to appear).

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

LEARNING BAYESIAN NETWORKS FOR REGRESSION FROM INCOMPLETE DATABASES*

Antonio Fernández

Department of Statistics and Applied Mathematics
University of Almería
04120 Almería, Spain
afalvarez@ual.es

Jens D. Nielsen

Computer Science Department
University of Castilla-La Mancha
02071 Albacete, Spain
dalgaard@dsi.uclm.es

Antonio Salmerón

Department of Statistics and Applied Mathematics
University of Almería
04120 Almería, Spain
antonio.salmeron@ual.es

Received (received date)
Revised (revised date)

In this paper we address the problem of inducing Bayesian network models for regression from incomplete databases. We use mixtures of truncated exponentials (MTEs) to represent the joint distribution in the induced networks. We consider two particular Bayesian network structures, the so-called naïve Bayes and TAN, which have been successfully used as regression models when learning from complete data. We propose an iterative procedure for inducing the models, based on a variation of the data augmentation method in which the missing values of the explanatory variables are filled by simulating from their posterior distributions, while the missing values of the response variable are generated using the conditional expectation of the response given the explanatory variables. We also consider the refinement of the regression models by using variable selection and bias reduction. We illustrate through a set of experiments with various databases the performance of the proposed algorithms.

Keywords: Bayesian networks, regression, mixtures of truncated exponentials, missing data

*This work has been supported by the Spanish Ministry of Science and Innovation, through projects TIN2007-67418-C03-01,02 and by EFRD funds. A preliminary version of this work was presented at the PGM'08 workshop.

1. Introduction

Mixtures of truncated exponentials (MTEs)[18] are receiving increasing attention in the literature, as a tool for handling hybrid Bayesian networks, as they are compatible with standard inference algorithms and no restriction on the structure of the network is imposed.[3,17,24] Recently, MTEs have also been successfully applied to regression problems considering different underlying network structures[8,9,20] obtained from complete databases. In a previous preliminary work[10] we approached the problem of inducing Bayesian networks for regression from incomplete databases by using an iterative algorithm for constructing naïve Bayes regression models. The algorithm was based on a variation of the data augmentation method[27] in which the missing values of the explanatory variables are filled by simulating from their posterior distributions, while the missing values of the response variable are filled with its conditional expectation given the explanatory variables. In this paper we extend the above mentioned method to obtain networks with TAN[12] structures. Also, the algorithm is extended to incorporate variable selection. Finally, we introduce a method for reducing the bias in the predictions that can be used in all the models, regardless they have been induced from complete or incomplete databases.

2. The MTE model

We denote random variables by capital letters, and their values by lowercase letters. We use boldfaced characters to represent random vectors and their values. The support of the variable $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. A potential of class MTE is defined as follows:[18]

Definition 1. (MTE potential) Let $\mathbf{X}$ be a mixed n-dimensional random vector. Let $\mathbf{W} = (W_1, \ldots, W_d)$ 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 $\mathbf{w} \in \Omega_{\mathbf{W}}$ of the discrete variables $\mathbf{W}$, the potential over the continuous variables $\mathbf{Z}$ is defined as:

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

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. (1).

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

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

A *conditional MTE density* can be specified by dividing the domain of the conditioning variables and specifying an MTE density for the dependent variable for each configuration of splits of the conditioning variables.[18,19]

Example 1. Consider two continuous variables X and Y. A possible conditional MTE density for Y given X is the following:

$$f(y|x) = \begin{cases} 1.26 - 1.15e^{0.006y} & \text{if } 0.4 \leq x < 5,\ 0 \leq y < 13\ , \\ 1.18 - 1.16e^{0.0002y} & \text{if } 0.4 \leq x < 5,\ 13 \leq y < 43\ , \\ 0.07 - 0.03e^{-0.4y} + 0.0001e^{0.0004y} & \text{if } 5 \leq x < 19,\ 0 \leq y < 5\ , \\ -0.99 + 1.03e^{0.001y} & \text{if } 5 \leq x < 19,\ 5 \leq y < 43\ . \end{cases} \quad (2)$$

3. Regression using MTEs

Assume we have a set of variables $Y, X_1, \ldots, X_n$, where Y is continuous and the rest are either discrete or continuous. Regression analysis consists of finding a model g that explains the *response* variable Y in terms of the *explanatory* variables $X_1, \ldots, X_n$, so that given an assignment of the explanatory variables, $x_1, \ldots, x_n$, a prediction about Y can be obtained as $\hat{y} = g(x_1, \ldots, x_n)$. Previous works on regression using MTEs[8,9,20] proceed by representing the joint distribution of $Y, X_1, \ldots, X_n$ as a Bayesian network, and then using the posterior distribution of Y given $X_1, \ldots, X_n$ (more precisely, its expectation) to obtain a prediction for Y. The learning procedure consists of fixing the structure and afterwards learning the parameters of the corresponding conditional densities using a procedure based on least squares estimation.[25]

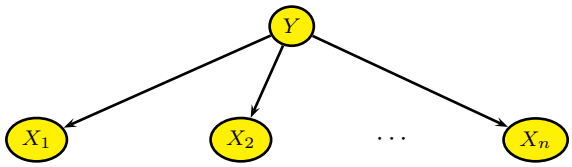

Fig. 1. Naïve Bayes structure for regression. The explanatory variables are assumed to be independent given the response variable Y.

In this paper we will focus on two particular Bayesian network structures, the so-called *naïve Bayes* (NB) and *Tree Augmented Naïve Bayes* (TAN). The NB[6] structure is an extreme case in which all the explanatory variables are considered independent given the response variable. This kind of structure is represented in figure 1. The reason to make the strong independence assumption behind NB models is that it is compensated by the reduction in the number of parameters to be

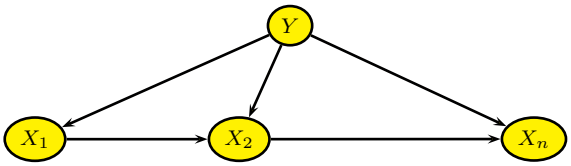

Fig. 2. A TAN structure for regression. Some more dependencies among the explanatory variables are allowed.

estimated from data, since in this case, it holds that the conditional distribution of the response variable can be factorised as

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

which means that, instead of one conditional density over a large domain ($n+1$ variables), n conditional densities over a smaller domain (2 variables) are estimated.

The TAN[12] represents a compromise between the strong independence assumption and the complexity of the model to be estimated from data. In this kind of models, additional dependencies are allowed, expanding the NB structure so that the subgraph over the explanatory variables is a directed rooted tree (see figure 2).

3.1. *Constructing a regression model from incomplete data*

As we use the conditional expectation of the response variable given the observed explanatory variables, our regression model will be

$$\hat{y} = g(x_1,\ldots,x_n) = E[Y|x_1,\ldots,x_n] = \int_{\Omega_Y} y f(y|x_1,\ldots,x_n)dy \ ,$$

where $f(y|x_1,\ldots,x_n)$ is the conditional density of Y given $x_1,\ldots,x_n$, which we assume to be of class MTE.

A conditional distribution of class MTE can be represented as in Eq. (2), where actually a marginal density is given for each element of the partition of the support of the variables involved. It means that, in each of the four regions depicted in Eq. (2), the distribution of the response variable Y is independent of the explanatory variables. Therefore, from the point of view of regression, the distribution for the response variable Y given an element in a partition of the domain of the explanatory variables $X_1,\ldots,X_n$, can be regarded as an approximation of the true distribution of the actual values of Y for each possible assignment of the explanatory variables in that region of the partition. This fact justifies the selection of $E[Y|x_1,\ldots,x_n]$ as the predicted value for the regression problem, because that value is the one that best represents all the possible values of Y for that region, in the sense that it

minimises the *mean squared error* between the actual value of Y and its predictions $\hat{y}$, namely

$$\text{mse} = \int_{\Omega_Y} (y - \hat{y})^2 f(y|x_1, \ldots, x_n) dy \ , \tag{4}$$

which is known to be minimised for $\hat{y} = E[Y|x_1, \ldots, x_n]$. Thus, the key point to find a regression model of this kind is to obtain a good estimation of the distribution of Y for each region of values of the explanatory variables. The original NB and TAN models[8,20] estimate that distribution by fitting a kernel density to the sample and then obtaining an MTE density from the kernel using least squares.[19,25] Obtaining such an estimation is more difficult in the presence of missing values. The first approach to estimating MTE distributions from incomplete data was developed in the more restricted setting of unsupervised data clustering.[14] In that case, the only missing values are on the class variable, which is hidden, while the data about the features are complete.

Here we are interested in problems where the missing values can appear in the response variable as well as in the explanatory variables. A first approach to solve this problem could be to apply the EM algorithm,[4] which is a commonly used tool in semi-supervised learning.[2] However, the application of this methodology is problematic because the likelihood function for the MTE model cannot be optimised in an exact way.[16,25]

Another way of approaching problems with missing values is the so-called *data augmentation* (DA) algorithm.[27] The advantage with respect to the EM algorithm is that DA does not require a direct optimisation of the likelihood function. Instead, it is based on imputing the missing values by simulating from the posterior distribution of the missing variables, which is iteratively improved from an initial estimation based on a random imputation. The DA algorithm leads to an approximation of the maximum likelihood estimates of the parameters of the model, as long as the parameters are estimated by maximum likelihood from the complete database in each iteration. As maximum likelihood estimates cannot be found in an exact way, we have chosen to use least squares estimation, as in the original NB and TAN regression models.

Furthermore, as our main goal is to obtain an accurate model for predicting the response variable Y, we propose to modify the DA algorithm in connection to the imputation of missing values of Y. The next proposition is the key on how to proceed in this direction.

Proposition 1. *Let Y and Y_S be two continuous independent and identically distributed random variables. Then,*

$$E[(Y - Y_S)^2] \geq E[(Y - E[Y])^2] \ . \tag{5}$$

Proof.

$$\begin{aligned} E[(Y - Y_S)^2] &= E[Y^2 + Y_S^2 - 2YY_S] \\ &= E[Y^2] + E[Y_S^2] - 2E[YY_S] \\ &= E[Y^2] + E[Y_S^2] - 2E[Y]E[Y_S] \\ &= 2E[Y^2] - 2E[Y]^2 \\ &= 2(E[Y^2] - E[Y]^2) \\ &= 2\mathrm{Var}(Y) \\ &\geq \mathrm{Var}(Y) = E[(Y - E[Y])^2] \ . \end{aligned}$$ □

In the proof we have relied on the fact that both variables are independent and identically distributed, and therefore the expectation of the product is the product of the expectations, and the expected value of both variables is the same.

Proposition 1 motivates our proposal for modifying the data augmentation algorithm, since it proves that using the conditional expectation of Y to impute the missing values instead of simulating values for Y (denoted as Y_S in the proposition), reduces the mse of the estimated regression model. Notice that it is true even if we are able to simulate from the exact distribution of Y conditional on any configuration on a region of the values of the explanatory variables.

3.2. *The algorithm for learning a regression model from incomplete data*

Our proposal consists of an algorithm which iteratively learns a regression model (which can be an NB or a TAN) by imputing the missing values in each iteration according to the following criterion:

- If the missing value corresponds to the response variable, it is imputed with the conditional expectation of Y given the values of the explanatory variables in the same record of the database, computed from the current regression model.
- Otherwise, the missing cell is imputed by simulating the corresponding variable from its conditional distribution given the values of the other variables in the same record, computed from the current regression model.

As the imputation requires the existence of a model, for the construction of the initial model we propose to impute the missing values by simulating from the marginal distribution of each variable computed from the observed values. In this way we have reached better results than using pure random initialisation, which is the standard way of proceeding in data augmentation.[27] Another way of proceeding could be to simulate from the conditional distribution of each explanatory variable given the response, but we rejected this option because the estimation of the conditional distributions requires more data than the estimation of the marginals, which can be problematic if the number of missing values is high.

Algorithm 1: Bayesian network regression model from missing data

Input: An incomplete database D for variables $Y, X_1, \ldots, X_n$. A test database D_t.

Output: A Bayesian network regression model for response variable Y and explanatory variables $X_1, \ldots, X_n$.

```
for each variable X ∈ {Y, X_1, ..., X_n} do
    Learn a univariate distribution f_X(x) from its observed values in D.
end
Create a new database D' from D by imputing the missing values for each
variable X ∈ {Y, X_1, ..., X_n} by simulating from f_X(x).
Learn a Bayesian network regression model M' from D'.
Let srmse' be the sample root mean squared error of M' computed using D_t
according to Eq. (6).
srmse ← ∞.
while srmse' < srmse do
    M ← M'.
    srmse ← srmse'.
    Create a new database D' from D by imputing the missing values as
    follows:
    for each variable X ∈ {X_1, ..., X_n} do
        for each record z in D with missing value for X do
            Obtain f_X(x|z) by probability propagation in model M.
            Impute the missing value for X by simulating from f_X(x|z).
        end
    end
    for each record z in D with missing value for Y do
        Obtain f_Y(x|z) by probability propagation in model M.
        Impute the missing value for Y with E_{f_Y}[Y|z].
    end
    Re-estimate model M' from D'.
    Let srmse' be the sample root mean squared error of M' computed using
    D_t.
end
return M
```

The algorithm (see algorithm 1) proceeds by imputing the initial database, learning an initial model and re-imputing the missing cells. Then, a new model is constructed and, if the mean squared error is reduced, the current model is replaced and the process repeated until convergence. As the mse in Eq. (4) requires the knowledge of the exact distribution of Y conditional on each configuration of the explanatory variables, we use as error measure the sample root mean squared error,

Algorithm 2: Selective Bayesian network regression model from missing data

Input: An incomplete database D for variables $Y, X_1, \ldots, X_n$. A test database D_t.

Output: A Bayesian network regression model made up of the response variable Y and a subset of explanatory variables $S \subseteq \{X_1, \ldots, X_n\}$.

```
1  for i ← 1 to n do
2  |  Compute Î(X_i, Y).
3  end
4  Let X_(1), ..., X_(n) be a decreasing order of the feature variables according to
   Î(X_(i), Y).
5  Using algorithm 1, construct a regression model M with variables Y and
   X_(1) from database D.
6  Let srmse(M) be the estimated accuracy of model M using D_t.
7  for i ← 2 to n do
8  |  Let M_1 be the model obtained by the algorithm 1 with the variables of
   |  M plus X_(i).
9  |  Let srmse(M_1) be the estimated accuracy of model M_1 using D_t.
10 |  if srmse(M_1) ≤ rmse(M) then
11 |  |  M ← M_1.
12 |  end
13 end
14 return M.
```

computed as

$$\text{srmse} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \ , \tag{6}$$

where m is the sample size, y_i is the observed value of Y for record i and $\hat{y}_i$ is its corresponding prediction through the regression model.

The details are given in algorithm 1. Notice that, in steps 5 and 22 the regression model is learnt from a complete database, and therefore the existing estimation methods for MTEs can be used.[25,20] Also, notice that the algorithm is valid for any Bayesian network structure, and therefore it is valid for our purpose, which is to learn an NB or a TAN, just by calling to the appropriate procedure in steps 5 and 22. For learning the NB regression model, we use the method described in Morales et al.[20] and for learning the TAN, the algorithm in Fernández et al.[8]

We have also incorporated variable selection in the construction of the regression models[9,20] as described in algorithm 2. We have followed a filter-wrapper approach, based on the one proposed by Ruiz et al.,[23] using as filter measure the *mutual information* between each variable and the class. The filter-wrapper approach proceeds

by sorting the variables according to a filter measure, and then constructing a series of models including the variables in sequence, one by one, in such a way that a variable is kept in the model only if it increases the accuracy with respect to the previous model.

The mutual information has been successfully applied as filter measure in classification problems with continuous features.[21] The mutual information between two random variables X and Y is defined as

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log_2 \frac{f_{XY}(x,y)}{f_X(x)f_Y(y)}dydx \ , \tag{7}$$

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 densities, the integral in Eq. (7) cannot be obtained in closed form. Therefore, we have estimated it by Monte Carlo.[20]

Algorithm 3: Computing a vector of bias to refine the predictions

Input: A full database D for variables $Y, X_1, \ldots, X_n$.
A regression model M.
Output: $vBias$, a vector of biases.

```
Run a hierarchical clustering to obtain a dendrogram for the values of Y.
Determine the number of clusters, numBias, using the dendrogram.
Partition D into numBias partitions D_1, ..., D_numBias by clustering Y
  using the k-means algorithm.
for i ← 1 to numBias do
    Compute vBias[i] by (8) using D_i and M.
end
return vBias, a vector of estimated expected biases.
```

4. Improving the final estimations by reducing the bias

In existing approaches for using MTEs for regression, the prediction that is used is a corrected version computed by subtracting an estimated expected bias from the prediction provided by the model.[20] That is, if Y is the response variable and Y^* is the response variable actually identified by the model, i.e., the one that corresponds to the estimations provided by the model, then the expected bias is $E[b(Y,Y^*)] = E[Y - Y^*]$, which is estimated as[20]

$$\hat{b} = \frac{1}{m}\sum_{i=1}^{m}(y_i - y_i^*) \ , \tag{8}$$

where y_i and y_i^* are the exact values of the response variable and their estimates in a test database of m records.

Finally, the estimates were corrected by giving $y_i^* - \hat{b}$ as the final estimation for item number i.

We have improved the estimation of the expected bias by detecting homogeneous regions in the set of possible values of Y and then estimating a different expected bias in each region. The domain of the response variable is split using the k-means clustering algorithm, determining k by exploring the dendrogram. In this work we have considered a maximum value of $k = 4$, as we didn't reach any improvement by increasing its value in the experiments carried out.

Therefore, instead of a single estimation of the expected bias $\hat{b}$, now we compute a vector of estimations of the expected bias, $\hat{b}_j$, $j = 1, \ldots, k$, and the final estimation given is $y_i^* - \hat{b}_{j(i)}$, where $j(i)$ denotes the cluster where y_i^* lies in. The procedure for estimating the bias is detailed in algorithm 3.

This new bias estimation heuristic is not really costly, and provides important increases in accuracy. Therefore, we have used it in the experiments reported in Sec. 5.

Database	Size	# Cont.	# Disc.
abalone	4176	8	1
auto-mpg	392	8	0
bodyfat	251	15	0
cloud	107	6	2
concrete	1030	9	0
forestfires	517	11	2
housing	506	14	0
machine	209	8	1
pollution	59	16	0
servo	166	1	4
strikes	624	6	1
veteran	137	4	4
mte50	50	3	1
extended_mte50	50	4	2
tan	500	3	2
extended_tan	500	4	3

Table 1. A description of the databases used in the experiments, indicating their size, number of continuous variables and number of discrete variables.

5. Experimental evaluation

In order to test the performance of the proposed regression models, we have carried out a series of experiments over 16 databases, four of which are artificial (`mte50`, `extended_mte50`, `tan` and `extended_tan`).

The `mte50` dataset[20] consists of a random sample of 50 records drawn from a Bayesian network with naïve 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. 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 follows a Beta distribution with parameters $p = 1/X$ and $q = 1/X$. Out of that database, an NB regression model was constructed using X as response variable, and a sample of size 50 drawn from it using the Elvira software.[7] Database `extended_mte50` was obtained from `mte50` by adding two columns independently of the others. One of the columns was drawn by sampling uniformly from the set $\{0, 1, 2, 3\}$ and the other by sampling from a Gaussian distribution with mean 4 and standard deviation equal to 3.

Database `tan` was constructed in a similar way. We generated a sample of size 1000 for variables $X_0, \ldots, X_4$, where X_0 follows a Gaussian distribution with mean 3 and standard deviation 2, X_1 follows a negative exponential distribution with mean $2 \times |X_0|$, X_2 is uniformly distributed in the interval $(X_0, X_0 + X_1)$, X_3 is sampled from the set $\{0, 1, 2, 3\}$ with probability proportional to X_0 and X_4 follows a Poisson distribution with mean $\lambda = \log(|X_0 - X_1 - X_3| + 1)$. Out of that database, a TAN regression model[8] was generated, and a sample of size 500 drawn from it using the Elvira software.[7] Finally, the dataset `extended_tan` was obtained from `tan` by adding two independent columns, one of them drawn by sampling uniformly from the set $\{0, 1, 2, 3\}$ and the other by sampling from a Gaussian distribution with mean 10 and standard deviation 5.

The aim of using the two extended databases (`extended_mte50` and `extended_tan`) is to test the performance of the variable selection scheme in two databases where we know for sure that some of the explanatory variables do not influence the response variable.

The other databases are available in the UCI[1] and StatLib[26] repositories. A description of the used databases can be found in Tab. 1.

In each database, we produced missing cells by removing values from cells selected at random, the rate of missing values ranging from 10% to 50%. The missing cells have been created in an incremental way, i.e., a database D with 20% of missing cells is constructed from the same database with a 10% of missing values and so on. That is, these two data sets have the same missing cells in a 10% of their positions. Over the resulting databases, we have run 5 algorithms: NB, TAN, SNB and STAN, where the last two correspond to the selective versions of NB

and TAN. We have also included the M5' algorithm in the comparison. The M5' algorithm[28] is an improved version of the model tree introduced by Quinlan.[22] 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. We chose the M5' algorithm because it was the state-of-the-art in graphical models for regression,[11] before the introduction of MTEs for regression.[20] We have used the implementation of that method provided by Weka 3.4.11.[29] Regarding the implementation of our regression models, we have included it in the Elvira software,[7] which can be downloaded from `http://leo.ugr.es/elvira`.

We have used 10-fold cross validation to estimate the srmse. The missing cells in the databases were selected before running the cross validation, therefore, in this case both the training and test databases contain missing cells in each iteration of the cross validation. We discarded from the test set the records for which the value of Y was missing. If the missing cells in the test set correspond to explanatory variables, algorithm M5' imputes them as column average for numeric variables and column mode for qualitative variables.[29] The regression models do not require the imputation of the missing explanatory variables in the test set, as the posterior distribution for Y is computed by probability propagation and therefore, the variables which are not observed are marginalised out. The results of the experimental comparison are displayed in figures 3, 4 and 5. The values represented correspond to the average srmse computed by 10-fold cross validation.

We used Friedman's test[5] to compare the algorithms, reporting statistically significant difference among them, with a p-value of 2.2×10^{-16}. Therefore, we continued the analysis by carrying out a pairwise comparison, following the procedure discussed by García and Herrera,[15] based on Nemenyi's, Holm's, Shaffer's and Bergmann's tests. The ranking of the algorithms analysed, according to Friedman's statistic, is shown in Tab. 2 Notice that a higher rank indicates that the algorithm is more accurate, as we are using the rmse as target. The result of the pairwise comparison is shown in Tab. 3. It can be seen that SNB and STAN outperform their versions without variable selection. Also, M5' is outperformed by SNB and STAN. Finally there are no statistically significant difference between the two most accurate methods: SNB and STAN. The conclusions are rather similar regardless of the test used. The only difference is that Holm's and Bergmann's tests also report significant differences between NB and TAN and between TAN and M5'.

5.1. *Results discussion*

The experimental evaluation shows a satisfactory behaviour of the proposed regression methods. The selective versions outperform the sophisticated M5' algorithm. Notice that the M5' algorithm also incorporates variable selection, through tree-pruning. The difference between the models based on Bayesian networks and model trees becomes sharper as the rate of missing values grows. Also, the use of vari-

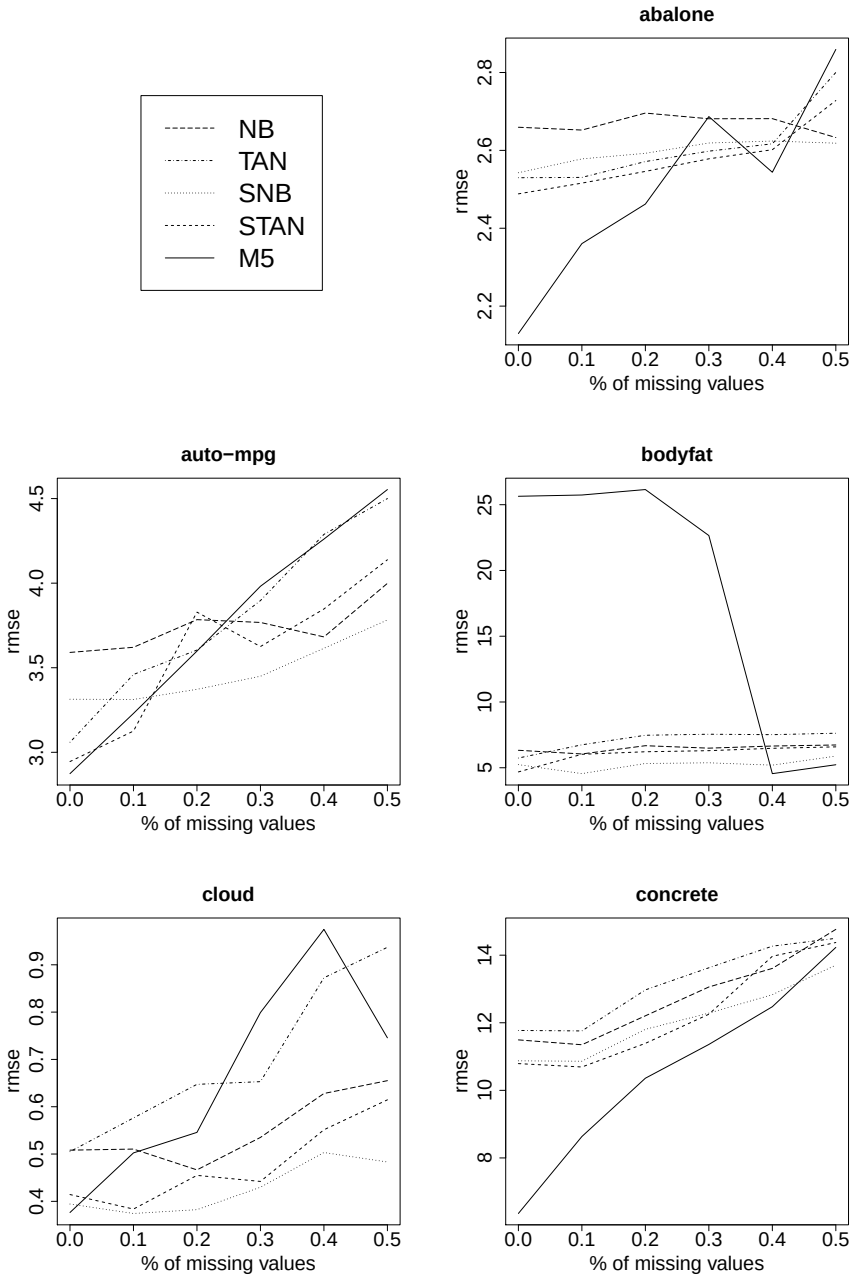

Fig. 3. Comparison of the different models for the data sets.

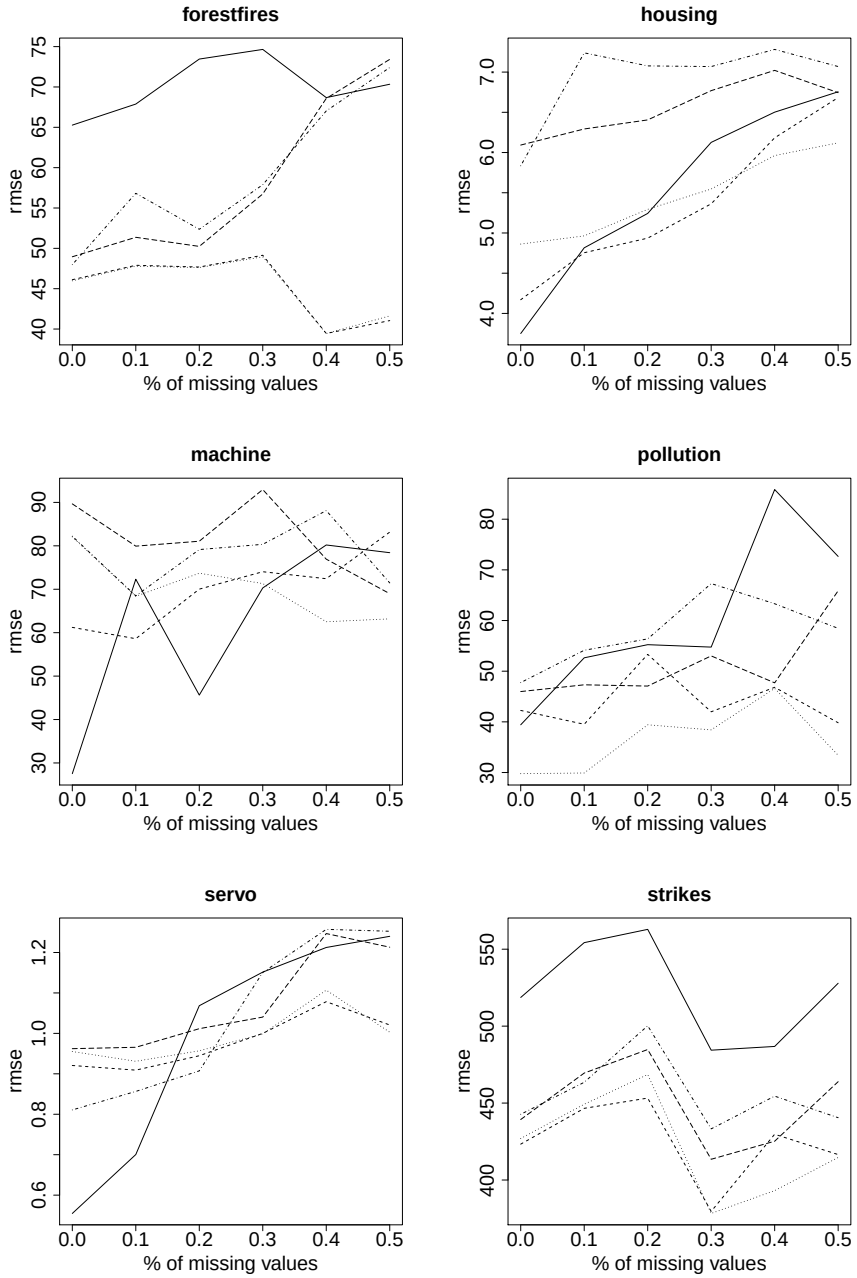

Fig. 4. Comparison of the different models for the data sets. The legends are the same as in figure 3.

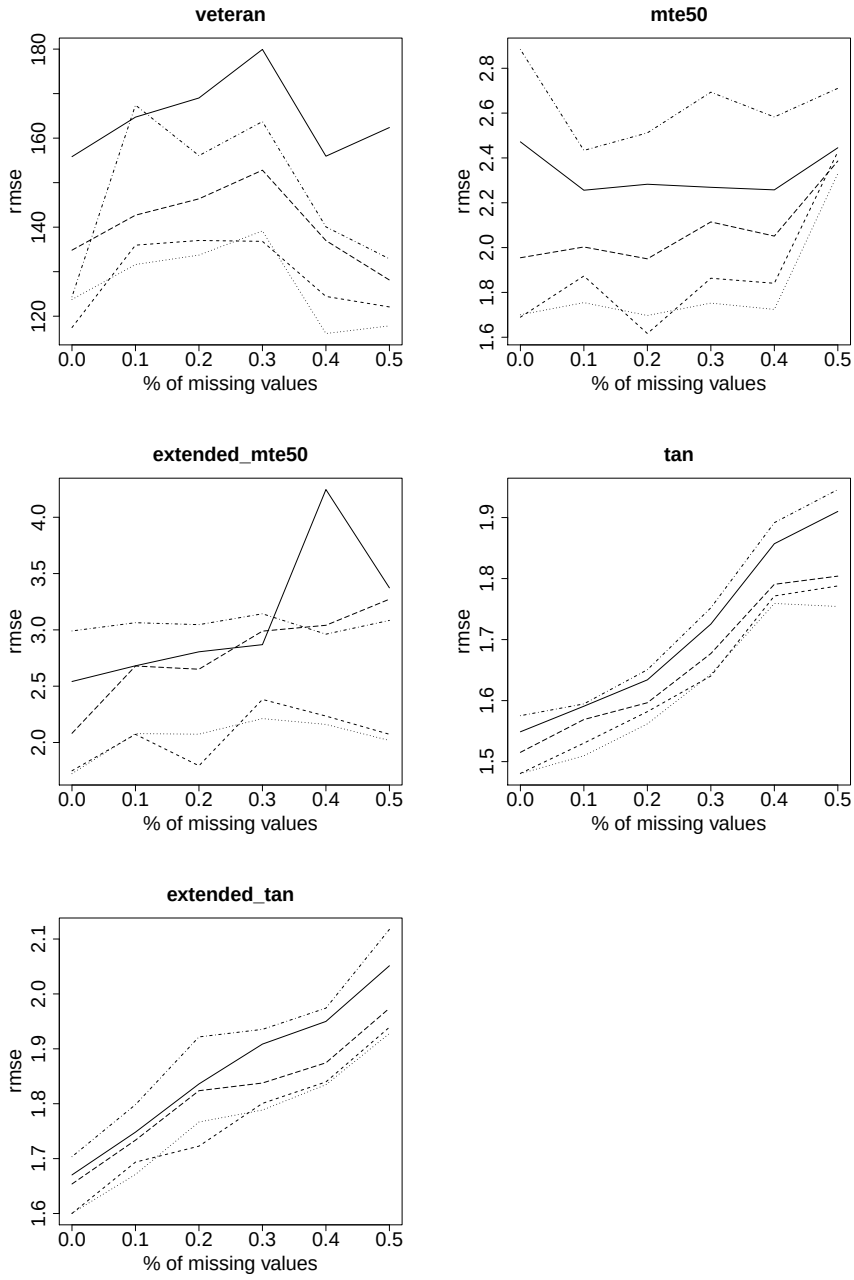

Fig. 5. Comparison of the different models for the data sets. The legends are the same as in figure 3.

Algorithm	Ranking
NB	2.4687500000000004
TAN	1.7916666666666676
SNB	4.302083333333335
STAN	3.9895833333333313
M5'	2.447916666666668

Table 2. Average rankings of the algorithms tested in the experiments using Friedman's test.

Hypothesis	Nemenyi	Holm	Shaffer	Bergmann
TAN vs. SNB	3.8173E-27	3.8173E-27	3.8173E-27	3.8173E-27
TAN vs. STAN	5.9273E-21	5.3346E-21	3.5564E-21	3.5564E-21
SNB vs. M5'	4.4902E-15	3.5922E-15	2.6942E-15	2.6942E-15
NB vs. SNB	9.4913E-15	6.6439E-15	5.6948E-15	3.7965E-15
STAN vs. M5'	1.4259E-10	8.5557E-11	8.5557E-11	4.2778E-11
NB vs. STAN	2.6655E-10	1.3328E-10	1.0662E-10	5.3310E-11
NB vs. TAN	0.0301	0.0120	0.0120	0.0120
TAN vs. M5'	0.0403	0.0121	0.0121	0.0121
SNB vs. STAN	1	0.3418	0.3418	0.3418
NB vs. M5'	1	0.9273	0.9273	0.9273

Table 3. Adjusted p-values for the pairwise comparisons using Nemenyi's, Holm's, Shaffer's and Bergmann's statistical tests.

able selection always increases the accuracy. The fact that there are no significant differences between SNB and STAN make the first one preferable, as it is simpler (contains fewer parameters).

Finally, consider the line corresponding to M5' in the graph for database `bodyfat` in figure 3. In that case, the error decreases abruptly for 40% and 50% of missing values, which is counterintuitive. We have found out that this is due to the presence of outliers in the database, which are removed when the rate of missing values is high. It suggests that M5' is more sensitive to outliers than the models based on Bayesian networks.

6. Conclusions

In this paper we have studied the induction of Bayesian network models for regression from incomplete data sets, based on the use of MTE distributions. We have considered two well known network structures in classification and regression: the naïve Bayes and TAN.

The proposal for handling missing values relies on the data augmentation algorithm, which iteratively re-estimates a model and imputes the missing values using it. We have shown that this algorithm can be adapted for the regression problem

by distinguishing the imputation of the response variable, in such a way that the prediction error is minimised.

We have also studied the problem of variable selection, following the same ideas as in the original NB and TAN models for regression. The final contribution of this paper is the method for improving the accuracy by reducing the bias, which can be incorporated regardless of whether the model is obtained from complete or incomplete data.

The experiments conducted have shown that the selective versions of the proposed algorithms outperform the robust M5' scheme, which is not surprising, as M5' is mainly designed for continuous explanatory variables, while MTEs are naturally developed for hybrid domains.

References

1. C.L. Blake and C.J. Merz. 1998. UCI repository of machine learning databases. www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences.
2. O. Chapelle, B. Shölkopf and A. Zien. "Semi-supervised learning". MIT Press. 2006.
3. B. Cobb and P.P. Shenoy. 2006. Inference in hybrid Bayesian networks with mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 41:257–286.
4. A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1 – 38.
5. J. Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1 – 30.
6. R.O. Duda, P.E. Hart, and D.G. Stork. "Pattern classification". Wiley Interscience, 2001.
7. Elvira Consortium. 2002. Elvira: An environment for creating and using probabilistic graphical models. In J.A. Gámez and A. Salmerón, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 222–230.
8. A. Fernández, M. Morales, and A. Salmerón. 2007. Tree augmented naïve Bayes for regression using mixtures of truncated exponentials: Applications to higher education management. *IDA'07. Lecture Notes in Computer Science*, 4723:59–69.
9. A. Fernández, and A. Salmerón. 2008. Extension of Bayesian network classifiers to regression problems. *IBERAMIA'08. Lecture Notes in Artificial Intelligence*, 5290:83–92.
10. A. Fernández, J. D. Nielsen, and A. Salmerón. 2008. Learning naïve Bayes regression models with missing data using mixtures of truncated exponentials. *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models, PGM'08*, pp. 105–112.
11. E. Frank, L. Trigg, G. Holmes, and I.H. Witten. "Technical note: Naive Bayes for regression". *Machine Learning* **41** (2000) 5–25.
12. N. Friedman, D. Geiger, and M. Goldszmidt. "Bayesian network classifiers". *Machine Learning* **29** (1997) 131–163.
13. Nir Friedman. 1997. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the ICML-97*.
14. J.A. Gámez, R. Rumí, and A. Salmerón. 2006. Unsupervised naïve Bayes for data clustering with mixtures of truncated exponentials. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models (PGM'06)*, pages 123–132.

15. S. García, and F. Herrera. 2008. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694.
16. H. Langseth, T.D. Nielsen, R. Rumí, and A. Salmerón. 2008. Parameter Estimation in Mixtures of Truncated Exponentials. *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models, PGM'08*, pp. 169–176.
17. H. Langseth, T.D. Nielsen, R. Rumí, and A. Salmerón. 2009. Inference in hybrid Bayesian networks. *Reliability Engineering and System Safety*, 94:1499–1509.
18. S. Moral, R. Rumí, and A. Salmerón. 2001. Mixtures of truncated exponentials in hybrid Bayesian networks. *ECSQARU'01. Lecture Notes in Artificial Intelligence*, 2143:135–143.
19. S. Moral, R. Rumí, and A. Salmerón. 2003. Approximating conditional MTE distributions by means of mixed trees. *ECSQARU'03. Lecture Notes in Artificial Intelligence*, 2711:173–183.
20. M. Morales, C. Rodríguez, and A. Salmerón. 2007. Selective naïve Bayes for regression using mixtures of truncated exponentials. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 15:697–716.
21. A. Pérez, P. Larrañaga, and I. Inza. "Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naïve Bayes". *International Journal of Approximate Reasoning* **43** (2006) 1–25.
22. J.R. Quinlan. 1992. Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore. World Scientific.
23. R. Ruiz, J. Riquelme and J.S. Aguilar-Ruiz. "Incremental wrapper-based gene selection from microarray data for cancer classification". *Pattern Recognition* **39** (2006) 2383–2392.
24. R. Rumí and A. Salmerón. 2007. Approximate probability propagation with mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 45:191–210.
25. R. Rumí, A. Salmerón, and S. Moral. 2006. Estimating mixtures of truncated exponentials in hybrid Bayesian networks. *Test*, 15:397–421.
26. StatLib. 1999. www.statlib.org. Department of Statistics. Carnegie Mellon University.
27. M.A. Tanner and W.H Wong. 1987. The calculation of posterior distributions by data augmentation (with discussion). *Journal of the American Statistical Association*, 82:528–550.
28. Y. Wang and I.H. Witten. 1997. Induction of model trees for predicting continuous cases. In *Proceedings of the Poster Papers of the European Conference on Machine Learning*, pages 128–137.
29. I.H. Witten and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann.