Mixtures of Polynomials for Regression Problems

J. Carlos Luengo, and Rafael Rumi

***Abstract*— This paper presents a novel methodology for solving regression problems, based on Bayesian networks. Graphical models have been used mostly for solving classification problem, but they can be extended to solve regression problems by allowing continuous variables in the model. Gaussian models were first proposed for dealing with continuous variables in Bayesian networks, but they have some limitations. The Mixtures of Polynomials have recently been proposed as a valid alternative to the Gaussian model, but it has not yet been tested in regression models. For this, a model is developed within a restricted Bayesian network (Naive Bayes), and the parameters involved analyzed. Two preprocessing steps are presented to improve the results, and the procedures for learning the corresponding distributions shown. The results are compared with some state-of-the-art models, obtaining promising results.**

I. Introduction and related work

Regression problems are one of the main problems in data mining. They are present in all science fields, dealing with many real-world problems, and therefore they have attracted much attention from the statistics and data mining community. A regression problem can be seen as a supervised classification problem, in which the goal variable is continuous or numeric. There is a great number of methods to solve regression problems; one of them are graphical models, and Bayesian networks in particular, which have recently raised as valid alternatives for classification problems.

Bayesian networks are known to be efficient tools for probabilistic reasoning, that can be applied also to classification problems, both supervised and unsupervised, due to their flexibility. In the case of supervised classification, if the goal variable is continuous, we are facing a regression problem, that most times is solved by a fixed structure model, in such a way that the number of parameters to estimate is minimized.

In order to incorporate continuous variables in these regression models, a structure compatible with discrete and continuous variables is presented, the Mixtures of Polynomials (MoPs), which have been recently developed in Bayesian networks [1], but not applied to regression problems.

The main goal of this paper is to show that MoPs models are a competitive model for regression problems, in comparison to well known state-of-the-art methods, and to analyze the impact on the results of the different parameters of the model.

A Bayesian network is a decomposition of a joint probability distribution in products of conditional distributions. It can be expressed by a directed acyclic graph, in which every node in the graph represents a random variable in the problem, and the presence of an edge linking two nodes represents a statistical relation between them. Associated to each node in the graph there is a probability distribution of the corresponding variable, given its parents in the graph. In the case of root nodes (without parents in the graph) this distribution reduces to a marginal distribution.

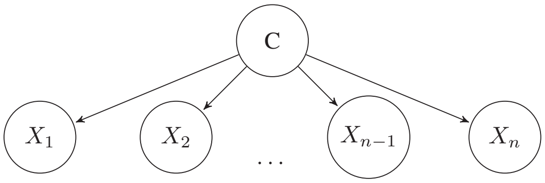

Fig. 1. Structure of Naive Bayes classifier, in which C is the goal variable, and the rest are the feature variables.

Let $X_1, X_2, \dots, X_n$ be the random variables defining our problem, then the joint probability distribution of the variables reduces then to the following decomposition, because of the independence relation encoded in the network [2]:

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} p(x_i | pa(x_i))$$

where $p(x_i)$ represents the parents (variables) of X_i in the graph.

When the main purpose of the model is to predict a goal variable, *i. e.* a classification problem, it is usual to employ a restricted Bayesian network, where the restriction is placed on the structure of the network. Examples of these models are kDB [3], Tree Augmented Network (TAN) [4] , Forest Augmented Network (FAN) [5], and the simplest model, the Naive Bayes (NB) model, in which all the features are assumed to be independent given the class. In the NB model, the class variable is the only root variable, which is linked to every feature variable in the model. An example of a NB model can be seen in Fig 1.

As simple as it may seem, NB is known to be an accurate classifier with a relatively small computational complexity. In this kind of models, the focus is placed on maximizing the accuracy of the prediction, rather than in estimating accurately the model parameters [6]. The rest of the restricted-models mentioned before increase the complexity of the model, relaxing the independence assumption of the NB, and so including more links in the graph.

A. Continuous variables

Bayesian network models were developed originally for discrete, *i. e.* categorical, variables; however in most of real-

J. Carlos Luengo is with the Alyanub IES, Department of Mathematics, Almeria, Calle Mayor, 58, 04630, Vera, Almeria, Spain (email: jcluengolopez@gmail.com).

Rafael Rumi is with the Department of Mathematics, Almeria University, 04120, Almeria, Spain (email:rrumi@ual.es).

world applications there exist continuous data. Practitioners and researchers usually solve this issue by discretizing the continuous variables and transforming them into discrete, by means of some common techniques, such as *equal frequency*, *equal width*, or *k-means* [7], or even some context-specific methods such as dynamic discretization [8] or classification-oriented methods [9]. All these solutions conveys a loss in information, and should be avoided when dealing with a regression problem, in which the goal variable is continuous, and so discretizing would intrinsically change the nature of the problem.

Therefore, methods able to include continuous variables in BNs are needed, and so, the Conditional Gaussian Model (CLG) [10] was conceived to overcome this problem, by defining the continuous variables as Gaussians. Even though this model is still widely used, it has two main disadvantages, the distribution of the continuous variables must be Gaussian, and discrete nodes cannot have continuous variables as parents. In particular, in the case that we use a NB model for regression, if any of the feature variables is discrete, the CLG model cannot be used because of this second constraint.

Following this idea appears the Mixtures of Truncated Exponentials model (MTE) [11], in which any probability distribution can be approximated and there is no restriction in the topology of the network. This model has been successfully applied in recent years to classification and regression problems [12], [13] and in general to hybrid BNs [14], [15]. The Mixtures of Polynomials model [1], [16] is similar, in the sense that it can also approximate any probability distribution, but uses polynomials as the basis functions, gaining fitting power just increasing the degree.

Therefore, in this paper we propose the use of MoPs models within Bayesian networks to deal with regression problems, and compare it with state-of-the-art methods. Section II-B presents the details of the MoP model used in this paper to represent the continuous variables.

There is not much literature on regression using mixed BNs. Previous works containing some connections with this current paper deal only with estimating MoPs distributions or using MTEs for regression problems.

MoPs have mostly been applied to approximate specific known distributions, as in [19], but not to include them as a tool to approximate any dataset, neither to use them for regression. Therefore, the parameter estimation procedure is based on minimizing the Mean Squared Error (MSE) between the origina data and the fitted distribution, rather than improving accuracy of forecasts. There are some recent publications that deal with this problem using different properties of the polynomials [22], [25], based on B-splines approximations and maximum likelihood, instead of focusing in accuracy indicators. MoPs have also recently been shown [25], [18] to be a valid alternative when facing a classification problem, using both a NB and a TAN structure, however they have not been applied yet to regression problems, even though the results obtained for classification problems were satisfactory.

MTEs have been used to solve regression problems following a similar general structure to this paper, but maintaining strong differences in terms of estimation of the parameters of the model, based only on least squares approaches (for more information, see [13]).

The rest of the paper is structured as follows: Section II presents the details of the methodology proposed, in Section III an exhaustive set of experiments are carried out, and the paper ends with some conclusions.

II. Proposed Methodologies

A. Preprocessing steps

Before carrying out the main procedure of learning the model, some preprocessing task are executed, in order to simplify or enhance the results.

1) Feature selection: Selecting the appropriate variables in a classification model is a crucial step for successfully solve the problem, since it avoids overfitting and noise in the model [17]. We have selected 3 *filter-wrapper* strategies, based on the mutual information between the goal variable and each feature. Although this procedure is included as a preprocessing step, it is actually embedded into the learning algorithm, not a as different step.

The computation of the mutual information is expressed in the following formula

$$MI(X;C) = \sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i,c_j) log \frac{p(x_i,c_j)}{p(x_i)p(c_j)}$$

obtained from the discretized database. We have chosen to discretize due to the ease and speed of this procedure, because this is just an initial step, that takes into account the information shared between the two variables, and is later updated in terms of accuracy of the prediction, which is more suitable to the current problem.

Once the mutual information has been obtained, the variables are ranked, and they are included (or not) in the model, according to three different heuristics:

- Type 1: Classical filter-wraper approach: Variables in the ranking are inserted in the model, as long as they improve the results, by decreasing the MSE (see Section III). The first time a variable does not improve the results, we stop.
- Type 2: Variables are ranked, and the three first variables not yet inserted in the model are considered. We create new model inserting the first one, if the error decreases, it is finally inserted. If not, we try with the second one. If it decreases the error, it is finally inserted. If not, we try again with the third one. Only if none of them improves the result, we stop. Once a variable is included, a new subset of the three best-ranked variables is considered.
- Type 3: Equivalent to Type 2, but we allow some increase of the MSE, only three times. The first one we allow a 10% increase, the second one a 8% decrease, and the third one a 5% decrease. This method resembles to the *Simulated Annealing*.

2) Discretization: Detecting when a variable is discrete or continuous is easy when looking at the definition, however, when dealing with real data, this difference is not straight. Some variables cannot be considered discrete neither continuous because of the great number of levles, or the difficulty for fitting a polynomial. Therefore, we have considered a special type of variable, *pseudo-continuous* variables, defined in [18], found mainly in large datasets, to treat those variables continuous by definition, but not in practice.

These *Pseudo-continuous* variables are defined as numeric features which have more then 20 different values, but less than the 5% of the total number of observations of the dataset.

In the *Experiments Results* section we will investigate if detecting and discretizing these variables yields in an accuracy-gain, as it was observed for classification problems in [18].

B. Learning the model

Learning a NB model from data using MoPs to represent the continuous variables reduces to estimating the corresponding parameters. According to Fig. 1, these are : 1) a marginal distribution for the goal variable, $f(c)$ 2) a conditional distribution for each feature variable, given the goal variable, $f(x_i|c)$. These distributions are represented within the BN by means of the MoP model.

1) Mixtures of Polynomials: The MoP framework is able to approximate any continuous distribution by a piecewise function which has in each piece a polynomial function. This allows to work directly with the continuous variables without the need of discretizing them. A MoP function is defined as follows [1]:

Definition 1: A one-dimensional function $f : \mathbb{R} \rightarrow \mathbb{R}$ is said to be a mixture of polynomials function if it is a piecewise function of the form

$$f(x) = \begin{cases} a_{0i} + a_{1i}x + \cdots + a_{ni}x^n & x \in A_i \quad , i = 1, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

where $A_1, \ldots, A_k$ are disjoint intervals in $\mathbb{R}$ and $a_{0i}, \ldots, a_{ni}$ are real constants for all i.

There are some variants to this definition, in [16] Shenoy *et al* re-defines the MoP function to include multivariate functions in which $A_1, \ldots, A_k$ may be also hyper-rhombuses. This re-definition makes MoP functions richer because now they can deal with deterministic functions, however it makes computations extremely complex [19], [20]. We do not use this variant because, in the NB model for regression, we need a tradeoff between accuracy and complexity.

The main reason for using MoPs is that they have a great fitting power, as we will see later in Section II-B.3. Note that this model is valid for hybrid datasets (in which discrete and continuous variables co-exist), since operations with discrete variables can be handled easily.

There is a close relation between MoPs and MTEs. In fact, they both arise from the same model, the Mixtures of Truncated Basis Functions (MoTBFs) [21], however the basis functions they use to represent the continuous distribution is different; MTEs uses exponentials functions, and MoPs uses polynomials. Even though they have a similar performance, it seems that MoPs have a greater accuracy when estimating probability distributions [22], however there is not a comparison between these two models for regression, in terms of prediction accuracy. This paper tries to address this issue.

2) Marginal densities: The estimation of the marginal continuos densities is carried out using the procedure in [18], which can be summarized as:

1) From the sample, obtain (x, y) where y is estimated through a kernel density estimator [23].
2) Estimate the parameters of the 1- degree polynomial $(p_1(x))$ by minimizing the MSE.
3) $p_i(x)$ may be negative in some points of the domain. Select only the most important positive part, in terms of weight or size
4) Normalize the polynomial so that its integral is equal to the proportion of points of the sample included in the selected subinterval.
5) If a part of the domain of $p_i(x)$ has been removed because of the negative values, add the necessary tails to extend the domain of the polynomial to include the whole domain of X , which yields in a piecewise function
6) Compute the corresponding MSE
7) Increase the degree
8) Repeat this procedure, and select the best polynomial.

The estimation of marginal densities is carried out for the root (goal) variable, and within the procedure for estimating conditional densities.

3) Conditional densities: The definition of the probability distributions for the features variables includes a conditional density, $f(x|c)$. If C were discrete, this conditional density would reduce to defining a different density for X for every state of the variable C. However, in the case of a continuous variable C, this conditional distribution is not so straightforward. In [24] it was shown that the only way to include C in the explicit formula of the conditional distribution for MTEs is in the definition if the domain; since MoPs belong to the same model MoTBFs than MTEs, we followed the same procedure, *i. e.*, the problem transforms to find an optimal discretization of the range of C, and for each interval in the discretization, estimate a marginal density for X.

The discretization procedure followed is an iterative procedure based on the *equal width* method:

. INPUT: Sample for X and C.
. OUTPUT: $f(x|c)$ a conditional density function.

1) Estimate $f_0(x)$ a marginal density function for X. and compute an estimation of the Mean Squared Error (MSE) of this model (see section III)..
2) Split the range of C, Ω_C in two equal width intervals.
3) In each one of the new intervals estimate $f_{11}(x)$ and f_{12} marginal density functions for X. and compute an

estimation of the MSE of this new model.

4) If the error decreases, go back to 1), try dividing every possible interval and effectively split the one with lower MSE; if not, stop and return the corresponding piecewise function.
5) Continue until the error does not decrease dividing any of the intervals, or the maximum number of intervals is reached.

This procedure is specifically designed for regression problems, since the final decision about splitting or not the domain is made according to the accuracy of the prediction in a test dataset.

III. Forecasting Results

The aim of these models is to predict as accurately as possible a goal variable.

Once the model is learnt, it is able to *predict* the goal value (c) of a given observation $\mathbf{x} = (x_1, \dots, x_n)$ using the updated probability distribution for C :

$$f(c|\mathbf{x}) = f(c|x_1, \dots, x_n) \propto f(c) \prod_{j=1}^{n} f(x_j|c)$$

This computation outputs a piecewise density for C. From this distribution, the point estimation of prediction of the variable is computed as the *Expected value*:

$$\hat{c} = E[C|x_1, \dots, x_n] = \int_{-\infty}^{\infty} c f(c|x_1, \dots, x_n) dc$$

where $x_1, \dots, x_n$ are known values, called *evidence*.

The procedures described in the previous sections have been implemented in *R* [26], and validated through an exhaustive set of experiments. A total of 10 different datasets have been selected for the experiments from the UCI Machine Learning Repository [27] and the KEEL repository [28]. They all have both discrete and continuous features, with a wide range of number of cases (from 60 to 1030), and features (from 4 to 15). In the case of missing values, the corresponding cases were removed from the dataset.

TABLE I
Databases used in experiments.

Dataset	Vars	Instances
Auto	8	392
BodyFat	15	252
Cloud	8	108
Concrete	9	1030
Housing	14	506
Machine	9	209
Pollution	16	60
Servo	5	167
Strikes	7	625
Veteran	8	137

In the next sections, several hypothesis are stated, related to the performance of the MoP model in regression, and the validity of the different parameters and preprocessing methods discussed above. The error of the different methods is computed in terms of the Mean Squared Error, computed as

$$error = MSE = \frac{\sum_{i=1}^{n} (\hat{c} - c)^2}{n}$$

where n is the size of the test sample. The validation of the methods was carried out by menas of a 5-fold Cross-Validation.

A. Feature Selection method

The first issue to check is whether the feature selection scheme yields better results than including in the model every variable of the problem, and, in the case of rejection, which of the three feature selection methods obtains better results. In Table II we can see the accuracy results of the different feature selection methods, in which the goal variable range was divided at most in 3 intervals to estimate the conditional distributions.

TABLE II
Comparison results for the different feature selection methods. Boldfaced number represent the most accurate result

Method	Auto	BodyFat	Cloud	Concrete	Housing
No	19.739	0.6566	0.6877	50964.75	35.674
Type 1	23.009	0.4012	0.5134	49618.85	27.4705
Type 2	**18.133**	**0.3897**	**0.4641**	**46683.13**	22.0975
Type 3	18.432	0.4032	0.4692	46734.49	**18.35**
Method	Machine	Pollution	Servo	Strikes	Veteran
No	6606.80	2743.18	0.8172	1578125	27613.66
Type 1	7635.33	2321.96	0.9937	324383.9	23980.95
Type 2	**3633.62**	2219.48	0.8048	**305999.5**	**23020.05**
Type 3	3823.53	**2087.88**	**0.8034**	323752.8	23662.24

This information can be graphically seen in Fig. 2, in which the *relative errors* are plotted, for a better visualization of the graph.

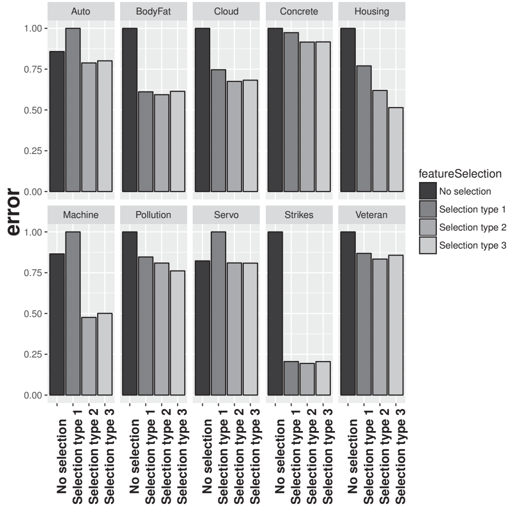

Fig. 2. Graphical representation of the prediction accuracy for the different feature selection methods

The Friedman statistical test was carried out to test if all the feature selection methods are equivalent, obtaining a p-value of $3.73e-05$, so not all the methods have similar results, in terms of accuracy. A posterior post-hoc test shows that Feature selection types 2 and 3 can be considered equivalent, but different to type 1 and no feature selection.

A box plot graph of the ranking of the different methods can be seen on Fig. 3.

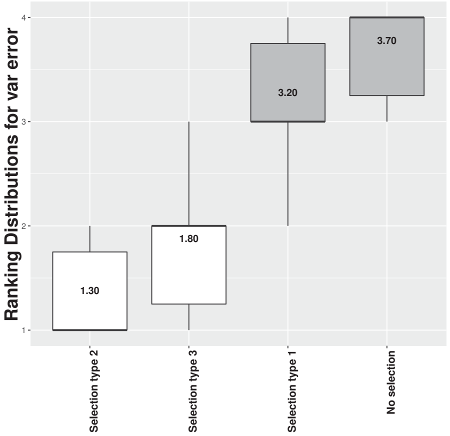

Fig. 3. Boxplot representing the ranking distribution for each feature selection method. In white color those methods obtaining better results

The conclusion we may draw from the test and the different plots is that the Feature selection method 2 is the one yielding better results.

B. Discretization method (pseudo continuous variables)

In section II-A.2 a new type of variable was defined, the pseudo continuous variable. It was shown in [18] that detecting and discretizing them was successful for discretization problems. Does it have the same impact in regression problems? To solve this issue, three new datasets were selected, since none of the ones summarized in Table I included such type of variables. The new datasets are selected from the ones in [18], which are focused on classification, however we selected a continuous variable as the goal variable, and transform the problem into a regression problem. In Table III these datasets are listed

TABLE III
DATABASES USED IN THE PSEUDO CONTINUOUS EXPERIMENTS

Dataset	Vars	Instances
Australian	15	690
Credit	16	653
German	25	1000

We can see the accuracy results of the different feature selection methods in Table IV.

This information can be graphically seen in Fig. 4, in which the *relative errors* are plotted, for a better visualization of the graph.

The Wilcoxon Signed Rank statistical test was carried out to test if detecting and discretizing pseudo continuous variables makes a difference, obtaining a p-value of 0.375, so there is not a statistical difference in terms of accuracy.

TABLE IV
COMPARISON OF RESULTS WHEN DETECTING PSEUDO-CONTINUOUS VARIABLES. BOLDFACED NUMBER REPRESENT THE MOST ACCURATE RESULT

Detect Pseudo continuous	Australian	Credit	German
No	104.7557	6327.3470	**420.2247**
Yes	**103.2565**	**6129.6650**	454.9360

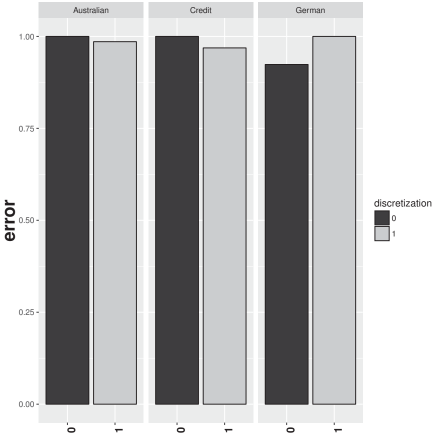

Fig. 4. Graphical representation of the prediction accuracy for when detecting and discretizing pseudo continuos variables

However, looking at Table IV and Fig. 4 we can argue that having pseudo continuous variables into account can benefit our results.

C. Number of intervals in conditional learning

One of the parameters that may affect most the estimation of conditional MoPs is the maximum number of intervals to split the domain of the goal variable into. In order to detect if it influences the result of the prediction, a new experiment was carried out to observe this issue, in which the Feature selection algorithm Type 2 was chosen.

In Table V we can see the accuracy results of selecting 3 or 5 intervals as maximum.

TABLE V
COMPARISON OF RESULTS FOR THE DIFFERENT INTERVALS. BOLDFACED NUMBER REPRESENT THE MOST ACCURATE RESULT

Intervals	Auto	BodyFat	Cloud	Concrete	Housing
3	18.13	0.3897	0.464	46683.13	**22.0975**
5	**17.21**	**0.368**	**0.4272**	**45930.91**	22.34
Intervals	Machine	Pollution	Servo	Strikes	Veteran
3	3633.62	2219.482	0.8048	305999.5	23020.05
5	**3486.17**	**2146.728**	**0.7148**	**292634.3**	**22811.33**

This information can be graphically seen in Fig. 5, in which the *relative errors* are plotted, for a better visualization of the graph.

The Wilcoxon Signed Rank statistical test was carried out to test if detecting and discretizing pseudo continuous

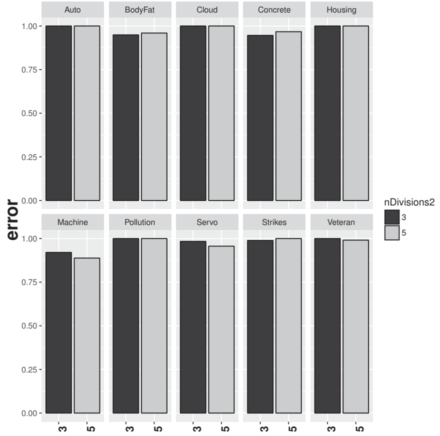

Fig. 5. Graphical representation of the prediction accuracy for different maximum number of intervals when splitting the range of the goal variable

variables makes a difference, obtaining a p-value of 0.0068, so there is a statistical difference in terms of accuracy, obtaining better results for 5 intervals than for 3 intervals. This supports the idea that, the more intervals used, the better the accuracy obtained; however the complexity of the model increases.

D. Global performance of MoP Naive Bayes model

Once the parameters and preprocessing methods affecting the MoP model estimation are checked, a global comparison with respect to some state-of-the-art methods is carried out. We have selected three methods: The MTE method within a Naive Bayes structure and a feature selection method defined in [13], as a natural competitor, the linear model, and the regression trees included in the CART model [29], both implemented in R. The MoPs model for this comparison includes feature selection type 2, and 5 intervals to split the domain of the goal variable when learning conditional distributions.

In Table VI we can see the accuracy results of selecting the different methods, including all dataset available:

TABLE VI

COMPARISON OF RESULTS FOR THE DIFFERENT METHODS. BOLDFACED NUMBER REPRESENT THE MOST ACCURATE RESULT

model	Australian	Auto	BodyFat	Cloud	
lm	118.565	**14.91**	**0.3256**	**0.22**	
MOPs	**100.01**	17.21	0.3687	0.42	
MTEs	112.36	18.82	0.411	0.84	
regTree	113.98	22.78	0.47	0.89	
model	Concrete	Credit	German	Housing	
lm	61047.23	-	1829.274	37.03	
MOPs	**45930.91**	**6078.65**	**420.224**	**18.35**	
MTEs	53610	68	502.65	40.63	
regTree	58079.17	7837.24	1827.66	35.519	
model	Machine	Pollution	Servo	Strikes	Veteran
lm	40000	2070.66	1.24	302423.5	26752.3
MOPs	**3486.17**	2087.88	0.71	292634.3	22811.3
MTEs	9820.37	**1646.33**	1.2975	**260324.4**	**15237.4**
regTree	11683.38	2300.52	**0.64**	350885.7	35181.9

This information can be graphically seen in Fig. 6, in which the *relative errors* are plotted, for a better visualization of the graph.

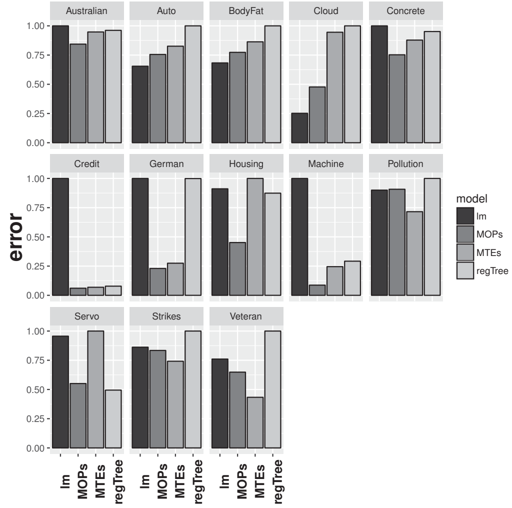

Fig. 6. Graphical representation of the prediction accuracy for the different feature selection methods

The Friedman statistical test was carried out to test if all the methods mentioned above are equivalent, in terms of accuracy, obtaining a p-value of 0.0093, so not all the methods have similar results. A posterior post-hoc test shows that all the methods have similar performance, except for regression trees, which is statistically different to MoPs model.

A box plot graph of the ranking of the different methods can be seen on Fig. 7. Even though MoPs model is not always the best model, it is clearly competitive with respect to the state-of-the-art, since there is no significative difference between their results, apart from regression trees.

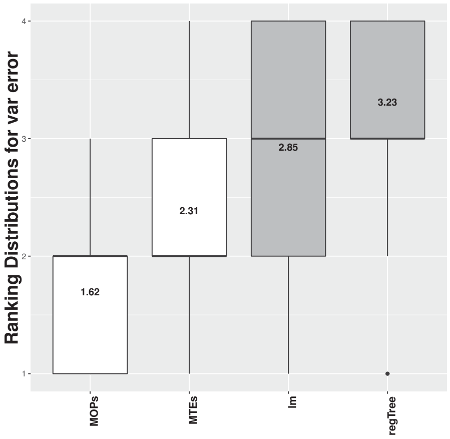

Fig. 7. Boxplot representing the ranking distribution for each feature selection method. In white color those methods obtaining better results

IV. Conclusions

We have developed a methodology for incorporating MoPs in a Bayesian network framework to solve regression models. The different parameters and preprocessing steps have been presented, as well as the learning procedures, to estimate marginal and conditional densities. Some experiments have been carried out, to look for the optimal configuration of the parameters, and the resulting method has been tested against several state-of-the-art methods, obtaining competitive results. This means that MoPs model are a valid alternative when solving a regression problem.

Some new issues can be developed in future papers, such as incorporating nee constrained structures, TAN and FAN for example, and studying the effect of this methodology in the size of the models obtained. Preliminary experiments on this issue report that MoPs models achieve similar accuracy, but incorporate a lower number of feature variables.

Acknowledgments

This work has been supported by the Spanish Ministry of Science and Innovation, through project TIN2013-46638-C3-1-P, by Junta de Andalucia through project P12-TIC-2541 and by ERDF (FEDER) funds.

References

[1] P. P. Shenoy and J. West, "Inference in hybrid Bayesian networks using mixtures of polynomials," *International Journal of Approximate Reasoning*, vol. 52, pp. 641–657, 2011.

[2] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*. Springer, 2007.

[3] M. Sahami, "Learning limited dependence Bayesian classifiers," in *KDD96: Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, pp. 335–338, 1996.

[4] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.

[5] P. J. Lucas, "Restricted Bayesian network structure learning," in *Proceedings of the 1st European Workshop on Probabilistic GraphicalModels (PGM'02)* (J. Gámez and A. Salmerón, eds.), pp. 117–126, 2002.

[6] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple bayesian classifier," in *Proceedings of the International Conference on Machine Learning*, 1996.

[7] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Machine Learning: Proceedings of the Twelfth International Conference* (A. P. y S. Russell, ed.), pp. 194–202, Morgan Kaufmann, San Francisco, 1995.

[8] D. Kozlov and D. Koller, "Nonuniform dynamic discretization in hybrid networks," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence* (D. Geiger and P. Shenoy, eds.), pp. 302–313, Morgan & Kaufmann, 1997.

[9] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp. 1022 – 1027, 1993.

[10] S. Lauritzen and N. Wermuth, "Graphical models for associations between variables, some of which are qualitative and some quantitative," *The Annals of Statistics*, vol. 17, pp. 31–57, 1989.

[11] S. Moral, R. Rumí, and A. Salmerón, "Mixtures of Truncated Exponentials in Hybrid Bayesian Networks," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (S. Benferhat and P. Besnard, eds.), vol. 2143 of *Lecture Notes in Artificial Intelligence*, pp. 156–167, Springer, 2001.

[12] A. Fernández and A. Salmerón, "Extension of Bayesian network classifiers to regression problems," in *Advances in Artificial Intelligence - IBERAMIA 2008* (H. Geffner, R. Prada, I. M. Alexandre, and N. David, eds.), vol. 5290 of *Lecture Notes in Artificial Intelligence*, pp. 83–92, Springer, 2008.

[13] M. Morales, C. Rodríguez, and A. Salmerón, "Selective naïve Bayes for regression using mixtures of truncated exponentials," *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, vol. 15, pp. 697–716, 2007.

[14] R. Rumí, A. Salmerón, and S. Moral, "Estimating mixtures of truncated exponentials in hybrid Bayesian network," *Test*, vol. 15, pp. 397–421, 2006.

[15] V. Romero, R. Rumí, and A. Salmerón, "Learning hybrid Bayesian networks using mixtures of truncated exponentials," *International Journal of Approximate Reasoning*, vol. 42, pp. 54–68, 2006.

[16] P. P. Shenoy, "A re-definition of mixtures of polynomials for inference in hybrid Bayesian networks," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (W. Liu, ed.), Lecture Notes in Artificial Intelligence 6717, pp. 98–109, Springer, 2011.

[17] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131 – 156, 1997.

[18] J. C. Luengo and R. Rumí, "Naive bayes classifier with mixtured of polynomials," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM-2015)*, pp. 14–24, 2015.

[19] P. P. Shenoy, R. Rumí, and A. Salmerón, "Some practical issues in inference in hybrid bayesian networks with deterministic conditionals," in *Proceedings of the Intelligent Systems Design and Applications (ISDA)*, 2011.

[20] R. Rumí, A. Salmerón, and P. P. Shenoy, "Tractable inference in hybrid bayesian networks with deterministic conditionals using reapproximations," in *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM'2012)*, pp. 275 – 282, 2012.

[21] H. Langseth, T. D. Nielsen, R. Rumí, and A. Salmerón, "Mixtures of truncated basis functions," *International Journal of Approximate Reasoning*, vol. 53, pp. 212 – 227, 2012.

[22] H. Langseth, T. D. Nielsen, I. Pérez-Bernabé, and A. Salmerón, "Learning mixtures of truncated basis functions from data.," *International Journal of Approximate Reasoning*, 2013.

[23] J. Simonoff, *Smoothing methods in Statistics*. Springer, 1996.

[24] H. Langseth, T. D. Nielsen, R. Rumí, and A. Salmerón, "Parameter estimation and model selection for mixtures of truncated exponentials," *International Journal of Approximate Reasoning*, vol. 51, no. 5, pp. 485–498, 2010.

[25] P. L. López-Cruz, C. Bielza, and P. Larrañaga, "Learning mixtures of polynomials of multidimensional probability densities from data using b-spline interpolation," *International Journal of Approximate Reasoning*, vol. In Press, 2013.

[26] R Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[27] K. Bache and M. Lichman, "UCI machine learning repository." http://archive.ics.uci.edu/ml, 2013.

[28] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255–287, 2011.

[29] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Chapman & Hall/CRC, 1984.