Pattern Recognition Letters 98 (2017) 39–45

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis

Kyoungok Kim, Jung-sik Hong*

Information Technology Management Programme, International Fusion School, Seoul National University of Science & Technology (SeoulTech), 232 Gongreungno, Nowon-gu, Seoul, 139-743, Republic of Korea

ARTICLE INFO

Article history:
Received 30 August 2016
Available online 10 August 2017

Keywords:
Mixed data
Regression
Hybrid model
Decision tree

ABSTRACT

In many real world problems, the collected data are not always numeric; rather, the data can include categorical variables. Inclusion of different types of variables may lead to complications in regression analysis. Many regression algorithms such as linear regression, support vector regression, and neural networks that train parameters of a model to identify relations between input and output variables, can easily process numeric variables; however, there are additional considerations for categorical variables. On the other hand, a decision tree algorithm estimates a target based on the specified rules; therefore, it can support categorical variables as well as numeric variables. Using this property, a new hybrid model combining a decision tree with another regression algorithm is proposed to analyze mixed data. In the proposed model, the portions explained by categorical variables in target values are estimated by the decision tree and the remaining parts are predicted by any regression algorithm trained by numerical variables. The proposed algorithm was evaluated using 12 datasets selected from real decision problems, and it was confirmed that the proposed algorithm achieved better or comparable accuracy than the comparison methods including the M5 decision tree and the evolutionary tree. In addition, the new hybrid method does not significantly increase computational complexity, even though it builds two separate models, which is an advantage that is in contrast with the M5 decision tree and the evolutionary tree.

1. Introduction

One of the primary tasks in data mining is regression analysis, the objective of which is to identify the relationships between a dependent variable and one or more independent variables and it has been used widely for prediction and forecasting in many real-world problems. In many cases, datasets consist of both numerical and categorical independent variables. Many regression algorithms such as linear regression, support vector regression, and neural networks are well-defined and validated in the support of the computation of numeric variables, because it is easy to model the relations between a target and its predictors when both data types are numeric. In contrast, categorical variables describe non-numeric, qualitative attributes of data and it is not possible to perform numeric operations on categorical variables.

One possible conversion of categorical variable data to numeric data is the use of various coding systems such as dummy coding, effects coding, and contrast coding to manage the presence of categorical predictors [8,16]. Through these coding systems, qualitative variables representing categories or group memberships are converted to quantitative variables. Another approach is to define similarity or dissimilarity measures between categorical and numeric variables. Distance-based regression algorithms have utilized this approach by defining appropriate distance metrics [5,10,11]. When dissimilar or similar functions are defined on both categorical and numeric variables, then an original input data matrix can be transformed into a distance configuration matrix. Any distance-based regression algorithm can be applied to the converted matrix. In addition, distance metrics are used to extract neighbor samples for *k*-nearest neighbor regression and non-parametric kernel regression.

These two approaches, however, have limitations. Coding systems can cause a significant increase in the number of predictors when the number of categorical variables or the number of categories in the categorical variable is large. For this reason, coding systems include additional steps in order to reduce the number of variables [23,36]. Additionally, characterizing suitable similarity measures between categorical variables or between categorical and numeric variables constitutes a complex challenge.

To address the complexity, we propose a hybrid regression model constructed using a decision tree algorithm. In the proposed model, categorical variables are used to train a decision tree, and

* Corresponding author.
E-mail address: hong@seoultech.ac.kr (J.-s. Hong).

http://dx.doi.org/10.1016/j.patrec.2017.08.011
0167-8655/

then the portions that are not explained by the decision tree are predicted from numeric variables using one of other regression algorithms. We compared the performance of the proposed model with that of the conventional coding approach in five different regression algorithms.

The proposed model has properties in common with M5, a hybrid decision tree model which combines decision tree and regression algorithms [27]. M5 is an improved regression tree algorithm, which splits the parameter space into subspaces and builds local linear regression models [27]. There are two differences between the proposed model and M5. First, the proposed model utilizes only categorical variables to split the input space. Second, a single regression model is trained based on numeric variables after tree growth while the M5 algorithm trains linear regression models at every terminal node. Basically, such differences lie in their different origins. The proposed model is devised to effectively manage the impact of categorical variables in a regression model whereas the M5 model was developed to improve the performance of regression trees. Despite such differences, we compared the performance of the proposed model with that of the M5 model tree in terms of accuracy and computational time. This is reasonable since the two algorithms resemble each other in that the objective of both algorithms is to predict continuous target variables by combining a regression model with a decision tree.

The remainder of this paper is organized as follows. In Section 2, previous studies of the inclusion of categorical variables in regression analysis and decision tree algorithms are reviewed. Section 3 provides a detailed description of the proposed hybrid regression algorithm. The experimental results are presented in Section 4. Finally, the conclusion of the paper is summarized in Section 5.

2. Literature review

2.1. Handling categorical variables

A categorical variable is a variable that can be assigned one value of a finite set of possible values. Each value is assigned to a particular group or category. Categorical variables are of two types: a nominal variable and an ordinal variable. While an ordinal variable can be assigned a value that can be logically ordered or ranked, a nominal variable can be assigned a value that is not able to be organized in a logical sequence. In both cases, the values are qualitative and non-numeric. It is not possible to perform arithmetic operations, as defined over the field of real numbers, on these values. These distinct properties of categorical variables require new algorithms that are capable of performing regression analysis on mixed data.

One approach is to convert values of categorical variable to quantitative values and then apply general regression algorithms. This can be achieved through the use of coding systems. Nominal variables are converted to quantitative data through dummy coding, effects coding, contrast coding and so on [9,16]. For ordinal variables, another coding scheme was proposed to identify the successive levels of ordinal variables [33]. The primary disadvantage of coding systems is that the number of predictors increases significantly if the number of categories defined for nominal variables is large.

Another method is to develop new distance or similarity measures for distance-based algorithms such as the nearest neighbor algorithms. The simplest similarity measure of categorical variables is to assign a value of one or zero by comparing the categories of a categorical variable, which is referred to as the Gower similarity [13]. The distance is defined as a opposite concept of similarity and is calculated as $d(x_{ik}, x_{jk}) = 1 - s(x_{ik}, x_{jk})$. Several other measures have been provided, but the Gower similarity has been verified as the optimal measure despite its simplicity [6,30].

Detailed explanations about different coding systems and the Gower similarity are provided in the supplementary material.

2.2. Decision tree

A decision tree is a data mining algorithm that is widely used for both classification and regression problems. Initially, the decision tree was proposed to solve classification problems, e.g., Iterative Dichotomiser 3 [26] and C4.5 [28,29]. It was then extended to create a regression tree, capable of supporting regression analysis and classification (CART) [7]. Each interior node of a tree algorithm corresponds to one input variable and is split into child nodes based on the values of the input variable. Each terminal node represents the particular value of a target variable, e.g., the specific class for classification problems and the specific real number value for regression problems.

We now explain the procedures to build a regression tree. In order to grow the decision tree, at every step, one of the input variables is selected to split the samples. An attribute value test is applied to a split point along a tentative variable, and the best split point is selected to divide the interior node into subsequent nodes. The purpose of growing a decision tree is to divide the input space in order to achieve smaller errors between the estimated outputs and the true outputs. In general, the estimated outputs are determined using the average of the true outputs of the training samples from a terminal node in the following way.

$$\hat{y}_i = \frac{\sum_{j \in t_i} y_j}{|t_i|} \tag{1}$$

where t_i represents the leaf node i and $|t_i|$ represents the number of samples in the leaf node i. The splitting criterion is based on the least squares deviation (LSD) impurity measure [22].

$$I(t_i) = \sum_{j \in t_i} (y_j - \hat{y}_i)^2 \tag{2}$$

where $I(t_i)$ is the impurity measure at node i. Using LSD, the splitting criterion is computed as

$$\Delta I = I(t_p) - P_l I(t_l) - P_r I(t_r) \tag{3}$$

where t_p is the parent node and t_l and t_r are the two child nodes of t_p. P_l and P_r are the proportions of data samples assigned to the left and right child nodes respectively. The split point is determined to maximize ΔI.

If the split rule is generated using a numeric or ordinal variable and the number of child nodes is two, the samples at the parent node are divided into two subsets: $\{x: x_k > s\}$ and $\{x: x_k \le s\}$ where x_k is the selected attribute and s is the split point. The same approach works for nominal predictors, but there are $2^q - 1$ possible splits for unordered categorical predictors with q categories.

Compared with other regression algorithms, the strength of the decision tree is more robust in processing categorical variables due to its inherent characteristics that predict a target value based on the induced rules. The decision tree does not require conversion of the categorical variables or definition of the distance measures for categorical variables; however, each leaf node estimates the target value as a single value although the input variables of the samples at the leaf node have different values.

2.2.1. Hybrid decision tree algorithms

The traditional approach to mitigate the weak points of different methods and to achieve better performance is by combining two different types of data mining algorithms. Hybrid decision tree algorithms have been investigated using the same concept. The decision tree has strong points, such as the ease of interpretation

and the nonparametric property, which does not require an implicit assumption regarding the underlying distribution of the input data, whereas the decision boundary lacks continuity and smoothness and the output value of the specific terminal node is shared by all cases that terminate at this node. Due to these properties, many hybrid decision tree algorithms use the decision tree as a space partitioning method to complement the discontinuous decision boundary of a decision tree. Splitting an internal node is the same as dividing the subspace into two smaller subspaces using the split point.

Utilizing this property, classification trees assist in partitioning the input space into several subspaces such that it is easy to build better classifiers in several hybrid models. Other classification algorithms are trained on subsets at the terminal nodes, and the hybrid models achieve better accuracies while building different models at each leaf node, a procedure that requires significantly more computation and results in imbalanced problems at the subclassifiers [17–20,34].

Similar to hybrid tree algorithms for classification problems, the model tree algorithms have been applied to regression problems. In the model tree, M5, each terminal node is assigned a linear model built through a least-squares method [27,28]. Like the hybrid decision trees for classification problems, the model trees require several regression models corresponding to terminal nodes, increasing the complexity of the model.

2.2.2. Methods to obtain better splits

The recursive partitioning characteristic of decision tree algorithms is a greedy approach to identify the best split point at each internal node; however, this procedure cannot guarantee a global optimum. Therefore, several advanced methods have been applied to grow a better decision tree. One of these approaches is to integrate evolutionary algorithms within decision tree algorithms. There are two distinct works related to this approach: tree analysis with randomly generated and evolved trees (TARGET) [2,12,14] and genetic programming approach for mining continuous-valued classes (GPMCC) [2,4,25]. Both TARGET and GPMCC utilize a genetic algorithm, inspired by natural Darwinian evolution that employs concepts such as inheritance, mutation, and natural selection [35]. Initially, several candidate solutions are randomly generated and then, through cross-over and mutation, new trees are constructed from parent trees. The better solutions are selected based on a specific fitness function such as the Bayesian information criterion [31] and the extended form of the adjusted coefficient of determination of linear regression models [1,25].

The evolutionary tree algorithms were compared to other tree algorithms such as CART and M5 using various datasets from UCI [21], and the evolutionary tree algorithms achieved better results. However, the evolutionary tree algorithms require setting up many configurable options that are specified by the users and more computation time to construct the final tree than typical decision tree algorithms.

3. The proposed algorithm

The primary purpose of the proposed algorithm is to provide a new hybrid algorithm that functions and performs better for mixed data. The algorithm combines the individual strengths of a decision tree and other regression algorithms and mitigates the disadvantages of the two methods. The primary concept was inspired by the observation that the behavior of categorical variables of linear regression models that include ridge and lasso with dummy coding are always discrete and discontinuous [32,37].

The proposed algorithm trains explained variations in a target variable separately by categorical and numeric variables. The two parts of the variations in the target are estimated in series. The combinations of values of several categorical variables decide their effect on the target by utilizing the decision tree. Then, the remaining part, unexplained by categorical variables, is estimated by numeric variables through one of the many regression algorithms. To identify relations between numeric variables and the remaining part, the new target values $y_{i,\ new}$ are obtained by subtracting the estimated output of the decision tree from the true target value y_i as follows.

$$y_{i,new} = y_i - \sum_j \delta_{ij}\hat{y}_j \tag{4}$$

In this equation, $\hat{y}_j$ is the estimated target value at the leaf node j and δ_{ij} has a value of 1 if, and only if, data point i belongs to the terminal node j; and otherwise, δ_{ij} is zero. The selected regression algorithm is trained to predict $y_{i,\ new}$ using numeric variables. The final predicted value is the sum of the outputs from the two models.

$$\hat{y} = \hat{y}_{tree} + \hat{y}_{reg} \tag{5}$$

The primary difference between the proposed algorithms relative to the model tree algorithms is that the decision tree is not used for space partitioning and LSD is used to select split rules. Additionally, the model tree algorithms employ different linear regression models at every terminal node. The advantages of the proposed method can be described as follows.

- The inherent non-smoothness of categorical variables does not matter since categorical variables are only used to build a decision tree.
- By combining a nonlinear regression algorithm with the decision tree, the nonlinear property of each variable can be easily considered in the proposed algorithm.
- The computation complexity is much less when compared with the model tree algorithms because the proposed algorithm trains only one regression model after the tree growth and only uses numerical variables.
- Implementation is simple and easy.

The proposed algorithm is quite simple. Hence, it is not always superior to the competing algorithms. When only two or three categorical variables are considered, the performance gap between the proposed algorithm and the coding approach may not be significant. In addition, if the structures of subspaces significantly differ from each other, training different regression models for each of the terminal nodes may produce better performance. Thus, the proposed model will work better when (1) the number of categorical variables is large; (2) the number of observations is relatively large; and (3) the structures of the subspaces are not significantly different. When all cases are satisfied, the proposed algorithm can outperform other competing methods or achieve comparable performance with less time, without degradation in interpretability.

4. Experiments

4.1. Data description and preparation

The purpose of the proposed method is to effectively handle mixed data in real world regression problems. Therefore, we selected 10 different mixed datasets from Kaggle (http://www.kaggle.com), an online platform for predictive modeling and analytics competition on which companies and researchers post their data and data miners compete to build the best models, one mixed data from UCI repository [21], and one mixed data from Korea Energy Statistical Information System. Preprocessing procedures for datasets were provided by the supplementary material because of the limitation of pages.

Table 1
Description of preprocessed datasets.

	Sales	*Fire*	*Loan*	*House*	*Allstate*	*Nashville*	*Horse*	*Autos*	*Bike*	*Liberty*	*KDD*	*Electricity*
# of samples	556	47,294	29,697	1460	188,318	24,162	73,596	65,689	10,886	50,999	3698	2459
# of numerical var.	7	290	33	35	14	10	6	4	6	16	308	8
# of categorical var.	361	10	12	44	116	8	10	7	4	16	47	15
Max cardinality	1,132	408	164	317	471	83	401	279	12	95	984	67

Table 1 describes the preprocessed datasets. The maximum cardinality is defined as the number of the total variables in the final data after dummy variable creation.

4.2. Experimental procedures

In order to evaluate the proposed algorithm, the decision tree was connected to five different regression algorithms: ordinary linear, lasso and ridge regression, *k*-nearest neighbor regression (*k*-NN) and support vector regression (SVR). Ridge and lasso regressions were selected in addition to the ordinary linear regression because they have the ability to reduce the effect of the irrelevant variables in training models. *k*-NN and SVR can be classified as distance-based algorithms because the nearest neighbors are defined by the distance in *k*-NN and a kernel function (e.g. Gaussian kernel) used in SVR can be interpreted as a similarity measure.

For a fair comparison, 5-fold cross-validation was used, and 100 iterations of the validation were executed. The sole exception was the case in which the SVR was used for *Allstate*. Because of a large number of samples from *Allstate*, we failed to obtain the result even given sufficient time when SVR was used.[1] The mean squared error (MSE) was introduced as an evaluation metric. Except for the ordinary linear regression algorithm, four regression algorithms require parameter selection in order to train the model. All parameters of the four algorithms were selected by cross-validation. The objective function to be minimized for ridge and lasso regression is the sum of the least square term and regularization term as follows:

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\left\{\alpha\sum_{j=1}^{p}|\beta_j|+(1-\alpha)\sum_{j=1}^{p}\beta_j^2\right\} \tag{6}$$

where n is the number of samples, p is the number of variables and β_j is the estimated coefficient. When α is 1, the objective function corresponds to lasso regression. When α is 0, the objective function represents ridge regression. The parameter, λ, controls the impact of the regularization term on the total cost. For *k*-NN, k was examined with $k \in \{1, 3, \ldots, 9\}$. In SVR, the Gaussian kernel was used. γ, the parameter of the Gaussian kernel ($k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel)$) and C, the parameter to control the effect of errors, were examined within the range of a logarithmic scale of 0.01–100.

To facilitate standardization of the parameter values, all numeric variables were standardized with a mean of 0 and standard deviation of 1 as the parameters for the lasso, ridge, and SVR regressions are related to the scale of the input variables.

The proposed algorithm was compared with other methods in two dimensions. First, the proposed method combines the decision tree with one of the five regression algorithms. The results were then compared with each of the five regression algorithms. For these experiments, two different comparative approaches were considered to evaluate the effectiveness of categorical variables; (1) Use of numeric variables and dummy-coded categorical variables (M1), and (2) Use of numeric variables only (M2). Fig. 1 illustrates the overall experimental procedures for a single iteration. The same preprocessing procedures were applied to the validation set (denoted as X_{val}); this set is not depicted in Fig. 1. Secondly, the proposed method was also compared with the model tree, M5 (called as M5P in Weka) and the evolutionary tree (denoted as evtree). The proposed algorithm combined with the best linear regression algorithm, selected from the ordinary, ridge and lasso regression algorithms for each dataset, was used for comparison with M5 (uses regularized linear regression) and evtree. For M5 and evtree, R implementation of Weka and R package, named as evtree were used respectively [15].

All experiments were performed on the computer with Intel Core i7-3820 (3.6 GHz) and 64GB RAM.

4.3. Experimental results

Table 2 summarizes the experimental results for the 12 datasets. The table describes the average MSE values from 100 iterations and the standard deviation values of the MSE values. M1 and M2 are the two different comparative conditions described in Section 4.2.

Based on Table 2, regardless of the datasets, the proposed method outperformed M1 and M2. The improvements in the hybrid method in terms of MSE depended on the regression algorithm combined with the decision tree. Comparing M1 and M2, the dummy-coded categorical variables usually improved the performance. That is, M1 generally achieved smaller MSEs than M2, except in the case of ordinary linear regression; this may be related to a large number of dummy variables causing degradation in performance of the model as all dummy variables were not significant in predicting the target. Similar to this trend, the ordinary linear regression algorithm generally showed the poorest performance. This may be because it always estimates coefficients of all input variables, which can cause problems such as multicollinearity if the number of independent variables is large. Through the proposed algorithm, the ordinary linear regression combined with the decision tree can achieve much smaller errors, which is quite good for analysts because the ordinary linear regression does not require any user-defined parameter and it is fast and simple.

Table 3 illustrates the average computation times and standard deviation values to train a model. This computation time represents the mean computation time to train a model using one training set during cross-validation. For the SVR, the reported results were obtained from the best parameter setting because the training time of the SVR changed as the parameter settings changed. Although the proposed algorithm combines two algorithms to provide an estimate, the training time was less than that of M1. For the *k*-NN and SVR, the training time depended on the number of samples, but the number of variables affected the computation time during the calculation of the distance or kernel function. Hence, the computation time of M2 was significantly less than that of M1 for *Sales* that included more than 300 categorical variables.

To verify whether the proposed method outperforms M1 and M2, the Nemenyi pairwise test was applied to the experimental results [24]. We applied statistical tests for MSE and time. Overall, *p*-values of pairwise tests to compare the proposed algorithm with M1 and M2 were 5.96×10^{-5} and 8.35×10^{-3} without considering the combined regression algorithms. These results verified that the proposed algorithm significantly better than M1 and

[1] The experiment with the specific parameters had not finished for 3 days.

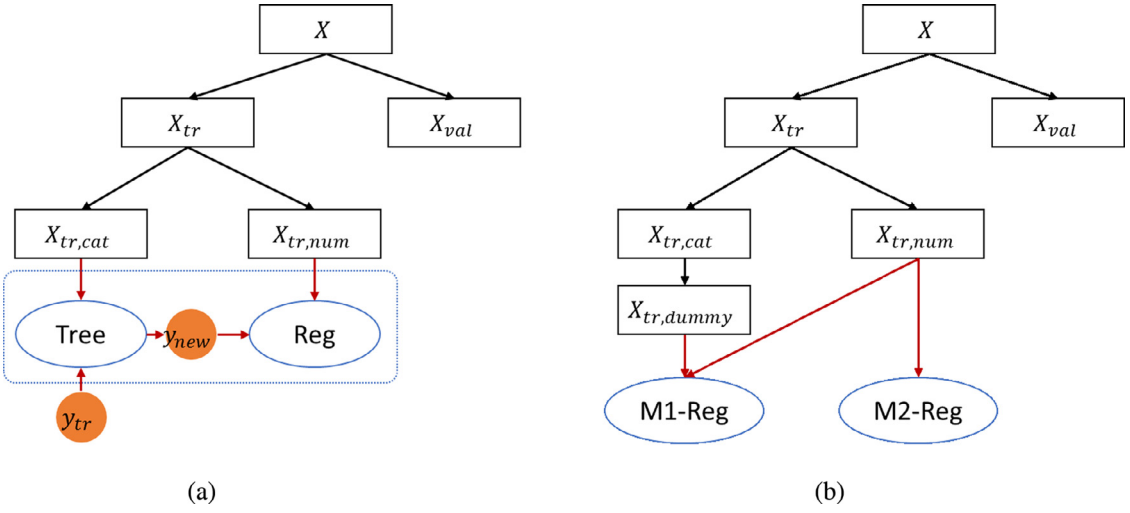

Fig. 1. Overall experimental procedures: (a) proposed (b) comparison.

Table 2
MSE values for 12 datasets.

Dataset	Method	Linear	Ridge	Lasso	*k*-NN	SVR
Sales ($\times 10^5$)	M1	$(1.11 \pm 6.82) \times 10^{12}$	2.53 ± 0.31	3.01 ± 0.21	2.32 ± 0.28	1.49 ± 0.09
	M2	3.07 ± 0.40	3.47 ± 0.21	3.47 ± 0.21	3.12 ± 0.28	3.12 ± 0.09
	Proposed	**1.81 ± 0.40**	**1.77 ± 0.10**	**1.63 ± 0.10**	**1.55 ± 0.23**	**1.24 ± 0.02**
Fire	M1	$(3.01 \pm 7.94) \times 10^{18}$	0.30 ± 0.00	0.31 ± 0.00	0.32 ± 0.00	0.31 ± 0.00
	M2	0.39 ± 0.00	0.40 ± 0.00	0.39 ± 0.00	0.34 ± 0.00	0.37 ± 0.00
	Proposed	**0.27 ± 0.00**	**0.27 ± 0.00**	**0.27 ± 0.00**	**0.30 ± 0.00**	**0.28 ± 0.00**
Loan	M1	$(3.01 \pm 7.94) \times 10^{18}$	1.11 ± 0.00	1.91 ± 0.00	8.77 ± 0.03	0.91 ± 0.00
	M2	9.17 ± 0.01	9.17 ± 0.01	9.18 ± 0.01	11.26 ± 0.03	5.45 ± 0.00
	Proposed	**0.59 ± 0.00**	**0.61 ± 0.00**	**0.58 ± 0.00**	**0.60 ± 0.00**	**0.49 ± 0.00**
House ($\times 10^9$)	M1	2.36 ± 0.51	1.93 ± 0.28	1.91 ± 0.20	1.48 ± 0.29	5.90 ± 0.01
	M2	1.55 ± 0.01	1.55 ± 0.01	1.54 ± 0.01	1.56 ± 0.00	5.88 ± 0.00
	Proposed	**1.08 ± 0.04**	**1.08 ± 0.042**	**1.08 ± 0.00**	**1.19 ± 0.00**	**1.48 ± 0.00**
Allstate	M1	$(7.46 \pm 9.00) \times 10^{16}$	0.41 ± 0.00	0.49 ± 0.00	0.48 ± 0.00	–
	M2	0.64 ± 0.00	0.65 ± 0.00	0.64 ± 0.00	0.69 ± 0.00	–
	Proposed	**0.46 ± 0.00**	**0.38 ± 0.00**	**0.38 ± 0.00**	**0.43 ± 0.00**	–
Nashville ($\times 10^{10}$)	M1	3.29 ± 0.00	3.03 ± 0.00	3.03 ± 0.00	3.27 ± 0.00	8.56 ± 0.00
	M2	3.91 ± 0.00	3.51 ± 0.00	3.51 ± 0.00	3.61 ± 0.00	8.13 ± 0.00
	Proposed	**2.71 ± 0.00**	**2.82 ± 0.00**	**2.72 ± 0.00**	**3.08 ± 0.00**	**4.28 ± 0.00**
Horse	M1	$(2.11 \pm 8.75) \times 10^{18}$	1.24 ± 0.00	1.25 ± 0.00	22.68 ± 0.06	1.61 ± 0.00
	M2	1.73 ± 0.00	1.75 ± 0.00	1.74 ± 0.00	3.87 ± 0.02	1.91 ± 0.00
	Proposed	**1.13 ± 0.01**	**1.13 ± 0.01**	**1.14 ± 0.00**	**1.31 ± 0.04**	**1.21 ± 0.00**
Autos ($\times 10^6$)	M1	$(2.52 \pm 4.58) \times 10^{19}$	5.54 ± 0.01	5.52 ± 0.01	**3.38 ± 0.01**	**4.51 ± 0.00**
	M2	9.77 ± 0.04	10.19 ± 0.01	10.06 ± 0.01	4.52 ± 0.01	5.24 ± 0.00
	Proposed	**4.97 ± 0.02**	**4.97 ± 0.02**	**4.84 ± 0.01**	4.43 ± 0.01	4.70 ± 0.00
Bike ($\times 10^{-3}$)	M1	**7.22 ± 0.05**	**7.36 ± 0.05**	**1.09 ± 0.02**	$(7.62 \pm 0.00) \times 10^6$	$(1.79 \pm 0.00) \times 10^5$
	M2	30.53 ± 0.12	30.57 ± 0.12	21.92 ± 0.12	**$(3.61 \pm 0.00) \times 10^6$**	$(1.78 \pm 0.00) \times 10^5$
	Proposed	31.13 ± 0.01	30.12 ± 0.01	21.94 ± 0.02	$(6.40 \pm 0.00) \times 10^6$	**$(0.59 \pm 0.00) \times 10^5$**
Liberty	M1	14.99 ± 0.00	14.99 ± 0.00	14.99 ± 0.00	15.76 ± 0.00	15.09 ± 0.00
	M2	15.71 ± 0.00	15.75 ± 0.00	15.71 ± 0.00	16.59 ± 0.00	16.99 ± 0.00
	Proposed	**14.68 ± 0.00**	**14.73 ± 0.00**	**14.67 ± 0.00**	**15.43 ± 0.00**	**14.65 ± 0.00**
KDD ($\times 10^2$)	M1	$(1.74 \pm 1.56) \times 10^{19}$	1.26 ± 0.06	0.91 ± 0.05	1.28 ± 0.09	1.17 ± 0.01
	M2	1.17 ± 0.01	1.14 ± 0.01	0.74 ± 0.01	1.29 ± 0.09	1.35 ± 0.01
	Proposed	**0.80 ± 0.02**	**0.79 ± 0.03**	**0.44 ± 0.01**	**0.58 ± 0.05**	**0.42 ± 0.02**
Electricity ($\times 10^5$)	M1	$(7.21 \pm 35.94) \times 10^6$	2.72 ± 0.00	2.71 ± 0.00	3.01 ± 0.00	2.81 ± 0.00
	M2	2.78 ± 0.00	2.78 ± 0.00	2.75 ± 0.00	3.33 ± 0.00	2.82 ± 0.00
	Proposed	**2.58 ± 0.00**	**2.58 ± 0.00**	**2.49 ± 0.00**	**2.80 ± 0.00**	**2.62 ± 0.00**

M2. The results of statistical tests when five regression algorithms are separately considered, are summarized in Table 5. Except for *k*-NN and lasso regression, the statistical tests verified that the proposed algorithm achieved better performance at 0.05 significant level. For lasso regression, if the result from *Bike* that is the least complex dataset was not considered, the proposed algorithm significantly outperforms M1 and M2 (p-value = 0.015). In addition, *k*-NN did show better performance that ridge and lasso regression algorithms except for *Sales* and *Autos*. In terms of computation time, M1 is not significantly faster than the proposed algorithm, which confirms that the proposed algorithm does not significantly increase computational complexity as compared to M1.

Finally, Table 4 shows results from the proposed method, M5, and evtree. Evtree ran only 10 times with *Allsate*, *Horse* and *Autos* because of long computation time. When we applied the Nemenyi test to compare MSE values of the three methods, p-value for the pair of M5 and the proposed method was 0.7500 and that for the pair of evtree and the proposed method was 0.0497. However, for computation time, p-values for the two pairs were 0.0382 and 2.9×10^{-6}. Although the proposed method achieved better or com-

Table 3
Average computation time for 12 datasets.

Dataset	Method	Linear	Ridge	Lasso	*k*-NN	SVR
Sales	M1	0.2281 ± 0.0074	0.2898 ± 0.0225	0.2828 ± 0.0393	0.0821 ± 0.0134	0.4220 ± 0.0440
	M2	0.0013 ± 0.0002	0.0020 ± 0.0021	0.0024 ± 0.0026	0.0010 ± 0.0015	0.0544 ± 0.0062
	Proposed	0.2589 ± 0.0143	0.3163 ± 0.0184	0.3056 ± 0.0214	0.3567 ± 0.0267	0.3772 ± 0.0416
Fire	M1	0.4716 ± 0.0229	0.3903 ± 0.0274	0.1591 ± 0.0119	2.0421 ± 0.3255	23.6020 ± 0.6787
	M2	0.1878 ± 0.0178	0.1254 ± 0.0152	0.0765 ± 0.0104	0.5246 ± 0.0838	30.8377 ± 0.8715
	Proposed	0.8940 ± 0.0484	0.7467 ± 0.0237	0.3163 ± 0.0184	0.9101 ± 0.1307	14.1220 ± 0.5811
Loan	M1	0.4879 ± 0.0148	0.4132 ± 0.0227	0.4511 ± 0.0461	3.5706 ± 0.5056	80.1247 ± 5.6400
	M2	0.0483 ± 0.0094	0.0372 ± 0.0064	0.0443 ± 0.0080	1.2808 ± 0.5602	56.8739 ± 3.1584
	Proposed	0.3736 ± 0.0026	0.2988 ± 0.0156	0.3433 ± 0.0339	1.6867 ± 0.5378	59.1980 ± 2.3214
House	M1	0.1328 ± 0.0369	0.1163 ± 0.0344	0.1065 ± 0.0271	0.0504 ± 0.0080	1.1369 ± 0.0203
	M2	0.0049 ± 0.0026	0.0039 ± 0.0040	0.0036 ± 0.0038	0.0181 ± 0.0044	0.4063 ± 0.0111
	Proposed	0.1069 ± 0.0059	0.0870 ± 0.0099	0.0792 ± 0.0086	0.1643 ± 0.0124	0.6085 ± 0.0153
Allstate	M1	13.6733 ± 0.7975	22.7630 ± 0.1813	10.4257 ± 0.1857	2029.3351 ± 9.7949	–
	M2	0.0711 ± 0.0087	0.1778 ± 0.0351	0.0895 ± 0.0095	1.8526 ± 0.0018	–
	Proposed	24.2133 ± 0.5548	25.4922 ± 0.4634	34.7532 ± 0.4118	34.7814 ± 0.1786	–
Nashville	M1	0.0292 ± 0.0011	0.1644 ± 0.0171	0.1611 ± 0.0127	0.6662 ± 0.0222	1.2867 ± 0.0006
	M2	0.0034 ± 0.0001	0.0137 ± 0.0052	0.0135 ± 0.0040	0.0929 ± 0.0069	1.0410 ± 0.0006
	Proposed	0.1657 ± 0.0062	0.2897 ± 0.0.0149	0.2897 ± 0.0110	0.6086 ± 0.0171	1.5496 ± 0.0078
Horse	M1	0.7381 ± 0.0370	5.1595 ± 0.2701	4.6912 ± 0.1242	1.5291 ± 0.0091	11.5589 ± 0.4552
	M2	0.0062 ± 0.0001	0.0151 ± 0.0039	0.0142 ± 0.0018	0.1367 ± 0.0017	5.8742 ± 0.4394
	Proposed	0.5342 ± 0.0060	0.5975 ± 0.0378	1.1476 ± 0.0310	1.3956 ± 10.5466	1.5291 ± 0.2463
Autos	M1	0.0127 ± 0.0081	2.7680 ± 0.1053	2.7100 ± 0.0380	15.0939 ± 3.8590	6.3028 ± 0.1231
	M2	0.0069 ± 0.0053	0.0202 ± 0.0042	0.0171 ± 0.0016	0.0813 ± 0.0272	5.2643 ± 0.6914
	Proposed	0.0823 ± 0.0126	0.9062 ± 0.0548	0.7854 ± 0.0360	1.0195 ± 0.2628	5.6350 ± 0.3906
Bike	M1	0.0075 ± 0.0046	0.0127 ± 0.0081	0.0124 ± 0.0062	0.1177 ± 0.0105	22.1354 ± 0.1127
	M2	0.0053 ± 0.0051	0.0069 ± 0.0053	0.0091 ± 0.0073	0.0403 ± 0.0062	19.2855 ± 0.1277
	Proposed	0.0565 ± 0.0067	0.0890 ± 0.0126	0.0929 ± 0.0130	0.2206 ± 0.0127	25.5029 ± 0.1026
Liberty	M1	0.4578 ± 0.0356	0.4605 ± 0.0257	0.4466 ± 0.0228	42.3747 ± 8.43889	150.2901 ± 4.2797
	M2	0.0515 ± 0.0118	0.0520 ± 0.0113	0.0504 ± 0.0074	5.8443 ± 1.5760	60.3437 ± 1.7583
	Proposed	2.1136 ± 0.2502	2.2163 ± 0.0867	2.1233 ± 0.0844	7.2399 ± 1.8984	120.6253 ± 3.6107
KDD	M1	0.8236 ± 0.0427	3.6769 ± 0.3818	0.3555 ± 0.0276	0.7545 ± 0.1028	27.1994 ± 0.3749
	M2	0.3499 ± 0.0366	0.2763 ± 0.0252	0.1593 ± 0.0108	0.5422 ± 0.5422	22.9722 ± 0.3264
	Proposed	0.6759 ± 0.0507	0.5281 ± 0.0249	0.2760 ± 0.0207	2.7099 ± 0.4552	29.6066 ± 0.2192
Electricity	M1	0.0096 ± 0.0023	0.0109 ± 0.0027	0.0067 ± 0.0024	0.0457 ± 0.0019	1.7112 ± 0.0173
	M2	0.0030 ± 0.0016	0.0035 ± 0.0018	0.0030 ± 0.0017	0.0045 ± 0.0016	1.0290 ± 0.0092
	Proposed	0.0296 ± 0.0071	0.0328 ± 0.0079	0.0694 ± 0.0073	0.0413 ± 0.0060	1.3658 ± 0.0105

Table 4
Comparison with M5 and evtree.

	Method	*Sales*	*Fire*	*Loan*	*House*	*Allstate*	*Nashville*
MSE	Proposed	$\mathbf{1.63 \pm 0.40}(\times \mathbf{10^5})$	**0.27 ± 0.00**	**0.58 ± 0.00**	$\mathbf{1.08 \pm 0.00}(\times \mathbf{10^9})$	0.37 ± 0.00	$\mathbf{2.72 \pm 0.00}(\times \mathbf{10^{10}})$
	M5	$2.78 \pm 0.43(\times 10^5)$	0.27 ± 0.01	0.55 ± 0.01	$1.30 \pm 0.22(\times 10^9)$	**0.36 ± 0.00**	$2.75 \pm 0.01(\times 10^{10})$
	evtree	$4.89 \pm 0.80(\times 10^5)$	0.34 ± 0.04	0.53 ± 0.00	$1.22 \pm 0.80(\times 10^9)$	0.37 ± 0.00	$3.50 \pm 0.05(\times 10^{10})$
Time	Proposed	0.3056 ± 0.0214	0.3163 ± 0.0184	0.3343 ± 0.0339	0.0792 ± 0.0086	34.75 ± 0.41	0.2897 ± 0.0110
	M5	1.4736 ± 0.0648	28.32 ± 0.94	18.57 ± 0.43	0.8911 ± 0.3582	7,991 ± 560	5.3814 ± 0.0823
	evtree	86.89 ± 39.98	19,768 ± 1,321	22,577 ± 403.56	113.90 ± 22.64	94,351 ± 121	23,759 ± 112
	Method	*Horse*	*Autos*	*Bike*	*Liberty*	*KDD*	*Electricity*
MSE	Proposed	**1.13 ± 0.01**	$4.84 \pm 0.01(\times 10^6)$	$2.09 \pm 0.00(\times 10^{-2})$	**14.67 ± 0.00**	**44.61 ± 1.12**	$\mathbf{2.49 \pm 0.00}(\times \mathbf{10^5})$
	M5	1.15 ± 0.00	$\mathbf{3.51 \pm 0.01}(\times \mathbf{10^6})$	$\mathbf{1.57 \pm 0.01}(\times \mathbf{10^{-3}})$	14.82 ± 0.00	83.04 ± 3.24	$2.72 \pm 0.00(\times 10^5)$
	evtree	1.12 ± 0.00	$3.63 \pm 0.11(\times 10^6)$	155.30 ± 13.72	15.01 ± 0.45	92.18 ± 6.57	$2.74 \pm 0.00(\times 10^5)$
Time	Proposed	0.5342 ± 0.0060	0.7854 ± 0.0360	0.0929 ± 0.0130	2.1233 ± 0.0844	0.2760 ± 0.0207	0.0694 ± 0.0073
	M5	48.5325 ± 1.3908	140.65 ± 9.26	0.3160 ± 0.0268	32.00 ± 2.48	25.77 ± 36.93	0.4538 ± 0.0229
	evtree	30,480 ± 1, 128	28,603 ± 161	3,911 ± 112	10,113 ± 1, 315	165.02 ± 37.71	11.70 ± 0.47

Table 5
p-values of Nemenyi pairwise test.

	Pairs	Linear	Ridges	Lasso	*k*-NN	SVR
MSE	M1-Proposed	0.0007	0.0380	0.0638	0.1020	0.0284
	M2-Proposed	0.0380	0.0001	0.0003	0.0120	0.0003
Time	M1-Proposed	0.6928	0.9122	0.9122	0.6928	0.4070
	M2-Proposed	0.0001	0.0003	0.0003	0.0031	0.1330

parable performance, the computation time was much less than M5 and evtree. In decision tree, every input variable is examined to find the best splits at appropriate nodes. The complexity to construct a decision tree is $O(d \cdot n\log(n))$ [3] where d is the number of variables and n is the number of samples. Therefore, the computation time to train M5 increases faster than the proposed algorithm when a dataset consists of a large number of samples. Empirically, evtree required considerably longer computation time for datasets with large n such as *Allstate*. This may be because evtree generates several trees for each population and iterates the selection and genetic operations until convergence which may consume substantial time if the data contains many samples and variables.

5. Conclusion

The primary objective of the proposed hybrid regression algorithm is to effectively process both categorical and numeric variables for inclusion in regression analysis. The primary concept of the proposed algorithm is to utilize a decision tree to estimate the effect of the categorical variables on the continuous target variable since a decision tree supports categorical variables as well as nu-

meric variables and is inherently non-parametric. In the proposed algorithm, the decision tree estimates that portion of the target that is only explained by categorical variables. Then, the remaining part is estimated using another regression technique that can provide a smooth estimate using numeric variables.

The proposed algorithm combines a decision tree with five regression algorithms, i.e., the ordinary linear, ridge and lasso regression, *k*-NN regression, and SVR. Each hybrid model was compared with each regression algorithm in two different cases according to the inclusion of categorical variables. The experiments were performed using the 12 datasets from various industries including both numerical and categorical variables. It was confirmed that the proposed algorithm achieved significant improvement in MSE. Moreover, the proposed algorithm is comparable to M1, in terms of computation time, even though the algorithm is a hybrid of two different algorithms. From the experiments that compared the proposed algorithm with M5 and evtree, the proposed algorithm showed comparable or better performance with significantly less computation time.

In summary, the advantages of the proposed algorithm are as follows; (1) the proposed algorithm does not require a coding system or dissimilarity/similarity functions for mixed data, (2) the proposed algorithm presents good interpretability, (3) the nonlinearity can be incorporated within the model in both categorical variables and numeric variables, (4) the computational complexity of the proposed algorithm increases slightly compared with a single regression model, and (5) the proposed algorithm is much faster than M5 and evtree although it achieved better or slightly worse performance on the tested datasets.

This work is limited because it does not consider the interaction between the categorical and numeric variables. In a future work, a method of analyzing the interaction between the two different types of variables while maintaining the simplicity of the proposed algorithm will be investigated.

Acknowledgment

This study was supported by the Research Program funded by the SeoulTech(Seoul National University of Science and Technology).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patrec.2017.08.011.

References

[1] R. Anderson-Sprecher, Model Comparisons and R^2, Am. Stat. 48 (2) (1994) 113–117.
[2] R.C. Barros, M.P. Basgalupp, A.C.P.L.F. de Carvalho, A.A. Freitas, A survey of evolutionary algorithms for decision-Tree induction, IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) 42 (3) (2012) 291–312, doi:10.1109/TSMCC.2011.2157494.
[3] R.C. Barros, A. de Carvalho, A.A. Freitas, Automatic Design of Decision-Tree Induction Algorithms, Springer Briefs in Computer Science, Springer International Publishing, 2015.
[4] R.C. Barros, D.D. Ruiz, M.P. Basgalupp, Evolutionary model trees for handling continuous classes in machine learning, Inf. Sci. 181 (5) (2011) 954–971.
[5] E. Boj Del Val, M.M. Claramunt Bielsa, J. Fortiana, Selection of predictors in distance-Based regression, Commun. Stat. Simul.Comput. 36 (1) (2007) 87–98.
[6] S. Boriah, V. Chandola, V. Kumar, Similarity measures for categorical data: acomparative evaluation, in: Proceedings of the {SIAM} International Conference on Data Mining, {SDM} 2008, April 24–26, 2008, Atlanta, Georgia, {USA}, 2008, pp. 243–254.
[7] L. Breiman, J. Friedman, R. Ohlsen, C. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984.
[8] J. Cohen, P. Cohen, S.G. West, L.S. Aiken, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, Taylor & Francis, 2013.
[9] J. Cohen, P. Cohen, S.G. West, L.S. Aiken, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, Taylor & Francis, 2013.
[10] C.M. Cuadras, C. Areans, J. Fortiana, Some computational aspects of a distance - based model for prediction, Commun. Stat. - Simul.Comput. 25 (3) (1996) 593–609, doi:10.1080/03610919608813332.
[11] C.M. Cuadras, C. Arenas, A distance based regression model for prediction with mixed data, Commun. Stat. - Theory Methods 19 (6) (1990) 2261–2279.
[12] G. Fan, J.B. Gray, Regression tree analysis using TARGET, J. Comput. Graphical Stat. 14 (1) (2005) 206–218.
[13] J.C. Gower, A general coefficient of similarity and some of its properties, Biometrics 27 (4) (1971) 857–871.
[14] J.B. Gray, G. Fan, Classification tree analysis using TARGET, Comput. Stat. Data Anal. 52 (3) (2008) 1362–1372.
[15] T. Grubinger, A. Zeileis, K.-P. Pfeiffer, Evtree : evolutionary learning of globally optimal classification and regression trees in *R*, J. Stat. Softw. 61 (1) (2014) 1–29, doi:10.18637/jss.v061.i01.
[16] M.A. Hardy, Regression with dummy variables, 1993, doi:10.1080/00401706.1994.10485828 Eric R. Ziegel.
[17] B.W. Heumann, An object-based classification of mangroves using a hybrid decision treesupport vector machine approach, Remote Sens. 3 (12) (2011) 2440–2460.
[18] L. Jiang, C. Li, Scaling up the accuracy of decision-tree classifiers: a Naive-Bayes combination, J. Comput. 6 (7) (2011).
[19] R. Kohavi, Scaling up the accuracy of Naive-Bayes Classifiers: a decision-tree hybrid, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 202–207.
[20] P.M. Kuhnert, K.-A. Do, R. McClure, Combining non-parametric models with logistic regression: an application to motor vehicle injury data, Comput. Stat. Data Anal. 34 (3) (2000) 371–386, doi:10.1016/S0167-9473(99)00099-7.
[21] M. Lichman, {UCI} Machine Learning Repository, 2013.
[22] G.J. Mclachlan, S.-k. Ng, X. Wu, V. Kumar, The top ten algorithms in data mining, 2009.
[23] L. Meier, S. Van De Geer, P. Bühlmann, The group lasso for logistic regression, J. R. Stat. Soc. 70 (1) (2008) 53–71.
[24] P.B. Nemenyi, Distribution-free Multiple Comparisons, Ph.D. thesis, Princeton University, 1963.
[25] G. Potgieter, A.P. Engelbrecht, Evolving model trees for mining data sets with continuous-valued classes, Expert Syst. Appl. 35 (4) (2008) 1513–1532.
[26] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[27] J.R. Quinlan, Learning with continuous classes, in: 5th Australian Joint Conference on Artificial Intelligence, 92, Singapore, 1992, pp. 343–348.
[28] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[29] J.R. Quinlan, Improved use of continuous attributes in C4.5, J. Artif. Intell. Res. 4 (1) (1996) 77–90.
[30] T.R. dos Santos, L.E. Zárate, Categorical data clustering: what similarity measure to recommend? Expert Syst. Appl. 42 (3) (2015) 1247–1260.
[31] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.
[32] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, J. R. Stat. Soc 73 (3) (2011) 273–282.
[33] S.D. Walter, A. R. Feinstein, C.K. Wells, Coding ordinal independent variables in multiple regression analyses, Am. J. Epidemiol. 125 (2) (1987) 319–323.
[34] L.M. Wang, X.L. Li, C.H. Cao, S.M. Yuan, Combining decision tree and Naive Bayes for classification, Knowl. Based Syst. 19 (7) (2006) 511–515.
[35] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85, doi:10.1007/BF00175354.
[36] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc. Ser. B 68 (1) (2006) 49–67.
[37] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. 67 (2) (2005) 301–320.