ACTA UNIVERSITATIS LODZIENSIS
FOLIA OECONOMICA 206, 2007

*Joanna Trzęsiok**

MULTIPLE ADDITIVE REGRESSION TREES (MART) AND THEIR APPLICATION

Abstract. Multiple additive regression trees MART is a methodology for trying to solve prediction problems in regression and classification. It's one of the boosting methods. It was introduced by J. H. Friedman (1999a). Besides accuracy, its primary goal is robustness. It tends to be resistant against outliers, missing values, and the inclusion of potentially large numbers of irrelevant predictor variables that have little or no effect on the response.

In this paper the MART algorithm and their applications will be discussed.

Key words: multivariate regression, adaptive method, regression trees, gradient boosting, MART.

1. INTRODUCTION

The subject of the regression analysis is a set of observations U. Each element of this set is characterized by the vector of variables

$$(X_1, X_2, ..., X_p, Y).$$

We look for a function f which describes the connection between response Y and predictors $X_1, X_2, ..., X_p$

$$Y = f(X_1, X_2, ..., X_p) + \varepsilon \tag{1}$$

where ε is an error rate.

* M.Sc., Department of Mathematics, Karol Adamiecki University of Economics, Katowice.

The classic solution of this problem is the least squares method. In this procedure we know an analytical form of the function f and we estimate the coefficients of f, but we must know the distribution of each predictor.

For the solving regression models the nonparametric methods were introduced. In this methods we don't have to know distributions of variables $X_1, X_2, ..., X_p$. They produce models which are often better fitted to the data than functions obtained by the least squares method. Nonparametric models are more robust and resistant against outliers (bad measurements) and missing values. The MART algorithm is a one of them.

2. BOOSTING METHODS AND "ADABOOST" ALGORITHM

Boosting methods are one of the most powerful learning ideas. They were originally designed for the classification problem, but later they were extended for regression.

In boosting methods we define the function f as a combination of many "weak" classifiers or functions of regression. Weak classifiers or functions of regression have the error rate which is only slightly better than random guessing. The accuracy of basic functions is increasing and the final model is very well fitted to the data.

"AdaBoost" is one of the first and most popular boosting algorithms developed by Y. Freund and R. Schapire (1997). This algorithm shows the idea of boosting methods very well, so it will be presented in this paper.

We consider the two-class problem. The response variable Y represents the number of class

$$Y \in \{-1, 1\}.$$

Given a vector of predictor variables $\mathbf{X}$ and classifier $G(\mathbf{X})$. The rate error on the training sample is

$$err = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G(\mathbf{x}_i)) \tag{2}$$

In the "AdaBoost" algorithm we look for the classifiers $G_1(\mathbf{x})$ on the training sample

$$U = \{(\mathbf{x}_i, y_i) : i = 1, ..., N\} \tag{3}$$

For the next iteration data are modified by applying weights $w_1, w_2, ..., w_N$:

- those observations that were misclassified have their weights increasing,
- the weights decrease for those observations that were classified correctly.

The classification algorithm is reapplied to the weighted observations and we look for new classifier $G_2(\mathbf{x})$. This procedure is repeated M times until the accuracy of the function doesn't significantly improve. We build the final model from the sequence of classifiers $G_m(\mathbf{x})$, $m = 1, 2, ..., M$.

Algorithm 1. "AdaBoost"

1. Initialize the observations weights $w_i = \frac{1}{N}$, for $i = 1, ..., N$.

2. For $m = 1, ..., M$:

a) fit a classifier $G_m(\mathbf{x})$ to the training data using $\{w_i : i = 1, ..., N\}$;

b) compute an error rate

$$err_m = \frac{\sum_{i=1}^{N} w_i \cdot I(y_i \neq G_m(\mathbf{x}))}{\sum_{i=1}^{N}} \tag{4}$$

c) compute coefficients:

$$\alpha_m = \log \frac{1 - err_m}{err_m} \tag{5}$$

d) set new weights

$$w_i \leftarrow w_i \cdot \exp(\alpha_m \cdot I(y_i \neq G_m(\mathbf{x}))) \quad \text{for} \quad i = 1, ..., N \tag{6}$$

3. The final classifier has the form

$$G(\mathbf{x}) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m \cdot G_m(\mathbf{x})\right] \tag{7}$$

The model (7) is a combination of function $G_m(\mathbf{x})$. Coefficients α_m weigh the contribution of each respective $G_m(\mathbf{x})$. The more accurate the classifiers in the sequence are the higher the influence of the effects becomes.

3. GENERALIZED BOOSTING MODEL

The model (7) may be generalized. The boosting model in a classification and regression problem takes the additive form

$$f(\mathbf{X}) = \sum_{m=1}^{M} \beta_m \cdot b(\mathbf{X}, \gamma_m) \tag{8}$$

where β_m are the expansion coefficients and $b(\mathbf{X}, \gamma_m)$ are usually simple functions of multivariate argument $\mathbf{X}$ characterized by a set of parameters γ_m.

In model (8) we give a *priori* function b and we estimate parameters β_m and γ_m. This model is fitting to the data by minimizing a loss function $L(y, f(\mathbf{x}))$ over the training sample U:

$$\min_{\{\beta_m, \gamma_m\}_{m=1,\ldots,M}} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{N} \beta_m \cdot b(\mathbf{x}_i, \gamma_m)\right) \tag{9}$$

In this minimizing problem we may use:

- a squared-error function

$$L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2 \tag{10}$$

- an absolute-error function

$$L(y, f(\mathbf{x})) = |y - f(\mathbf{x})| \tag{11}$$

- an exponential loss function for classification:

$$L(y, f(\mathbf{x})) = \exp(-y \cdot f(\mathbf{x})) \tag{12}$$

By taking different types of loss function we get the model that is more or less robust.

The solution of minimizing problem (9) is very computationally demanding, so we use forward stagewise modeling to solve it. This strategy approximates the parameters γ_m and coefficients β_m in (9) by sequentially adding new basic functions. At each iteration we look for only a local optimal solution, but this procedure may lead to the final function f close to the global solution of this task.

Algorithm 2. Forward stagewise additive modeling

1. Initialize

$$f_0(\mathbf{x}) = 0.$$

2. For $m = 1, ..., M$:

a) compute β_m and γ_m by the minimizing the function

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta \cdot b(\mathbf{x}_i, \gamma)) \tag{13}$$

b) set

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m \cdot b(\mathbf{x}, \gamma_m) \tag{14}$$

3. The final model has the form

$$f(\mathbf{x}) = f_M(\mathbf{x}) = \sum_{m=1}^{M} \beta_m \cdot b(\mathbf{x}, \gamma_m) \tag{15}$$

4. BOOSTING TREES

In model (15) we take regression trees as a basic functions b because they have good statistical properties. They incorporate mixtures of numeric and categorical predictor variables. The numeric predictor variables may be generally measured on very different scales. We don't need to know the distribution of variables. Trees are also resistant.

In data mining applications, usually only small fractions of the large number of predictor variables that have been included in the analysis are actually relevant to prediction. In models based on trees we use only these predictors which are important for the variable Y.

Regression trees partition the space of all joint predictor variable values into disjoint regions R_j (for $j = 1, ..., J$). A constant γ_j is assigned to each such region and the predictive rule is

$$\mathbf{x} \in R_j \Rightarrow f(\mathbf{x}) = \gamma_j \quad \text{for} \quad j = 1, ..., J \tag{16}$$

In the regression problem we set as a constant γ_j the average of $y_i = f(\mathbf{x})$ for $\mathbf{x}_i$ in regions R_j

$$\gamma_j = \overline{y}_i.$$

Thus the model (tree) can be formally expressed in the additive form

$$T(\mathbf{x}, \Theta) = \sum_{j=1}^{J} \gamma_j \cdot I(\mathbf{x} \in R_j) \tag{17}$$

with parameters $\Theta = \{R_j, \gamma_j\}_{j=1,\ldots,J}$.

The parameters are found by minimizing the loss function

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{j=1}^{J} \sum_{\mathbf{x}_i \in R_j} L(y_i, \gamma_j) \tag{18}$$

The boosted tree model is a sum of trees $T(\mathbf{x}, \Theta_m)$ of the form (17)

$$f(\mathbf{x}) = \sum_{m=1}^{M} T(\mathbf{x}, \Theta_m) \tag{19}$$

induced in a forward stagewise manner. At each step of this procedure we must solve the minimizing problem

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L\left(y_i, f_{m-1}(\mathbf{x}_i) + T(\mathbf{x}, \Theta_m)\right) \quad \text{for} \quad m = 1, \ldots, M \tag{20}$$

At each iteration of a forward stagewise procedure we find the region set and constants $\Theta_m = \{R_{jm}, \gamma_{jm}\}_{j=1,\ldots,J_m}$ of the next tree, given the current model $f_{m-1}(\mathbf{x}_i)$.

Given the regions R_{jm} we find the optimal constant

$$\hat{\gamma}_{jm} = \arg\min_{\gamma_{jm}} \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}) + \gamma_{jm}) \tag{21}$$

5. MINIMIZING LOSS FUNCTION – STEEPEST DESCENT ALGORITHM

The solution of minimizing problem (20) is also very computationally demanding. We find it by using one of the forward stagewise manners called the steepest descent algorithm (Friedman 1999a). At each step of this procedure we look for that tree $T(\mathbf{x}, \Theta_m)$, which gives the steepest descent of loss function. This idea is also used in the MART algorithm.

For the model (17) loss function has the form

$$L(f) = \sum_{i=1}^{N} L(y_i, f(\mathbf{x})) \tag{22}$$

where $f(\mathbf{x}) = \sum_{m=1}^{M} T(\mathbf{x}, \Theta_m)$ is a regression tree.

The goal is to minimize $L(f)$ with respect to f over the training date U. Minimizing (22) can be viewed as a numerical optimization

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} L(\mathbf{f}) \tag{23}$$

where $\mathbf{f} \in \mathbf{R}^N$ are the values of the approximating function $f(\mathbf{x}_i)$ at each of the N data points $\mathbf{x}_i$

$$\mathbf{f} = \{f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N)\} \tag{24}$$

In the "steepest descent method" we want to maximally reduce a loss function, so we move the function L towards to the negative gradient of L. Now we fix $\mathbf{f}$ at

$$\mathbf{f}_m = -\rho_m \cdot \mathbf{g}_m \quad \text{for} \quad m = 1, ..., M \tag{25}$$

where $\rho_m \in \mathbf{R}$ is a scalar and $\mathbf{g}_m \in \mathbf{R}^N$ is a gradient of $\mathrm{L}(\mathbf{f})$. The components of the gradient $\mathbf{g}_m$ are

$$g_m = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}\right]_{f(\mathbf{x}_i) = f_{(m-1)}(\mathbf{x}_i)} \quad \text{for} \quad i = 1, ..., N \tag{26}$$

The "step length" ρ_m is the solution to

$$\rho_m = \arg\min_{\rho} L(\mathbf{f}_{m-1} - \rho \cdot \mathbf{g}_m) \quad \text{for} \quad m = 1, ..., M \tag{27}$$

The current solution is then updated

$$\mathbf{f}_m = \mathbf{f}_{m-1} - \rho_m \cdot \mathbf{g}_m \tag{28}$$

and the process is repeated at the next iteration.

The steepest descent method is one of forward stagewise procedures. The estimating of parameters of the model in the stagewise approach is analogous to the solution in the steepest descent method. At each iteration in both algorithms we look for function f_m which maximally reduces L.

6. GRADIENT BOOSTING AND MART ALGORITHM

MART – multiple additive regression trees – is a tree-boosting algorithm for multivariate regression. It was developed by J. H. Friedman (1999a). MART is based on the steepest descent method, so we call it gradient boosting.

Algorithm 3. MART – multivariate adaptive regression trees

1. Initialize as the model f_0

$$f_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) \tag{29}$$

2. For $m = 1, ..., M$:

a) for $i = 1, ..., N$ compute the components of gradient of L

$$g_m = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \right]_{f = f_{m-1}} \tag{30}$$

b) fit a regression tree to the target g_m giving terminal regions R_{jm}, $j = 1, ..., J_m$;

c) for $j = 1, ..., J_m$ compute

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_j \in R_{mj}} L(y_i, f_{m-1}(\mathbf{x}) + \gamma) \tag{31}$$

d) update

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \sum_{j=1}^{J_m} \gamma_{jm} \cdot I(\mathbf{x} \in R_{jm}) \tag{32}$$

3. The final model has the form

$$f(\mathbf{x}) = f_M(\mathbf{x}) + \sum_{j=1}^{J_M} \gamma_{jM} \cdot I(\mathbf{x} \in R_{jM}) \tag{33}$$

The first line of the algorithm initializes to the optimal constant model. Next, we compute the components of the negative gradient g_{im}. They are referred to as "pseudo" residuals. We fit the regression tree to the g_{im}, update the model and repeat this procedure. In the end, we obtain model (33) which is very well fitted to the date.

7. EXAMPLE OF APPLICATION OF MART

For the illustration of the MART algorithm in regression we conduct computation on the "Boston" data set.

Data in the "Boston" set were collected by P. Harrison and D. L. Rubenfeld (1978), who conducted research on relationships between the quality of life and prices of property in the Boston area.

Data in this set are characterized by twelve predictor variables:

$X_1 = CRIM$ – *per capita* crime rate by town,
$X_2 = INDUS$ – proportion of non-retail business acres per town,
$X_3 = CHAS$ – Charles River dummy variable (1 if tract bounds river 0 otherwise),
$X_4 = NOX$ – nitrogen oxides concentration (parts per 10 million),
$X_5 = RM$ – average number of rooms,
$X_6 = AGE$ – proportion of owner-occupied units built prior to 1940,
$X_7 = DIS$ – weighted mean of distances to Boston employment centers,
$X_8 = RAD$ – index of accessibility to radial highways,
$X_9 = TAX$ – full-value property-tax rate per 10 000 dollars,
$X_{10} = PTRATIO$ – pupil-teacher ratio by town,
$X_{11} = BLACK$ – the proportion of Afro-Americans by town,
$X_{12} = LSTAT$ – lower status of the population (percent).

Response variable is $Y = MEDV$ – median value of owner-occupied homes in thousand dollars.

The MART algorithm was performed in 10 000 times (for 10 000 trees) and in a step 9851 we get the lowest test error

$$err = \sum_{i=1}^{n} |y_i - f_M(x_i)| = 1.931.$$

Figure 1 presents the average absolute test error on the set of "Boston" data. The lower line is the training error as a function of number of iterations. The upper line is the corresponding test error.

MART introduces to the model only these predictors which have the largest influence on the response variable Y. Figure 2 shows the estimated relative importance of six predictor variables. Variable $LSTAT$ – lower status of the population – is the most relevant predictor. $CHAS$ – Charles River dummy variable is the least important.

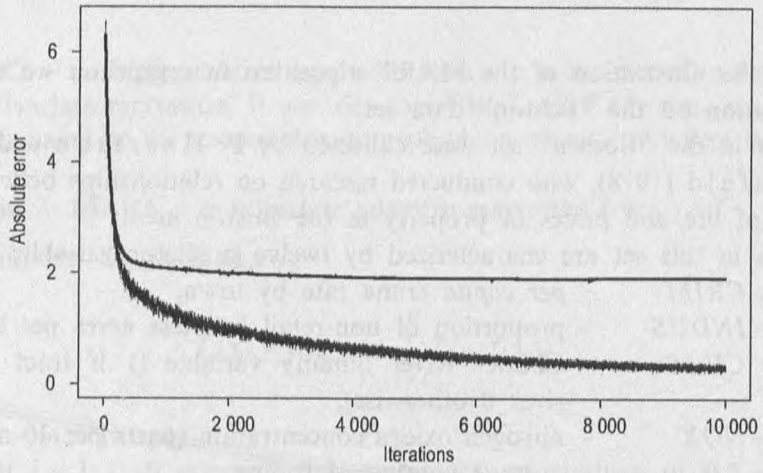

Fig. 1. Average absolute error as a function of number of iterations for the testing sample (the upper line) and training sample (the lower line)

Source: own results.

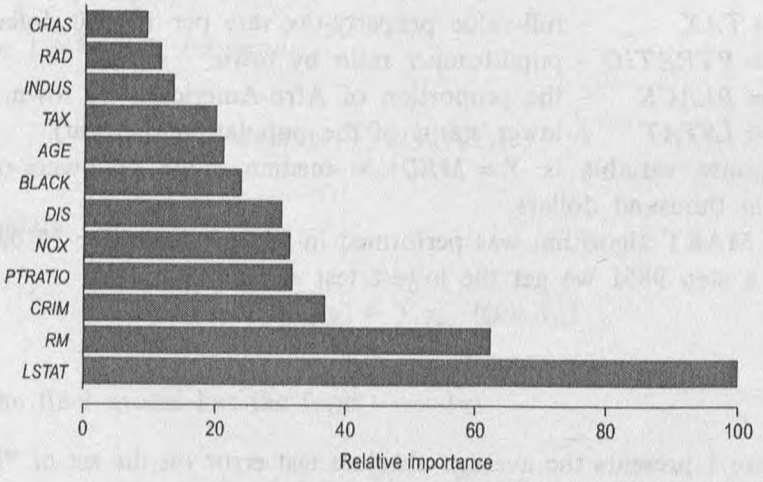

Fig. 2. Relative variable importance of the predictors for the "Boston" data

Source: own results.

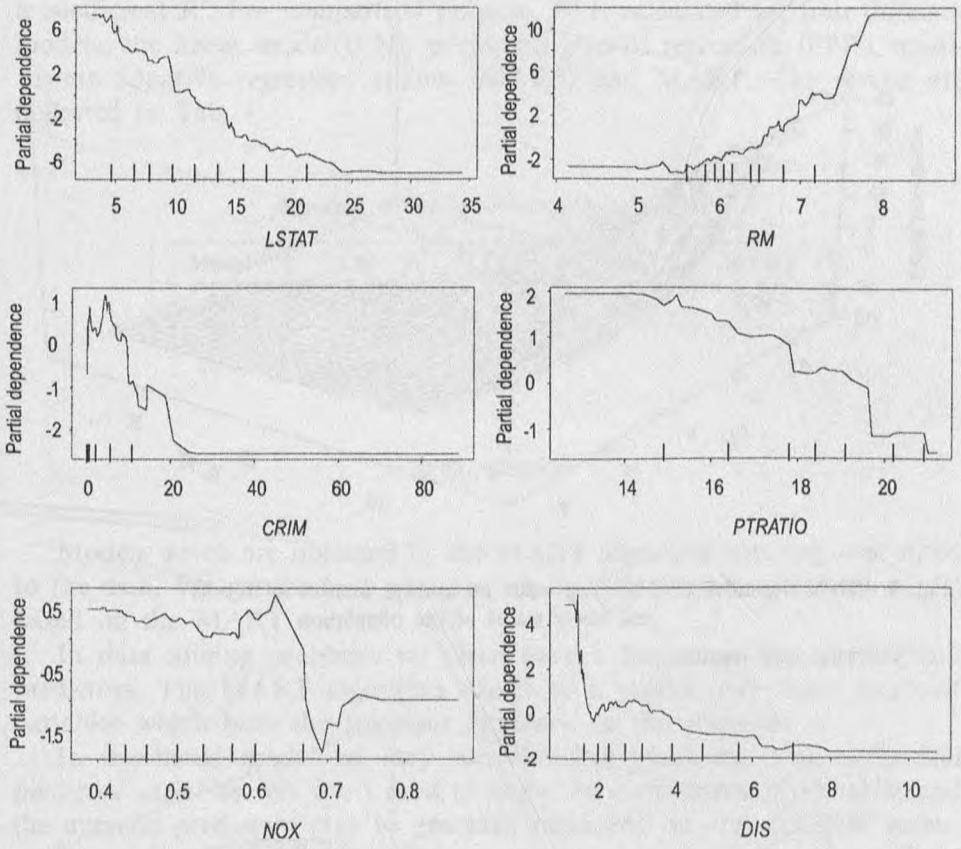

Fig. 3. Partial dependence of housing value on the different predictors for the "Boston" data

Source: own results.

Figure 3 displays single-variable partial dependence plots. It collects the results for the six most important predictor variables. Note that the plots are not strictly smooth. This is the consequence of using tree-based models which produce discontinuous piecewise constants models.

Figure 4 and 5 are perspective mesh plots showing the dependence of two chosen predictors on housing value in Boston.

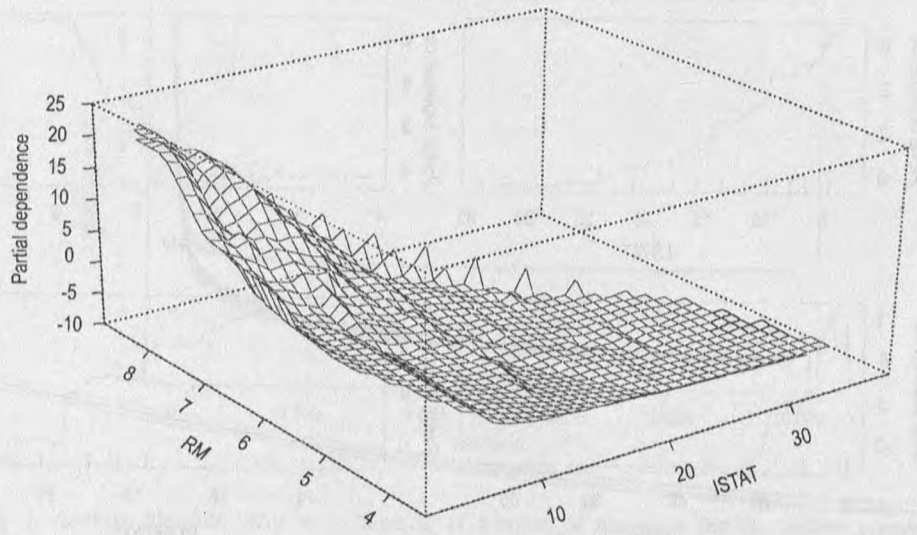

Fig. 4. Partial dependence of housing value on average number of rooms and lower status of the population

Source: own results.

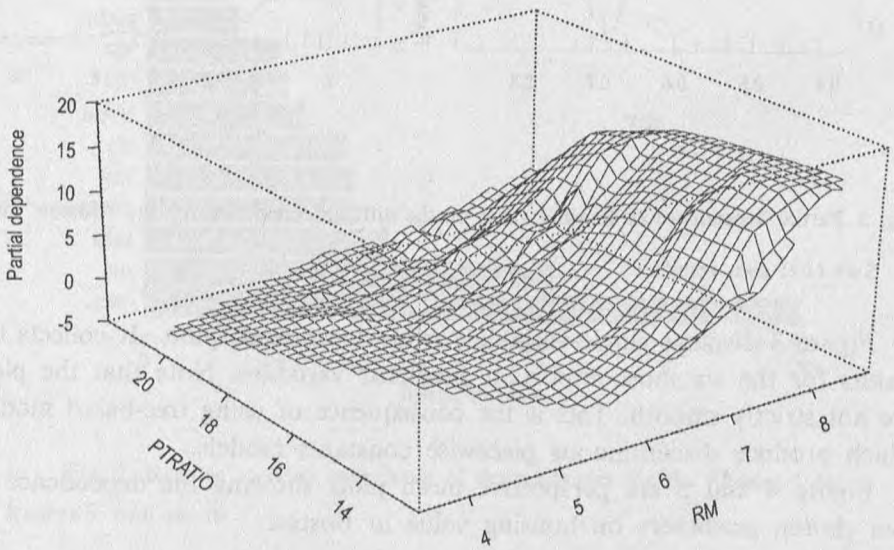

Fig. 5. Partial dependence of housing value on average number of rooms and pupil-teacher ratio by town

Source: own results.

The accuracy of the tree-based model is estimated with the help of a coefficient R^2. For comparison purpose, R^2 is calculated for four different models: the linear model (LN), projection pursuit regression (PPR), multivariate adaptive regression splines (MARS) and MART. The results are collected in Tab. 1.

Table 1

Accuracy of the various regression models

Model	LN	MARS	PPR	MART
R^2	0.779	0.780	0.883	0.988

Source: own results.

8. SUMMARY

Models which are obtained by the MART algorithm are very well fitted to the data. We get the best accuracy for the "Boston" data for the model based on the MART method.

In data mining problems we often have a large number of irrelevant predictors. The MART algorithm selects to a model only these predictor variables which have the strongest influence on the response.

In tree-based model we may use and mix numerical and categorical predictor variables. We don't need to know the distribution of variables and the numeric predictors may be generally measured on very different scales.

The primary goal of MART is robustness. It tends to be resistant against outliers (bad measurements) and missing values. MART produces competitive, highly robust, interpretability results.

REFERENCES

Freund Y., Schapire R. (1997), *A decision-theoretic generalization of on-line learning and an application to boosting*, "Journal of Computer and System Sciences", **55**, 119–139.

Friedman J. H. (1999a), *Greedy Function Approximation: A Gradient Boosting Machine Technical Report*, Dept. of Statistics, Stanford University, Stanford.

Friedman J. H. (1999b), *Stochastic Gradient Boosting Technical Report*, Dept. of Statistics, Stanford University, Stanford.

Harrison D., Rubinfeld D. L. (1978), *Hedonic Prices and the Demand for Clean Air*, "Journal of Environmental Economics and Management", **5**, 81–102.

Hastie T., Tibshirani R., Friedman J. H. (2001), *The Elements of Statistical Learning*, Springer-Verlag, New York.

Joanna Trzęsiok

ADDYTYWNA METODA BUDOWY DRZEW REGRESYJNYCH (MART) I JEJ ZASTOSOWANIE

Addytywna metoda budowy drzew regresyjnych (MART), została zaproponowana przez J. H. Friedmana w 1999 r. (1999a, b). Jest to jedna z metod agregacyjnych, mająca zastosowanie w regresji i dyskryminacji opierająca się na modelach w postaci drzew. Jej zaletami, poza dokładnością predykcji, jest odporność na wartości oddalone i braki danych. Bardzo dobrze radzi sobie również z dużą liczbą zmiennych objaśniających, wśród których wiele może nie mieć istotnego wpływu na zmienną zależną.

W artykule przedstawiona została ogólna idea metod agregacyjnych. Zaprezentowano i omówiono kolejne kroki algorytmu MART, a następnie, dla ilustracji, podany został przykład zastosowania procedury MART dla zbioru danych „Boston".