ACTA UNIVERSITATIS LODZIENSIS
FOLIA OECONOMICA 228, 2009

*Dorota Rozmus**

BOOSTING REGRESSION MODELS

ABSTRACT. In a wide variety of classification problems, boosting technique have proven to be very effective method for improving prediction accuracy (Bauer, Kohavi, 1999). While more evidence compiles about the utility of these technique in classification problems, little is known about their effectiveness in regression. Freund and Schapire (1995) gave a suggestion as to how boosting might improve regression models using their algorithm *AdaBoost.R*.

The main aim of this article is to present an application of the new boosting method for regression problems which was introduced by Ridgeway (2005). We will discuss the influence of the main parameters of this algorithm, such as eg. learning rate or number of iterations on the model performance.

Key words: regression, aggregated model (ensebles), boosting.

I. EVOLUTION OF *BOOSTING* METHOD

The starting point of this paper is an interesting procedure called "boosting", which is a way of combining many "weak" classifiers[1] to produce a powerful "committee". The first simple boosting procedure was introduced by Schapire (1990). The work on this algorithm had a culmination in the work of Freund and Schapire (1995) who introduced the *AdaBoost* algorithm. They discovered an algorithm that sequentially fits "weak" classifiers to different weightings of the observations in the data set. Those observations that the previous classifier poorly predicts receive greater weight on the next iteration. The final *AdaBoost* classifier is a weighted average of all weak classifiers. In the conclusion paper, Freund and Schapire (1995) outline their ideas for applying the boosting method for regression problem and introduced *AdaBoost.R* algorithm.

Friedman (2001) and the companion paper Friedman (1998) extended the work of Freund and Schapire (1995) and created the ground work for a new generation of boosting algorithms: gradient boosting machine. Gradient boostingconstructs

* Ph.D., Department of Statistics, The Karol Adamiecki University of Economics, Katowice.

[1] A weak learner is an algorithm for producing a two-class classifier with performance guaranteed to be better than a coinflip.

additive regression models by sequentially fitting a simple parameterized function (base learner) to current "pseudo" residuals by least-squares at each iteration. The pseudo-residuals are the gradient of the loss functional being minimized with respect to the model values at the training data point, evaluated at a current step.

Boosting technique has proven to be very effective method for improving accuracy (Bauer, Kohavi, 1999) mainly in a wide variety of classification problems. While more evidence compiles about the utility of these technique in classification, little is known about their effectiveness in regression.

The main aim of this article is to present an application of the new boosting method for regression problems which was introduced by Ridgeway (2005). The gbm package in *R* program, where this new solution is implemented, takes the approach described in Friedman (2001, 2002) and uses his gradient descent optimization algorithm.

II. GRADIENT DESCENT OPTIMIZATION ALGORITHM

In the function estimation problem we have a system with a random output y and a set of random input variables $\mathbf{x} = \{x_1, x_2, ..., x_M\}$. Given a training sample $\{\mathbf{x}_i, y_i\}$ (i = 1, 2,..., N), the goal is to find a function $F^*(\mathbf{x})$ that maps $\mathbf{x}$ to y such that the expected value of some specified loss function $L(y, F(\mathbf{x}))$ is minimized:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y,\mathbf{x}}\left[L(y, F^*(\mathbf{x})\right] \tag{1}$$

Boosting approximates $F^*(\mathbf{x})$ by an additive expansion of the form (Friedman, 2002):

$$F(\mathbf{x}) = \sum_{m=0}^{M} \gamma_m h(\mathbf{x}, \mathbf{a}_m), \tag{2}$$

where the function $h(\mathbf{x}, \mathbf{a})$ ("base learner") are usually chosen to be simply functions of $\mathbf{x}$ with parameters $\mathbf{a} = \{a_1, a_2, ...\}$. The expansion coefficients γ_m (m = 0, 1,..., M) and the parameters $\mathbf{a}_m$ are jointly fit to he training data in a forward "stage-wise" manner.

We start with an initial guess $F_0(\mathbf{x})$:

$$F_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma) \tag{3}$$

For $m = 1, 2, \ldots, M$ the algorithm determines the direction, the gradient (pseudo-residuals):

$$\tilde{y}_{im} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \tag{4}$$

in which it is needs to improve the fit to the data and selects a particular model from the allowable class of functions that is most in agreement with this direction. It means that the base learner $h(\mathbf{x}, \mathbf{a})$ is fit by least-squares to the current pseudo-residuals $\tilde{y}_{im}$:

$$\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} [\tilde{y}_{im} - \beta h(\mathbf{x}_i, \mathbf{a})]^2. \tag{5}$$

Then, given $h(\mathbf{x}, \mathbf{a}_m)$, the optimal value of the coefficient γ_m is determined:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \gamma h(\mathbf{x}_i, \mathbf{a}_m)) \tag{6}$$

At the end update $F_m(\mathbf{x})$ as:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma_m h(\mathbf{x}, \mathbf{a}_m). \tag{7}$$

Friedman (2001) takes also into account the possibility of overfitting[2] occurrence. The natural source of overfitting in boosting algorithm is the number of iterates M. To avoid this problem Friedman proposed a slight modification of (7) introducing a regularization parameter (learning rate) λ ($0 < \lambda \leq 1$):

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \lambda \gamma_m h(\mathbf{x}, \mathbf{a}_m), \tag{8}$$

[2] The problem occurs when the model is to complex that it takes into account not only the dependence but also the noise.

When using classification or regression trees Friedman relates the learning rate to regularization by shrinking. Motivated by Breiman (1999), a minor modification of gradient boosting was made to incorporate randomness as an integral part of procedure. Particularly at each iteration a subsample of the training data is drawn at random (without replacement) from the full training data set. This randomly chosen subsample is then used, instead of the whole sample, to fit the base learner and compute the updated model for the current iteration.

III. THE GBM ALGORITHM

The gbm implementation of boosting is as follows (Ridgeway, 2005):

Initialize $F_0(\mathbf{x})$ to be constatnt, $F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, \rho)$. For m = 1, 2,..., M:

1. Compute the negative gradient as a working response:

$$z_i = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \tag{9}$$

2. Select $p \times N$ cases from the dataset and fit a regression tree with K terminal nodes:

$$g(\mathbf{x}) = E(z \mid \mathbf{x}). \tag{10}$$

3. Compute the optimal terminal node prediction $\rho_1, \rho_2, ..., \rho_K$, as:

$$\rho_k = \arg\min_{\rho} \sum_{\mathbf{x}_i \in G_k} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho), \tag{11}$$

where G_k is a set of **x**s that define terminal node k.

4. Update $F_m(\mathbf{x})$ as:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \lambda \rho_{k(\mathbf{x})}, \tag{12}$$

where $k(\mathbf{x})$ indicates the index of the terminal node into which an observation with features **x** would fall,. λ is the shrinkage (or learning rate) parameter.

The main control parameters in this implementation of boosting are:

- the number of iterations M (n.tree),
- the shrinking (learning rate) parameter λ (shrinkage),
- the subsampling rate p (bagg.fraction).

IV. EXPERIMENTS

The main aim of the paper is to analyze the influence of the main parameters of the algorithm on the model accuracy. In the experiments three benchmarking datasets were used (Blake C., Keogh E., Merz C. J., 1988). They were divided into training (80%) and test (20%) sets.

Table 1. Used data sets

Name	Number of observations	Number of predictors
Boston	506	13
Ozon	366	12
Friedman 1	500	10

The aim of the first experiment was to analyze the influence of different learning rate values on the aggregated model accuracy, measured by the model error. We used four possible values: 0.001, 0.01, 0.1 and 1. The ensemble includes 10000 single models. The error was calculated as:

$$\varepsilon_{agr} = \frac{1}{\sum w_i} \sum_i w_i \left(y_i - f(\mathbf{x}_i)\right)^2. \tag{13}$$

It was calculated on training and test sets and by cross-validation method.

As there were space limits only some results will be presented. The results show that decreasing value of learning rate needs more iterations in order to gain lower resubstitution error[3]. But the error calculated on test set and by cross-validation method reveals overfitting. It is more noticeably for higher values of the learning rate. Moreover, especially in the beginning iterations, we can observe that for higher values of the learning rate, the error reduction is more rapid. With higher values of the learning rate, the error more quickly becomes very flat.

[3] It is calculated on the training set.

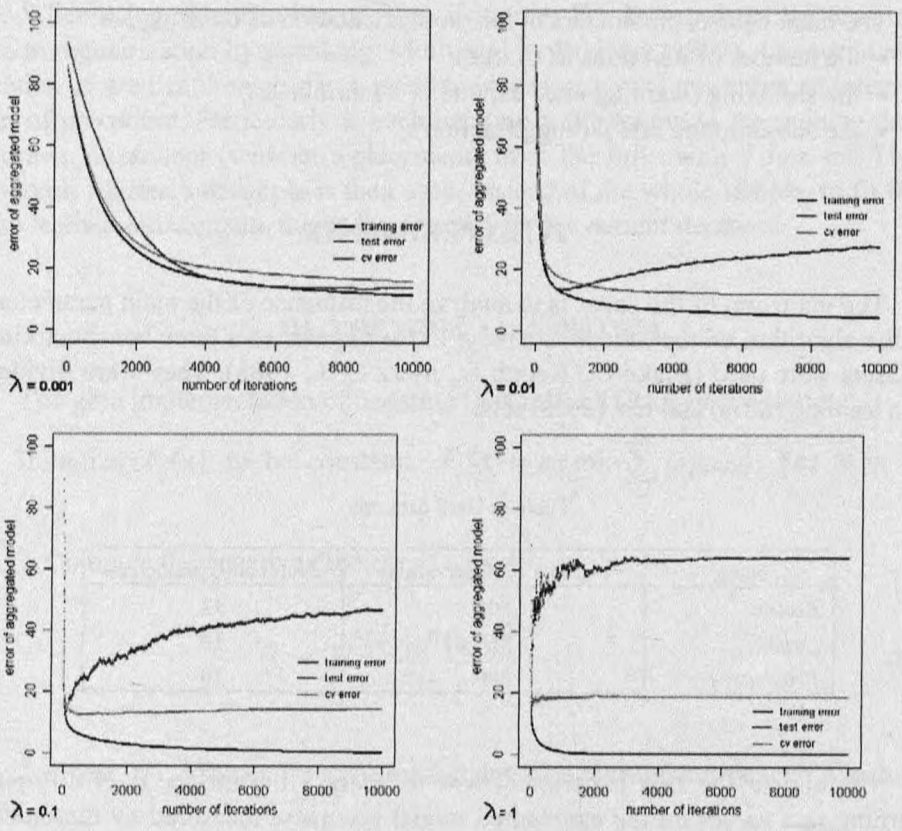

Figure 1. Influence of the learning parameter value (λ) on an aggregated model error

Experiments with other data sets confirm the same behavior. It allows us to say that the learning rate value depends on the number of iterations. For a given number of iterations (parameter M), the learning rate shouldn't be too high, because we can cause overfitting, but it also shouldn't be too low, because the model won't reduce the error enough.

The aim of the second experiment was to analyze the influence of the bagg.fraction parameter value on the error of the aggregated model. This parameter is responsible for the fraction of observations from the original training set that are chosen to the next training data subsets. In experiments we used four values: 0.3, 0.5, 0.7, 0.9. The aggregated model consists of 1000 single models, and the learning rate parameter is equal to 0.01. The results show that the more observation we choose to the subsets, the lower resubstitution error we can get, but also we can cause overfitting, what is seen on the base of test set error.

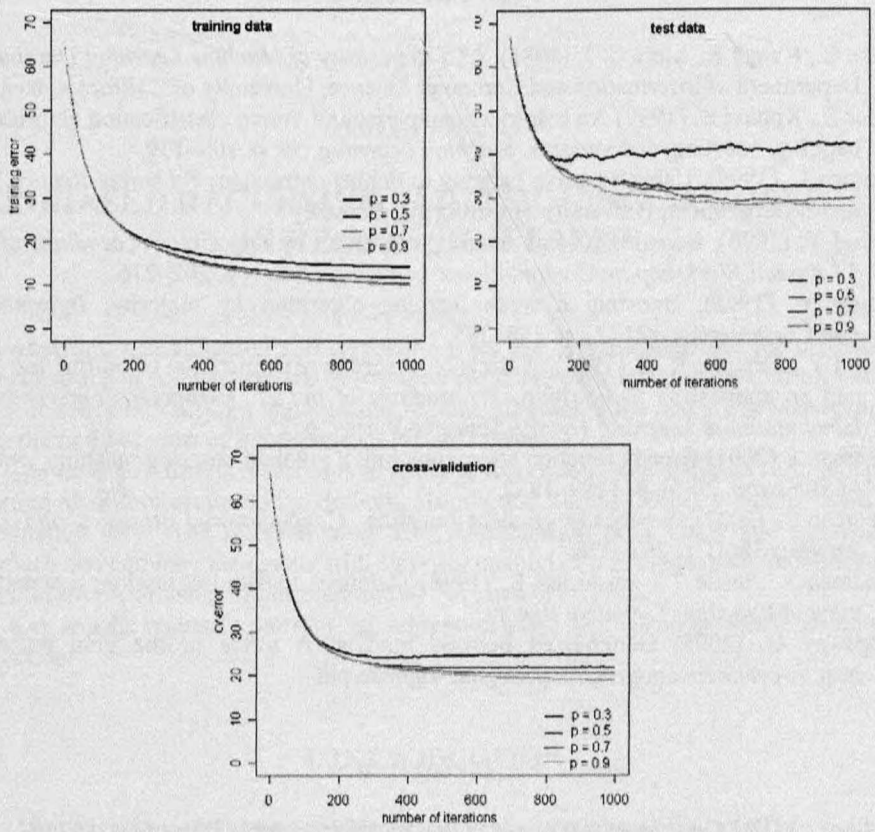

Figure 2. Influence of the bagg.fraction parameter value (p) on an aggregated model error

V. CONCLUSIONS

To summarize we can say that accuracy of aggregated model being built by the gbm package depends mainly on values of two parameters: the number of iterations (parameter M) and the learning rate (parameter λ). Very important influence on their optimal setting has the possibility of overfitting occurrence. Higher values of the learning rate need less iterations in order to protect against it. It results in shorter computational time, but we get relatively higher error value. Lower values of the λ parameter need more iterations. Time needed for model construction is longer then but lower error rate is obtained. As far the third parameter – fraction of observations chosen to following training subsets – the algorithm seems not to be very sensitive on its value.

REFERENCES

Blake C., Keogh E., Merz C. J. (1988), *UCI Repository of Machine Learning Databases.* Department of Information and Computer Science, University of California, Irvine.

Bauer E., Kohavi R. (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36, p. 105–139.

Breiman L. (1999), Using adaptive bagging to debias regression, *Technical Report*, Statistics Department, University of California, Berkeley.

Freund Y. (1990), Boosting a weak learning algorithm by majority, *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, p. 202–216.

Freund Y. (1995), Boosting a weak learning algorithm by majority, *Information and Computation*, 121 (2), p. 256–285.

Freund Y., Schapire R. E. (1995), A decision-theoretic generalization of on-line learning and an application to boosting, *Proceedings of the 2nd European Conference on Computational Learning Theory*, Springer-Verlag, p. 23–27.

Friedman J. (2001) Greedy function approximation: a gradient boosting machine, *Annals of Statistics,* 29(5), p. 1189–1232.

Friedman J. (2002), Stochastic gradient boosting, *Computational Statistics and Data Analysis* 38(4), p. 367–378.

Friedman J., Hastie T., Tibshirani R. (1998), Additive logistic regression: a statistical view of boosting, *Technical Report.*

Ridgeway G. (2005) Generalized boosted models: A guide to the gbm package, http://i-pensieri.com/gregr/papers/gbm-vignette.pdf

Dorota Rozmus

AGREGACJA MODELI REGRESYJNYCH METODĄ *BOOSTING*

Boosting jest jedną z najlepszych metod agregacji modeli dyskryminacyjnych (Bauer, Kohavi, 1999). Liczne badania empiryczne potwierdzają możliwość znacznej poprawy jakości modeli klasyfikacyjnych, niewiele jednakże wiadomo na temat efektywności tej metody w przypadku modeli regresyjnych. Freund i Schapire (1995), stosując swój algorytm *AdaBoost.R*, podjęli próbę wykorzystania metody *boosting* do tego typu zagadnień.

Głównym celem artykułu jest prezentacja nowej implementacji metody *boosting* w regresji, która opracowana została przez Ridgeway'a (2005). W przeprowadzonych eksperymentach zbadany został wpływ wartości podstawowych parametrów tego algorytmu, takich jak np. współczynnik uczenia, czy też liczba iteracji, na jakość modelu zagregowanego.