International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **6**, June **2012**

 ISSN 1349-4198
pp. 3953–3961

COMBINING BAGGING, BOOSTING AND RANDOM SUBSPACE ENSEMBLES FOR REGRESSION PROBLEMS

Sotiris Kotsiantis and Dimitris Kanellopoulos

Educational Software Development Laboratory
Department of Mathematics
University of Patras
Patras 26504, Greece
sotos@math.upatras.gr; dkanellop@teipat.gr

Received February 2011; revised August 2011

Abstract. *Bagging, boosting and random subspace methods are well known re-sampling ensemble methods that generate and combine a diversity of learners using the same learning algorithm for the base-regressor. In this work, we built an ensemble of bagging, boosting and random subspace methods ensembles with 8 sub-regressors in each one and then an averaging methodology is used for the final prediction. We performed a comparison with simple bagging, boosting and random subspace methods ensembles with 25 sub-regressors, as well as other well known combining methods, on standard benchmark datasets and the proposed technique had better correlation coefficient in most cases.*
Keywords: Machine learning, Data mining, Regression

1. Introduction.

Many regression problems involve an investigation of relationships between attributes in heterogeneous databases, where different prediction models can be more appropriate for different regions [5,9]. As a consequence multiple learner systems (an ensemble of regressors) try to exploit the local different behavior of the base learners to improve the correlation coefficient and the reliability of the overall inductive learning system [10].

Three of the most popular ensemble algorithms are bagging [3], boosting [1] and random subspace method [21]. In bagging [3], the training set is randomly sampled k times with replacement, producing k training sets with sizes equal to the original training set. Theoretical results show that the expected error of bagging has the same bias component as a single bootstrap replicate, while the variance component is reduced [6]. Boosting, on the other hand, induces the ensemble of learners by adaptively changing the distribution of the training set based on the performance of the previously created regressors. There are two main differences between bagging and boosting. First, boosting changes adaptively the distribution of the data set based on the performance of previously created learners while bagging changes the distribution of the data set stochastically [33]. Second, boosting uses a function of the performance of a learner as a weight for averaging, while bagging utilizes equal weight averaging. On the other hand, in random subspace method [21] the regressor consists of multiple learners constructed systematically by pseudo-randomly selecting subsets of components of the feature vector, that is, learners constructed in randomly chosen subspaces.

Boosting algorithms are considered stronger than bagging and random subspace method on noise-free data; however, bagging and random subspace methods are much more robust than boosting in noisy settings. For this reason, in this work, we built an ensemble combing bagging, boosting and random subspace version of the same learning algorithm using an averaging methodology. We performed a comparison with simple bagging, boosting and random subspace method ensembles as well as other known ensembles on standard benchmark datasets and the proposed technique had better correlation coefficient in most cases. For the experiments, representative algorithms of well known machine learning techniques, such as model trees, rule learners and support vector machines were used.

Section 2 presents the most well known algorithms for building ensembles that are based on a single learning algorithm, while Section 3 discusses the proposed ensemble method. Experiment results using a number of data sets and comparisons of the proposed method with other ensembles are presented in Section 4. We conclude with summary and additional research topics in Section 5.

2. **Ensembles of Regressors.** As we have already mentioned, the Bagging algorithm (Bootstrap aggregating) [3] averages regressors generated by different bootstrap samples (replicates). The main explanation of bagging operation is given in terms of its capability to reduce the variance component of the error, which was related to the degree of instability of the base learner [3], informally defined as the tendency of undergoing large changes in its decision function as a result of small changes in the training set: Theoretical investigations of why bagging works have also been found in [6,7,14]. With the influence equalization viewpoint, bagging is interpreted as a perturbation technique aiming at improving the robustness against outliers [19]. Works in the literature focused on determining the ensemble size sufficient to reach the asymptotic error, empirically showing that suitable values are between 10 and 20 depending on the particular data set and base learner [22].

Quite well known is Random Subspace Method [21], which consists of training several regressors from input data sets constructed with a given proportion k of features picked randomly from the original set of features the author of this method suggested in his experiment to select around 50.

As we have already mentioned, boosting attempts to generate new regressors that are able to better predict the hard instances for the previous ensemble members. Roughly speaking, two different approaches for boosting have been considered. The first one is related to the gradient-based algorithm following the ideas initiated by [15,33]. In each iteration, the algorithm constructs goal values for each data-point x_i equal to the (negative) gradient of the loss of its current master hypothesis on x_i. The base learner then finds a function in a class minimizing the squared error on this constructed sample. On the other hand, the AdaBoost.R algorithm [12] attacks the regression problems by reducing them to classification problems. Drucker [11] proposes a direct adaptation of the classification technique of boosting (AdaBoost) to the regression framework, which exhibits interesting performance by boosting regression trees [2]. Shrestha and Solomatine [29] also proposed Adaboost.RT which firstly employs a pre-set relative error threshold value to demarcate predictions to be correct and incorrect, and the following steps are the same as those in Adaboost that solves binary classification problems. Park and Reddy [23] proposed a scale-space based boosting framework which applies scale-space theory for choosing the optimal regressors during the various iterations of the boosting algorithm. Yin et al. [31] introduced a strategy of boosting based feature combination, where a variant of boosting is proposed for integrating different features. Redpath and Lebart [24] identified feature subset by the regularized version of Boosting, i.e., AdaboostReg. Additionally, their search strategy is the floating feature search.

Based on Principal Component Analysis (PCA), Rodriguez et al. [25] proposed a new ensemble generation technique Rotation Forest. Its main idea is to simultaneously encourage diversity and individual performance within an ensemble. Specifically, diversity is promoted by using PCA to do feature extraction for each base learner and performance is sought by keeping all principal components and also using the whole data set to train each base learner. Zhang et al. [34] investigates the performance of Rotation Forest ensemble method in improving the generalization ability of a base predictor for solving regression problems through conducting experiments on several benchmark data sets.

3. **Proposed Methodology.** Several authors [3,16,21] have proposed theories for the effectiveness of bagging, boosting and random subspace method based on bias plus variance decomposition. The success of the techniques that combine regression models comes from their ability to reduce the bias error as well as the variance error [12]. Unlike bagging and random subspace method, which is largely a variance reduction method, boosting appears to reduce both bias and variance [4]. Clearly, boosting attempts to correct the bias of the most recently constructed base model by focusing more attention on the instances that it erroneous predicted. This skill to reduce bias enables boosting to work especially well with high-bias, low-variance base models. As mentioned in [22] the main trouble with boosting seems to be robustness to noise. This is expected because noisy examples tend not to correctly predicted, and the weight will increase for these instances.

(Input LS learning set; T(= 8) number of bootstrap samples; LA learning algorithm output R* regressor)
begin
for i = 1 to T do
begin
S_i:= bootstrap sample from LS; {sample with replacement}
R_i:= LA (S_i); {generate a base regressor}
end;{endfor}
for i = T+1 to T+8 do
begin
S_i:= random projection from the d-dimensional input space to a k-dimensional subspace;
R_i:= LA (S_i); {generate a base regressor}
end;{endfor}
Initialize the observation weights $w_k = 1/n$, $i = 1, 2, \ldots, n$
for i = T+9 to T+16 do
begin
Produce from LA regressor R_i to the training data using weights w_k
Calculate the adjusted error e_k^i for each instance:
Let $D_i = \max_{j=1}^{k} |y_i - h_i(x_j)|$
Then $e_k^i = |y_k - h_i(x_k)|/D_i$
$\epsilon_i = \sum_{\mathrm{i}=1}^{k} e_k^i\ w_k^i$; if $\epsilon_i > 0,5$ stop
$\beta_i = \epsilon_i/(1-\epsilon_i)$
$w_k^{i+1} = w_k^i \beta_i^{1-e_k^i}$
R_i = the weighted median using $\ln(1/\beta_{\mathrm{k}})$ as the weight for hypothesis
end;{endfor}
Output = R* $=\sum_{\mathrm{i}=1}^{3T} R_i(x)/3T$
End

FIGURE 1. The average B&B&R algorithm

For additional improvement of the prediction of a regressor, we suggest combing bagging, boosting and random subspace methodology with averaging process (Average B&B &R). The approach is presented briefly in Figure 1. It has been observed that for bagging, boosting and random subspace method, an increase in committee size (sub-regressors) usually leads to a decrease in prediction error, but the relative impact of each successive addition to a committee is ever diminishing. Most of the effect of each technique is obtained by the first few committee members [3,17,21]. We used 8 sub-regressors for each sub-ensemble for the proposed algorithm. The presented ensemble is effective owing to representational reason. The hypothesis space h may not contain the true function f, but several good approximations. Then, by taking combinations of these approximations, learners that lie outside of h may be represented. Both theory and experiments show that averaging helps most if the errors in the individual regression models are not positively correlated [19].

4. **Comparisons and Results.** For the comparisons of our study, we used 33 well-known datasets mostly from many domains from the UCI repository [13]. In order to calculate the learners correlation coefficient, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the learner was trained on the union of all of the other subsets. Then, cross validation was run 10 times for each algorithm and the average value of the 10-cross validations was calculated. In the following tables, we represent with * that the specific ensemble looses from the proposed ensemble. That is, the proposed algorithm performed statistically better than the specific ensemble according to t-test with $p<0.05$. In addition, in the tables, we represent with v that the proposed ensemble looses from the specific ensemble according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). In the last rows of the tables one can see the aggregated results in the form (a/b/c). In this notation a means that the specific ensemble algorithm is significantly more accurate than the proposed ensemble in a out of 33 data sets, c means that the proposed ensemble is significantly more accurate than the specific ensemble in c out of 33 data sets, while in the remaining cases (b), there is no significant statistical difference between the results.

For bagging, boosting and random subspace methodology, much of the reduction in error appears to have occurred after ten to fifteen learners. But boosting continues to measurably improve their test-set error until around 25 learners [22]. For this reason, we used 25 sub-learners for all the tested ensembles of regressors. The time complexity of the proposed regressor is about the same with simple bagging, boosting and random subspace methodology with 25 sub-regressors. This happens because we use 8 sub-regressors for each sub-ensemble (totally 24).

In the sequel, we present the experiment results for different base learners. For the experiments, representative algorithms of well known machine learning techniques, such as model trees, rule learners and support vector machines were used. Model trees are the counterpart of decision trees for regression tasks [30]. Model trees generalize the concepts of regression trees, which have constant values at their leaves. Thus, they are analogous to piece-wise linear functions (and hence nonlinear). The most well known model tree inducer is the M5 [32]. Model trees can tackle tasks with very high dimensionality-up to hundreds of attributes; however, computational requirements grow rapidly with dimensionality.

We compare the presented ensemble with bagging, boosting [18], and Random-SubSpace version of M5 (using 25 sub-regressors). In the last raw of Table 1 one can see the concentrated results. The presented ensemble is significantly more accurate than Bagging M5 in 3 out of the 33 data sets, while it has significantly higher error rates in one data set. The presented ensemble is significantly more accurate than Boosting M5 in 4 out of the

TABLE 1. Comparing the proposed ensemble with other well known ensembles that uses as base regression model the M5

Dataset	Average B&B&R M5	Bagging M5		Boosting M5		Random Subspace M5	
auto93	0.9	0.89		0.87	*	0.91	
autoHorse	0.96	0.95		0.95		0.95	
autoMpg	0.94	0.93		0.94		0.92	
autoPrice	0.91	0.91		0.9		0.92	
baskball	0.64	0.64		0.65		0.64	
bodyfat	0.99	0.99		0.99		0.95	*
bolts	0.97	0.84	*	0.99		0.81	*
breastTumor	0.32	0.32		0.31		0.33	
cholesterol	0.21	0.24		0.21		0.23	
cleveland	0.72	0.73		0.7		0.73	
cloud	0.94	0.94		0.94		0.92	
cpu	0.99	0.99		0.99		0.99	
detroit	0.3	0.3		0.3		0.3	
echoMonths	0.71	0.69		0.71		0.71	
elusage	0.88	0.89		0.87		0.88	
fishcatch	0.99	0.99		0.99		0.99	
gascons	0.96	0.93	*	0.98		0.95	
housing	0.92	0.92		0.91		0.9	
hungarian	0.7	0.68		0.69		0.7	
lowbwt	0.8	0.81		0.79		0.78	
mbagrade	0.46	0.44		0.44		0.47	
meta	0.43	0.44		0.36	*	0.45	
pbc	0.62	0.62		0.6		0.61	
pharynx	0.25	0.03	*	0.06	*	0.47	v
pollution	0.8	0.8		0.79		0.77	*
pwLinear	0.93	0.94		0.94		0.9	*
quake	0.09	0.1		0.08		0.08	
sensory	0.54	0.53		0.51	*	0.53	
servo	0.93	0.94		0.94		0.82	*
strike	0.54	0.53		0.53		0.53	
veteran	0.54	0.52		0.54		0.56	
vineyard	0.7	0.74	v	0.67		0.66	*
Average	0.71	0.69		0.69		0.69	
W/D/L		1/29/3		0/29/4		1/27/6	

33 data sets whilst it has significantly higher error rates in none data set. Furthermore, Random Subspace M5 has significantly lower error rates in 1 out of the 33 data sets than the proposed ensemble, whereas it is significantly less accurate in 6 data sets. To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the M5 algorithm.

A regression rule is an IF-THEN rule that has as conditional part a set of conditions on the input features and as conclusion a regression model. M5rules algorithm produces regression rules using routines for generating a decision list from M5 Model trees [32]. We

TABLE 2. Comparing the proposed ensemble with other well known ensembles that uses as base regression model the M5rules

Dataset	Average B&B&R M5rules	Bagging M5rules		Boosting M5rules		Random Subspace M5rules	
auto93	0.89	0.89		0.86	*	0.91	
autoHorse	0.96	0.96		0.94		0.96	
autoMpg	0.93	0.93		0.93		0.92	
autoPrice	0.91	0.91		0.82	*	0.93	
baskball	0.65	0.65		0.65		0.63	
bodyfat	0.99	0.99		0.99		0.95	*
bolts	0.95	0.9	*	0.86	*	0.81	*
breastTumor	0.32	0.32		0.31		0.33	
cholesterol	0.22	0.24		0.23		0.22	
cleveland	0.72	0.73		0.7		0.73	
cloud	0.93	0.94		0.93		0.91	
cpu	0.99	1		0.98		0.99	
detroit	0.3	0.3		0.3		0.3	
echoMonths	0.71	0.69		0.71		0.71	
elusage	0.88	0.89		0.86		0.88	
fishcatch	0.99	0.99		0.99		0.99	
gascons	0.98	0.93	*	0.99		0.96	
housing	0.93	0.92		0.91		0.9	*
hungarian	0.7	0.68		0.69		0.7	
lowbwt	0.8	0.81		0.79		0.78	
mbagrade	0.46	0.44		0.44		0.47	
meta	0.43	0.54	v	0.3	*	0.42	
pbc	0.62	0.62		0.6		0.61	
pharynx	0.24	0.04	*	0.03	*	0.47	v
pollution	0.79	0.78		0.78		0.77	
pwLinear	0.93	0.94		0.94		0.9	*
quake	0.09	0.11		0.08		0.08	
sensory	0.54	0.53		0.49	*	0.52	
servo	0.93	0.95		0.94		0.82	*
strike	0.55	0.52	*	0.53		0.53	
veteran	0.53	0.5	*	0.53		0.57	v
vineyard	0.72	0.76	v	0.68	*	0.7	
Average	0.71	0.7		0.68		0.7	
W/D/L		2/26/5		0/26/7		2/26/5	

compare the presented ensemble with bagging, boosting and Random-SubSpace version of M5rules. In the last raw of Table 2 one can see the aggregated results.

The presented ensemble is significantly more accurate than Bagging M5rules in 5 out of the 33 data sets, while it has significantly higher error rates in 2 data sets. The presented ensemble is significantly more accurate than Boosting M5rules in 7 out of the 33 data sets whilst it has significantly higher error rates in none data set. Furthermore, Random Subspace M5rules has significantly lower error rates in 2 out of the 33 data sets than the proposed ensemble, whereas it is significantly less accurate in 5 data sets. To sum up, the performance of the proposed ensemble is more accurate than the other well-known ensembles that use only the M5rules algorithm.

The sequential minimal optimization algorithm (SMO) has been shown to be an effective method for training support vector machines (SVMs) on classification tasks defined on sparse data sets [35]. SMO differs from most SVM algorithms in that it does not require a quadratic programming solver. SMO is generalized so that it can handle regression problems (SMOreg) [26]. We compare the presented ensemble with bagging, boosting

TABLE 3. Comparing the proposed ensemble with other well known ensembles that uses as base regression model the SMOreg

Dataset	Average B&B&R SMOreg	Bagging SMOreg		Boosting SMOreg		Random Subspace SMOreg	
auto93	0.86	0.85		0.75	*	0.9	v
autoHorse	0.96	0.96		0.95		0.96	
autoMpg	0.92	0.92		0.92		0.91	
autoPrice	0.9	0.89		0.9		0.9	
baskball	0.63	0.64		0.61		0.62	
bodyfat	0.99	0.99		0.99		0.95	*
bolts	0.8	0.72	*	0.82		0.73	*
breastTumor	0.29	0.27		0.3		0.33	v
cholesterol	0.15	0.17		0.14		0.17	
cleveland	0.72	0.72		0.71		0.72	
cloud	0.93	0.94		0.94		0.91	
cpu	0.96	0.97		0.97		0.96	
detroit	0.3	0.3		0.3		0.3	
echoMonths	0.7	0.68		0.69		0.7	
elusage	0.86	0.87		0.87		0.84	
fishcatch	0.97	0.97		0.97		0.96	
gascons	0.98	0.99		1		0.94	*
housing	0.83	0.84		0.84		0.82	
hungarian	0.64	0.6	*	0.59	*	0.68	v
lowbwt	0.79	0.79		0.78		0.75	*
mbagrade	0.45	0.45		0.45		0.42	*
meta	0.49	0.5		0.49		0.48	
pbc	0.58	0.58		0.57		0.59	
pharynx	0.61	0.63		0.5	*	0.62	
pollution	0.8	0.81		0.75	*	0.8	
pwLinear	0.86	0.86		0.86		0.85	
quake	0.05	0.05		0.07		0.06	
sensory	0.36	0.36		0.34		0.36	
servo	0.84	0.85		0.84		0.81	*
strike	0.56	0.56		0.56		0.55	
veteran	0.52	0.52		0.52		0.51	
vineyard	0.7	0.7		0.7		0.64	*
Average	0.69	0.68		0.67		0.68	
W/D/L		0/31/2		0/29/4		3/23/7	

and Random-SubSpace version of SMOreg (using 25 sub-regressors). In the last row of Table 3 one can see the aggregated results.

The presented ensemble is significantly more accurate than Bagging SMOreg in 2 out of the 33 data sets, while it has significantly higher error rates in none data set. The presented ensemble is significantly more accurate than Boosting SMOreg in 4 out of the 33 data sets whilst it has significantly higher error rates in none data set. Furthermore, Random Subspace SMOreg has significantly lower error rates in 3 out of the 33 data sets than the proposed ensemble, whereas it is significantly less accurate in 7 data sets. To sum up, the performance of the presented ensemble is much more accurate than the other well-known ensembles that use only the SMOreg algorithm.

5. Conclusions.

An ensemble of learners is a set of regressors whose individual decisions are combined in some way (typically by weighted or unweighted averaging) to predict the values of new examples [20]. The main discovery is that ensembles are often much more accurate than the individual learners that make them up [8]. The main reason is that many learning algorithms apply local optimization techniques, which may get stuck in local optima. As a consequence even if the learning algorithm can in principle find the best hypothesis, we really may not be able to find it [27]. Building an ensemble may achieve a better approximation, even if no assurance of this is given.

In this work, we built an ensemble using an averaging methodology of bagging, boosting and random subspace ensembles. It was proved after a number of comparisons with other ensembles, that the proposed methodology gives better correlation coefficient in most cases. Finally, there is an open problem in ensemble of learners, such as how to understand and interpret the decision made by an ensemble of learners because an ensemble provides little insight into how it makes its decision. This is a research topic we are currently working on and expect to report our findings in the near future.

REFERENCES

[1] R. Avnimelech and N. Intrator, Boosting regression estimators, *Neural Computation*, vol.11, pp.491-513, 1999.

[2] S. Borra and A. D. Ciacco, Improving nonparametric regression methods by bagging and boosting, *Comput. Statist. Data Anal.*, vol.38, pp.407-420, 2002.

[3] L. Breiman, Bagging predictors, *Machine Learning*, vol.24, no.3, pp.123-140, 1996.

[4] L. Breiman, Using iterated bagging to debias regressions, *Machine Learning*, vol.45, no.3, pp.261-277, 2001.

[5] G. Brown, J. Wyatt and P. Tino, Managing diversity in regression ensembles, *Journal of Machine Learning Research*, vol.6, 2005.

[6] P. Bhlman and B. Yu, Analyzing bagging, *The Annals of Statistics*, vol.30, 2002.

[7] A. Buja and W. Stuetzle, Observations on bagging, *Statistica Sinica*, vol.16, pp.323-351, 2006.

[8] B. R. Chang and H. F. Tsai, Training support vector regression by quantum-neuron-based hopfield neural net with nested local adiabatic evolution, *International Journal of Innovative Computing, Information and Control*, vol.5, no.4, pp.1013-1026, 2009.

[9] C. C. Chou, A mixed fuzzy expert system and regression model for forecasting the volume of international trade containers, *International Journal of Innovative Computing, Information and Control*, vol.6, no.6, pp.2449-2458, 2010.

[10] T. G. Dietterich, Ensemble methods in machine learning, *Multiple Classifier Systems, LNCS*, vol.1857, pp.1-15, 2001.

[11] H. Drucker, Improving regressors using boosting techniques, *Proc. of the 14th International Conference on Machine Learning*, pp.107-115, 1997.

[12] N. Duffy and D. Helmbold, Boosting methods for regression, *Machine Learning*, vol.47, no.2-3, pp.153-200, 2002.

[13] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml, University of California, Irvine, CA, USA, 2010.

[14] J. H. Freidman and P. Hall, On bagging and nonlinear estimation, *Journal of Statistical Planning and Inference*, vol.137, no.3, pp.669-683, 2007.

[15] J. Friedman, Stochastic gradient boosting, *Computational Statistics and Data Analysis*, vol.38, pp.367-378, 2002.

[16] Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, *Proc. of ICML*, vol.96, pp.148-156, 1996.

[17] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. System Sci.*, vol.55, no.1, pp.119-139, 1997.

[18] S. Gey and J.-M. Poggi, Boosting and instability for regression trees, *Computational Statistics & Data Analysis*, vol.50, pp.533-550, 2006.

[19] Y. Grandvalet, Bagging equalizes influence, *Machine Learning*, vol.55, pp.251-270, 2004.

[20] N. L. Hjort and G. Claeskens, Frequentist model average estimators, *Journal of the American Statistical Association*, vol.98, pp.879-899, 2003.

[21] T. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, no.8, pp.832-844, 1998.

[22] D. Opitz and R. Maclin, Popular ensemble methods: An empirical study, *Artificial Intelligence Research*, vol.11, pp.169-198, 1999.

[23] J.-H. Park and C. K. Reddy, Scale-space based weak regressors for boosting, *ECML 2007, LNAI*, vol.4701, pp.666-673, 2007.

[24] D. B. Redpath and K. Lebart, Boosting feature selection, *The 3rd International Conference on Advances in Pattern Recognition*, Bath, UK, pp.305-314, 2005.

[25] J. J. Rodrguez, L. I. Kuncheva and C. J. Alonso, Rotation forest: A new classifier ensemble method, *IEEE Trans. Pattern Anal. Machine Intell.*, vol.28, no.10, pp.1619-1630, 2006.

[26] S. Shevade, S. Keerthi, C. Bhattacharyya and K. Murthy, Improvements to the SMO algorithm for SVM regression, *IEEE Transactions on Neural Networks*, vol.11, no.5, pp.1188-1183, 2000.

[27] T.-S. Shih, M.-G. Lee, C.-C. Ho and H.-M. Lee, Fuzzy regression analysis based on signed distance method, *ICIC Express Letters*, vol.4, no.6(A), pp.2235-2242, 2010.

[28] S. Shirai, M. Kudo and A. Nakamura, Bagging, random subspace method and biding, *SSPR&SPR 2008, LNCS*, vol.5342, pp.801-810, 2008.

[29] D. L. Shrestha and D. P. Solomatine, Experiments with Adaboost.RT, an improved boosting scheme for regression, *Neural Comput.*, vol.18, no.7, pp.1678-1710, 2006.

[30] L. Torgo, Inductive learning of tree-based regression models, *AI Communications*, vol.13, no.2, pp.137-138, 2002.

[31] X.-C. Yin, C.-P. Liu and Z. Han, Feature combination using boosting, *Pattern Recognition Letters*, vol.26, pp.2195-2205, 2005.

[32] Y. Wang and I. H. Witten, Induction of model trees for predicting continuous classes, *Proc. of the Poster Papers of the European Conference on ML*, pp.128-137, 1997.

[33] R. Zemel and T. Pitassi, A gradient-based boosting algorithm for regression problems, *Neural Information Precessing Systems*, vol.13, pp.696-702, 2001.

[34] C.-X. Zhang, J.-S. Zhang and G.-W. Wang, An empirical study of using rotation forest to improve regressors, *Applied Mathematics and Computation*, vol.195, pp.618-629, 2008.

[35] X. Zhang, J. Zhao and W. Wang, A new multiple kernel learning based least square support vector regression and its application in on-line gas holder level prediction of steel industry, *ICIC Express Letters*, vol.4, no.5(B), pp.1767-1772, 2010.