A Multi-agent System to Assist with Real Estate Appraisals Using Bagging Ensembles

Tadeusz Lasota[1], Zbigniew Telec[2], Bogdan Trawiński[2], and Krzysztof Trawiński[3]

[1] Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
Ul. Norwida 25/27, 50-375 Wroclaw, Poland
[2] Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
[3] European Centre for Soft Computing, Edificio Científico-Tecnológico, 3ª Planta,
C. Gonzalo Gutiérrez Quirós S/N, 33600 Mieres, Asturias, Spain

Abstract. The multi-agent system for real estate appraisals MAREA was extended to include aggregating agents, which could create ensemble models applying the bagging approach, was presented in the paper. The major part of the study was devoted to investigate to what extent bagging approach could lead to the improvement of the accuracy machine learning regression models. Four algorithms implemented in the KEEL tool, including linear regression, decision trees for regression, support vector machines, and artificial neural network of MLP type, were used in the experiments. The results showed that bagging ensembles ensured higher prediction accuracy than single models.

1 Introduction

In the last ten years the ensemble learning systems gained a large attention of the researchers. This technique combine the output of machine learning algorithms, called "weak learners", in order to get smaller prediction errors (in regression) or lower error rates (in classification). The individual estimator must provide different patterns of generalization, thus in the training process diversity is employed. Otherwise, the ensemble would be composed of the same predictors and would provide as good accuracy as the single one. It has been proved that the ensemble performs better when each individual machine learning system is accurate and makes errors on different examples [19], [20], [22], [27].

Although, there are many taxonomies for that, there is one recognized group the so-called data resampling, which generates different training sets to obtain unique regressor or classificator. To this group we may include bagging [5], [6] and boosting [10], [11], [15], [16], [30], [31]. In boostrap aggregation (bagging), each machine learning system is independently learned on resampled training set, which is randomly picked from the original samples of the training set. Hence, bagging is devoted to the unstable algorithms, where the small changes in the training set, result in large changes in the output of that system. Training of each predictor could be in parallel, in due to the independence of training of each machine. In turn, boosting provides sequential learning of the predictors. The first one is learned on the whole

N.T. Nguyen, R. Kowalczyk, and S.-M. Chen (Eds.): ICCCI 2009, LNAI 5796, pp. 813–824, 2009.

data set, while the following are learned on the training set based on the performance of the previous one. In other words, the examples which were predicted improperly are noted. Then, such examples are more probable to appear in the training set of the next predictor. It results with different machines being specialized in predicting some areas of the dataset.

Although the ensemble learning systems are more common in the classification problems, they found some applications in the regression field of machine learning. Bagging has been applied to a large number of model structures, i.e. decision trees [8], regression with variable selection [33], and neural networks [18], [35]. In the literature several types of bagging have been offered: "nice" bagging [32], subbagging the so-called subsample aggregating [9], and iterated bagging [7]. In regression, to combine generated predictors in the ensemble, weighted average [13], [19], [34] is usually used Apart from that, studies on weighted median [2], [3], [9], [12], [13], [21] were also carried out. Traditional average or median are the simplest approaches to do so.

In our previous work [24] we outlined the concept of a multi-agent system for real estate appraisals, called MAREA, employing six different machine learning algorithms. The main purpose of this paper is to extend the design of the MAREA system to include aggregating agents, which create ensemble models applying the bagging procedure to single ones. The next goal was to investigate to what extent bagging approach could lead to the improvement of the accuracy machine learning regression models devoted to assist with real estate appraisals.

2 Multi-agent System for Real Estate Appraisal

The general architecture of the extended multi-agent system for real estate appraisals (MAREA) is presented in Figure 1. Source databases are heterogeneous and distributed, they comprise a cadastral database, cadastral digital map and registry of real estate sales/purchase transactions. The source databases are continuously updated by an information centre located at a self-government office. Data agents retrieve data necessary to build appraisal models, cleanse and integrate them and save into an appraisal data warehouse. Learning agents create data driven appraisal models from scratch or update existing ones basing on the analysis of data included in the warehouse. Interface, query and result agents serve the appraisers.

Interface Agents. Interface agents interact with appraisers, i.e. system users, gather parameters of properties being appraised and present suggested prices of properties exposed to valuation.

Query Agents. Query agents are generated at each demand of a user. They apply to available models assisting with real estate appraisals. These models are used to generate the suggested prices of properties being appraised.

Result Agents. Result agents process the results provided by the models assisting with valuation and prepare suggested prices to be presented to the users.

Data Agents. Data agents process data gathered in source databases. They extract, cleanse and integrate data preparing them to be useful to learn real estate appraisal models. They create and maintain a certain kind of appraisal data warehouse.

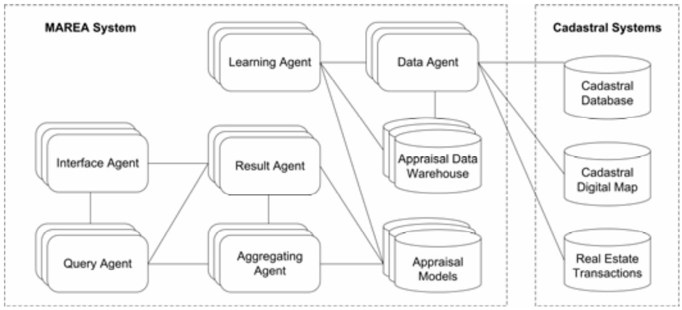

Fig. 1. General architecture of the MAREA system

Learning Agents. Learning agents perform overall and supplementary learning, creating in result different data driven models for real estate appraisal.

Aggregating Agents. Aggregating agents create ensemble models.

The JADE platform supplies mechanisms ready for communication between agents and their collaboration and mobility [4]. JADE's advantage is also possibility to create distributed systems working on many computers at once. Architecture of the MAREA system in the JADE platform is depicted in Figure 2. A MAREA system construction is brought to proper implementation of programming agents in the JADE environment. Each agent is attributed with different role during creation of MAREA system on the JADE platform: A1 – Interface Agents, A2 – Query Agents, A3 – Result Agents, A4 – Learning Agents using linear regression algorithm, A5 – Learning Agents using decision trees for regression, A6 – Learning Agents using support vector machines for regression, A7 – Learning Agents using multilayer perceptron for modelling, A8 – Aggregating Agents, and A9 – Data Agents.

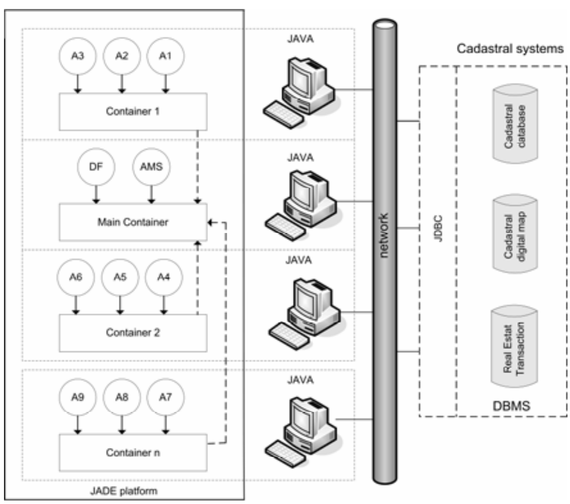

Fig. 2. Architecture of the MAREA system in the JADE platform

3 Machine Learning Algorithms Used in Experiments

KEEL is a non-commercial Java software tool designed to assess various algorithms for regression, classification, clustering, pattern mining problems [1]. KEEL contains several dozen of algorithms for data pre-processing, designing and conducting the experiments, data post-processing and evaluating and visualizing the results obtained, which have been bound into one flexible and user friendly system. The great advantage of KEEL is the possibility to create different experimental sets of data automatically and to perform cross-validation of learned models within the same process, what substantially decreases time needed to prepare and accomplish experiments and allows to avoid or diminish the threat of model overfitting. Previous investigations by the authors of this paper [23] proved the appropriateness of KEEL to build and evaluate data driven models for real estate appraisal. Four algorithms implemented in KEEL, including linear regression, decision trees for regression, support vector machines, and artificial neural network of MLP type, were applied to carry out our experiments. They are listed in Table 1:

Table 1. KEEL algorithms used in study

Code	KEEL name	Description
LRM	Regr-LinearLMS	Linear regression models
DTR	Regr-M5	Decision trees for regression
SVM	Regr-NU_SVR	Support vector machines for regression
ANN	Regr-MLPerceptronConj-Grad	Multilayer perceptron for modeling

Regr-LinearLMS (LRM). Linear regression method is a standard statistical approach to build a linear model predicting a value of the variable while knowing the values of the other variables. It uses least mean squares in order to adjust the parameters of the linear model/function [29].

Regr-M5 (DTR). The M5 model tree is a system solving regression problems. It is based on decision tree approach which can learn efficiently being capable to solve tasks with high dimensionality. As it is in decision trees, it builds tree based model, however instead of having values at their nodes it contains multivariate linear regression models at each node. The main advantage of M5 approach over traditional regression trees is that model trees are much smaller than regression trees [28].

Regr-NU_SVR (SVM). The SVM (Support Vector Machine) model uses the sequential minimal optimization training algorithm and treats a given problem in terms of solving a quadratic optimization problem. The NU-SVR, called also v-SVM, for regression problems is an extension of the traditional SVM and it aims to build a loss function [14].

MLPerceptronConj-Grad (ANN). Proposed by Moller [25] conjugate gradient based algorithm is an approach for supervised learning of the neural nets avoiding a time consuming line-search. Algorithm is performed on networks consisting of multiple layers, where each neuron on layer has directed connections to the neurons of the subsequent layer.

3 Results of Experiments

The goal of research reported was to investigate to what extent bagging approach led to the improvement of the accuracy machine learning regression models devoted to assist with real estate appraisals. Actual data used to generate and learn appraisal models came from the cadastral system and the registry of real estate transactions referring to residential premises sold in one of the big Polish cities at market prices within two years 2001 and 2002. They constituted original data set of 1098 cases of sales/purchase transactions. Four attributes were pointed out as price drivers: usable area of premises, floor on which premises are located, year of construction, number of storeys in the building, in turn, price of premises established the output variable.

Schema of the experiments is depicted in Figure 3. On the basis of the original data set 28 bootstrap replicates, called also bags, were created by drawing with replacement the elements from original data set. When determining the number of bags we followed Breiman [5], who stated that about 25 replicates seemed reasonable to obtain satisfactory ensemble. These replicates were used to generate models employing LMR, DTR, SVM and ANN regression algorithms implemented in KEEL. Normalization of data was performed using the min-max approach. As fitness function the mean square error (MSE) was applied and 10-fold cross validation (10cv) was accomplished. In consequence 112 different models were created. As aggregation functions simple averages and weighted averages of mean squared error obtained for test sets were used.

Descriptive statistics of the results obtained for individual bootstrap sets were presented in Table 2. The lowest median values of MSE were provided by SVM and ANN models and they were smaller than MSE achieved by the models constructed using original data set. By contrast, the LRM models revealed the highest value of MSE median. On the other hand the DTR models were characterized by the highest variance of MSE. In turn, the performance of models created by LRM, DTR, SVM and ANN algorithms for individual bootstrap sets against the MSE obtained for the model built over original data set are shown in Figures 4, 5, 6 and 7 respectively.

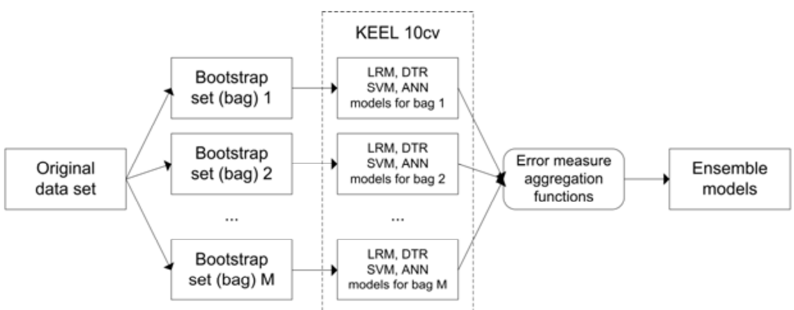

Fig. 3. Schema of bagging ensemble model development

Table 2. Descriptive statistics of performance (MSE) of individual bootstrap replicates

Bootstrap sets	LRM	DTR	SVM	ANN
min	0.00376	0.00385	0.00349	0.00346
max	0.00135	0.00117	0.00105	0.00112
median	0.00232	0.00209	0.00186	0.00198
avg	0.00243	0.00226	0.00203	0.00215
var*	0.44607	0.60825	0.41737	0.44531
Original set	**0.00253**	**0.00252**	**0.00216**	**0.00232**

- *For clarity of presentation actual values of variance were multiplied by 10^6*

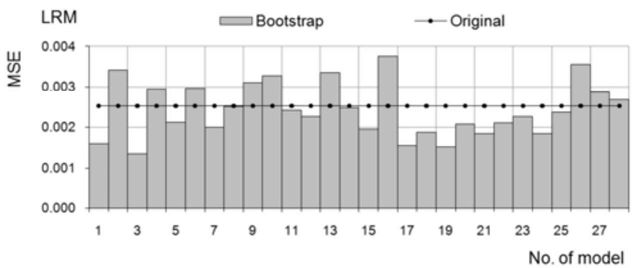

Fig. 4. Performance of individual bootstrap models created by LRM

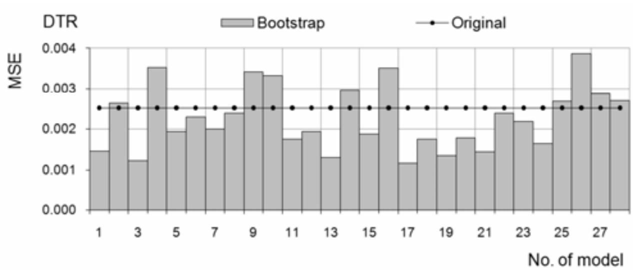

Fig. 5. Performance of individual bootstrap models created by DTR

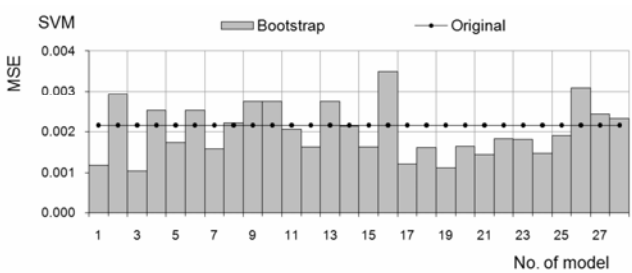

Fig. 6. Performance of individual bootstrap models created by SVM

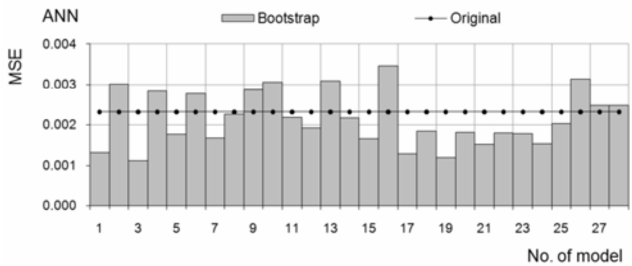

Fig. 7. Performance of individual bootstrap models created by ANN

First series of bagging models were created for individual algorithms using simple averages of prediction errors expressed in terms of MSE. The performance of bagging ensembles comprising from 1 to 28 bootstrap models was presented in Table 3 and illustrated in Figures 8, 9, 10, and 11. Table 3 contains also percentage reduction of MSE of respective ensembles compared to MSE provided by the original model. It can be observed that, except for some first ensembles, which should be passed over in order to avoid the effect of favourable drawing, the bagging models including 24 and

Table 3. Performance of bagging ensembles for individual algorithms

No. of models	LRM	Error reduc.	DTR	Error reduc.	SVM	Error reduc.	ANN	Error reduc.
1	0.00160	36.5%	0.00146	41.9%	0.00118	45.5%	0.00132	42.8%
2	0.00251	0.6%	0.00205	18.6%	0.00205	4.9%	0.00217	6.4%
3	0.00212	16.0%	0.00177	29.6%	0.00172	20.4%	0.00182	21.5%
4	0.00233	7.8%	0.00221	12.2%	0.00192	11.1%	0.00207	10.5%
5	0.00229	9.5%	0.00216	14.3%	0.00188	12.7%	0.00201	13.1%
6	0.00240	5.1%	0.00218	13.4%	0.00199	7.7%	0.00214	7.6%
7	0.00234	7.4%	0.00215	14.4%	0.00193	10.5%	0.00207	10.4%
8	0.00236	6.5%	0.00218	13.3%	0.00197	8.8%	0.00210	9.4%
9	0.00244	3.2%	0.00232	7.9%	0.00205	4.8%	0.00218	5.7%
10	0.00253	0.0%	0.00242	3.9%	0.00212	1.6%	0.00227	2.0%
11	0.00252	0.3%	0.00236	6.3%	0.00212	1.9%	0.00226	2.3%
12	0.00250	1.2%	0.00232	7.7%	0.00208	3.8%	0.00223	3.5%
13	0.00256	-1.4%	0.00224	10.9%	0.00213	1.4%	0.00230	0.7%
14	0.00256	-1.2%	0.00229	8.8%	0.00213	1.4%	0.00229	1.0%
15	0.00252	0.4%	0.00227	9.9%	0.00209	2.9%	0.00225	2.9%
16	0.00259	-2.7%	0.00234	6.8%	0.00218	-1.1%	0.00232	-0.4%
17	0.00253	-0.3%	0.00227	9.6%	0.00212	1.5%	0.00226	2.2%
18	0.00250	1.2%	0.00225	10.7%	0.00210	2.9%	0.00224	3.2%
19	0.00244	3.2%	0.00220	12.6%	0.00205	5.2%	0.00219	5.6%
20	0.00243	3.9%	0.00218	13.4%	0.00202	6.2%	0.00217	6.4%
21	0.00240	5.0%	0.00214	14.8%	0.00200	7.5%	0.00214	7.8%
22	0.00239	5.5%	0.00215	14.4%	0.00199	7.8%	0.00212	8.4%
23	0.00238	5.8%	0.00216	14.3%	0.00198	8.2%	0.00211	9.0%
24	**0.00236**	**6.6%**	**0.00213**	**15.2%**	**0.00196**	**9.2%**	**0.00208**	**10.0%**
25	**0.00236**	**6.6%**	**0.00216**	**14.3%**	**0.00196**	**9.3%**	**0.00208**	**10.1%**
26	0.00240	4.8%	0.00222	11.7%	0.00200	7.3%	0.00212	8.4%
27	0.00242	4.1%	0.00225	10.8%	0.00202	6.5%	0.00213	7.8%
28	0.00243	3.8%	0.00226	10.1%	0.00203	6.0%	0.00215	7.3%

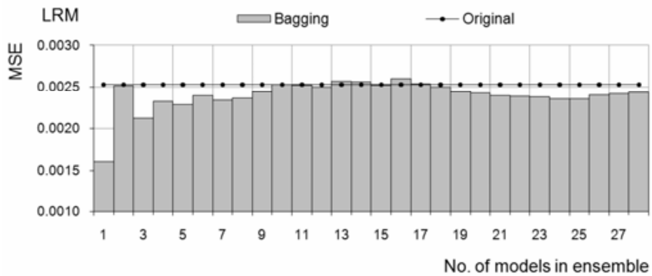

Fig. 8. Performance of bagging ensembles compared with original model created by LRM

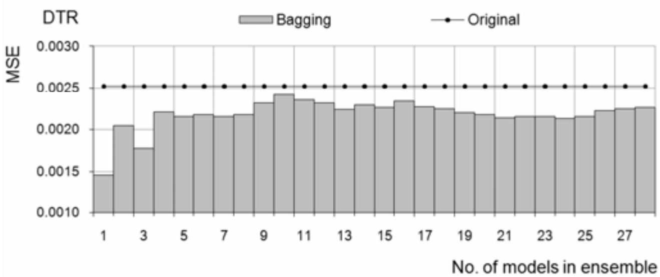

Fig. 9. Performance of bagging ensembles compared with original model created by DTR

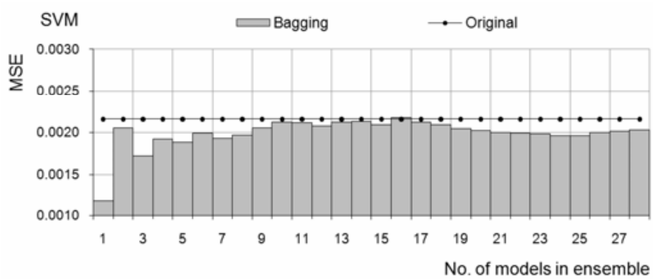

Fig. 10. Performance of bagging ensembles compared with original model created by SVM

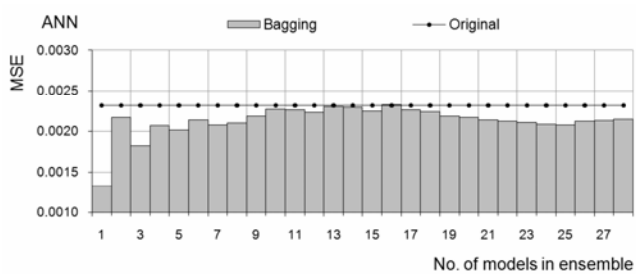

Fig. 11. Performance of bagging ensembles compared with original model created by ANN

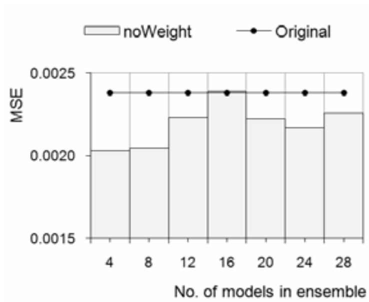

Fig. 12. Performance of ensembles using unweighted averages

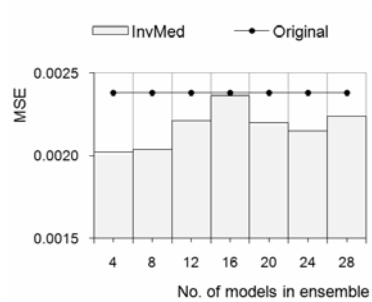

Fig. 13. Performance of ensembles using InvMed weighted averages

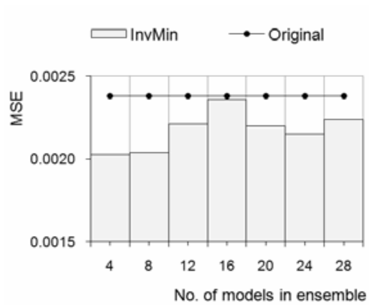

Fig. 14. Performance of ensembles using InvMin weighted averages

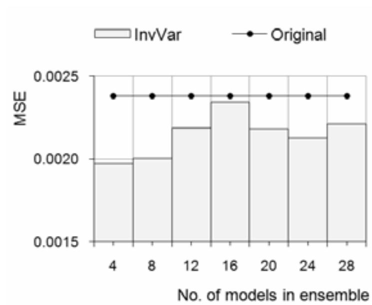

Fig. 15. Performance of ensembles using InvVar weighted averages

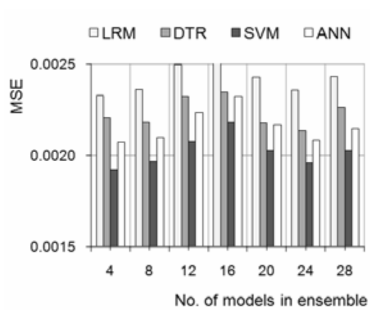

Fig. 16. Performance comparison of bagging ensembles created by individual algorithms

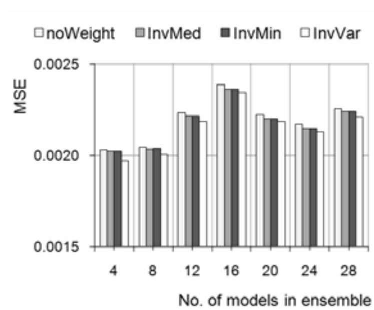

Fig. 17. Comparison of ensembles using unweighted and weighted averages

25 bootstrap replicates achieved the best prediction performance. They outperformed single models based on original data set created by LRM, DTR, SVM, and ANN algorithms, reducing average MSE by 6.6%, 15.2%, 9.3%, and 10.1% respectively. It should be also noted that at the combination of 16 models there was a point where the bagged predictors had larger prediction error than the unbagged. But this did not occurred in the case of DTR models, which were characterized by the greatest

variance of MSE. The figures confirmed the superior prediction accuracy of the bagging method to single models based on the original data set. Moreover, the results conformed to the intuitive Breiman's statement [5] that 25 bootstrap replicates could be reasonable to find the ensemble with the lowest prediction error.

Next series of bagging models encompassed the combinations of bootstrap models generated by different algorithms using different bootstrap replicates. This time, simple averages as well as weighted averages of prediction errors expressed in terms of MSE were applied. Bagging ensembles contained from 1 to 7 groups of four models obtained employing LMR, DTR, SVM and ANN to successive bootstrap sets (the results are presented in Table 4). Weights assigned to prediction errors were calculated on the basis of descriptive statistics presented in Table 3, and they were proportional to inverse median (InvMed), inverse minimum (InvMin), and inverse variance (InvVar). The performance of bagging ensembles comprising from 4 to 28 bootstrap models was illustrated in Figures 12, 13, 14, and 15. Similar observations to those referring to the first series could be done, namely except two first ensembles, the bagging models including 24 bootstrap replicates achieved the best prediction performance, and they outperformed single models based on original data set created by LRM, DTR, SVM, and ANN algorithms. The average of MSE averages calculated for all algorithms, which was equal to 0.00238, was reduced by 8.9%, 9.7%, 9.7%, and 10.5% for noWeight, InvMed, InvMin, and InvVar ensembles respectively. At the combination of 16 models only the unweighted bagged predictor had larger prediction error than the unbagged. The figures also confirmed the higher prediction accuracy of the bagging method to average of models based on the original data set.

Table 4. Performance of unweighted and weighted bagging ensembles

No. of models	noWeight	Error reduc.	InvMed	Error reduc.	InvMin	Error reduc.	InvVar	Error reduc.
4	0.00203	14.7%	0.00202	15.0%	0.00202	15.0%	0.00197	17.2%
8	0.00205	14.0%	0.00204	14.4%	0.00204	14.4%	0.00200	15.8%
12	0.00223	6.2%	0.00221	7.0%	0.00221	7.0%	0.00218	8.2%
16	0.00239	-0.3%	0.00236	0.7%	0.00236	0.8%	0.00234	1.5%
20	0.00222	6.6%	0.00220	7.5%	0.00220	7.6%	0.00218	8.2%
24	**0.00217**	**8.9%**	**0.00215**	**9.7%**	**0.00215**	**9.7%**	**0.00213**	**10.5%**
28	0.00226	5.2%	0.00224	5.9%	0.00224	5.9%	0.00221	7.0%

4 Conclusions and Future Work

An extension of the MAREA multi-agent system for real estate appraisals to include aggregating agents, which created ensemble models applying the bagging approach, was presented in the paper. The major part of the study was devoted to investigate to what extent bagging approach could lead to the improvement of the accuracy machine learning regression models. Four algorithms implemented in KEEL including linear regression, decision trees for regression, support vector machines, and artificial neural network of MLP type, were used in the experiments. 112 bootstrap models and 140 bagging ensembles combining the former by means of unweighted and weighted averages were created and evaluated in respect of prediction accuracy. The smallest values of MSE ensured bagging ensembles encompassing 24 and 25 bootstrap SVM and ANN models. However the largest percentage reduction of prediction error achieved the ensemble of DTR models, which were characterized by the greatest variance of MSE.

Moreover, bagging models comprising the combinations of bootstrap models generated by all algorithms using different bootstrap replicates were investigated. Similar observations could be done, the bagging models including 24 bootstrap replicates achieved the best prediction performance, and they outperformed single models based on original data set created by individual algorithms. The largest percentage reduction of prediction error achieved the ensemble with weights proportional to inverse variance of MSE.

Further research on ensemble models devoted to real estate appraisal is planned in order to find components assuring error diversity and to extend the algorithm range to evolutionary fuzzy systems. Moreover the impact of different model aggregation methods ranging from algebraic expressions, such as minimum, maximum, sum, mean, product, median, etc. to more sophisticated measures will be investigated.

References

1. Alcalá-Fdez, J., et al.: KEEL: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13(3), 307–318 (2009)
2. Avnimelech, R., Intrator, N.: Boosting regression estimators. Neural Computation 11, 491–513 (1999)
3. Bertoni, A., Campadelli, P., Parodi, M.: A boosting algorithm for regression. In: Proc. Int. Conference on Artificial Neural Networks, pp. 343–348 (1997)
4. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. EXP 3(3), 6–19 (2003)
5. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)
6. Breiman, L.: Stacked Regressions. Machine Learning 24(1), 49–64 (1996)
7. Breiman, L.: Using iterated bagging to debias regressions. Machine Learning 45, 261–277 (2001)
8. Büchlmann, P., Yu, B.: Analyzing bagging. Annals of Statistics 30, 927–961 (2002)
9. Büchlmann, P.: Bagging, subagging and bragging for improving some prediction algorithms. In: Akritas, M.G., Politis, D.N. (eds.) Recent Advances and Trends in Nonparametric Statistics, pp. 19–34. Elsevier, Amsterdam (2003)
10. Drucker, H., Cortes, C.: Boosting decision trees. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 479–485. Morgan Kaufmann, San Francisco (1996)
11. Drucker, H., Schapire, R.E., Simard, P.: Boosting performance in neural networks. Int. J. of Pattern Recogn. and Artificial Intel. 7(4), 705–719 (1993)
12. Drucker, H.: Improving regressors using boosting techniques. In: Proc. 14th Int. Conf. on Machine Learning, pp. 107–115. Morgan Kaufmann, San Francisco (1997)
13. Duffy, N., Helmbold, D.P.: Leveraging for regression. In: Proceedings of the 13th Conference on Computational Learning Theory, pp. 208–219 (2000)
14. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using the second order information for training SVM. J. of Mach. Learning Res. 6, 1889–1918 (2005)
15. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting, J. of Comp 55(1), 119–139 (1997)

16. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Proc. of the Thirteenth Int. Conf. on Machine Learning, pp. 148–156 (1996)
17. Friedman, J.: Greedy function approximation: a gradient boosting machine. Technical report, Dept. of Statistics, Stanford University (1999)
18. Gencay, R., Qi, M.: Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. IEEE Transactions on Neural Networks 12, 726–734 (2001)
19. Hansen, L., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
20. Hashem, S.: Optimal linear combinations of neural networks. Neural Networks 10(4), 599–614 (1997)
21. Kégl, B.: Robust regression by boosting the median. In: Proc. of the 16th Conference on Computational Learning Theory, pp. 258–272 (2003)
22. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Advances in Neural Inf. Proc. Systems, pp. 231–238. MIT Press, Cambridge (1995)
23. Lasota, T., Mazurkiewicz, J., Trawiński, B., Trawiński, K.: Comparison of Data Driven Models for the Validation of Residential Premises using KEEL. International Journal of Hybrid Intelligent Systems (in press, 2009)
24. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Concept of a Multi-agent System for Assisting in Real Estate Appraisals. In: Håkansson, A., et al. (eds.) KES-AMSTA 2009. LNCS (LNAI), vol. 5559, pp. 50–59. Springer, Heidelberg (2009)
25. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533 (1990)
26. Optiz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study. Journal of Artificial Intelligence Research 11, 169–198 (1999)
27. Opitz, D., Shavlik, J.W.: Actively searching for an effective neural network ensemble. Connection Science 8(3-4), 337–353 (1996)
28. Quinlan, J.R.: Learning with Continuous Classes. In: Proc. 5th Australian Joint Conference on Artificial Intelligence (AI 1992), Singapore, pp. 343–348 (1992)
29. Rustagi, J.S.: Optimization Techniques in Statistics. Academic Press, London (1994)
30. Schapire, R.E.: The Boosting approach to machine learning: An overview. In: Denison, D.D., et al. (eds.) Nonlinear Estimation and Classification. Springer, Heidelberg (2003)
31. Schapire, R.E.: The strength of weak learnability. Machine Learning 5(2), 197–227 (1990)
32. Skurichina, M., Duin, R.P.W.: Bagging for linear classifiers. Pattern Recognition 31, 909–930 (1998)
33. Triadaphillou, S., et al.: Fermentation process tracking through enhanced spectral calibration modelling. Biotechnology and Bioengineering 97, 554–567 (2007)
34. Zemel, R.S., Pitassi, T.: A gradient based boosting algorithm for regression problems. In: Adv. in Neural Inf. Processing Systems, vol. 13, pp. 696–702 (2001)
35. Zhang, J.: Inferential estimation of polymer quality using bootstrap aggregated neural networks. Neural Networks 12, 927–938 (1999)