International Journal of Soft Computing 2 (1): 80-87, 2007
© Medwell Journals, 2007

# Local Ensembles vs. Global Ensembles

S.B. Kotsiantis and D.N. Kanellopoulos
Educational Software Development Laboratory, Department of Mathematics,
University of Patras, Hellas

**Abstract:** Many data mining problems involve an investigation of relationships between features in heterogeneous datasets, where different learning algorithms can be more appropriate for different regions. We propose the locally application of ensembles' techniques. This methodology identifies local regions having similar characteristics and then uses combining techniques to describe the relationship between the data characteristics and the target value. We performed a comparison of the locally application of the combining techniques with the globally application of the combining techniques, on standard benchmark datasets and the locally application of the ensembles gives more accurate results.

**Key words:** Machine learning, data mining, classification, regression

## INTRODUCTION

When the size of the training set is small compared to the complexity of the learner, the learning algorithm frequently overfits the noise in the training set. Thus effective control of complexity of a learning algorithm plays a basic role in achieving good generalization. Some theoretical results and experimental results[1] indicate that a local learning algorithm provides a practicable solution to this problem. In local learning, each local model is trained independently of all other local models such that the total number of local models in the learning system does not directly affect how complex a function can be learned. This property avoids overfitting if a robust learning scheme exists for training the individual local model.

Local learning[2] is an extension of instance-based learning. Local learners wait until they have seen the test set instances before making a prediction. This allows the learner to make predictions based on specific instances that are most similar to the test set instances. Local learning can be understood as a general theory that allows extending learning algorithms, to the case of complex data for which the algorithm's assumptions would not necessarily hold globally, but can be thought as valid locally. A simple example is the assumption of linear separability, which in general is not satisfied globally in classification problems. Yet any learning algorithm able to find only a linear separation, can be used inside a local learning procedure, yielding an algorithm able to model complex non-linear class boundaries.

In the recent years researchers have continuously argued for the benefits of using multiple models to solve complex problems. The main idea behind combining learners is based on the assumption that different learners using different data representations, different concepts and modeling techniques are most likely to arrive at results with different patterns of generalization.

In this study, we propose the locally application of ensembles' techniques. This methodology identifies local regions having similar characteristics and then uses combining techniques to describe the relationship between the data characteristics and the target class. We performed a comparison of the locally application of the combining techniques with the globally application of the combining techniques, on standard benchmark datasets and the locally application of the ensembles gives better accuracy. For the experiments, a number of combining techniques were used such as bagging, boosting, voting and averaging.

**Ensembles of learners:** Empirical studies showed that ensembles are often much more accurate than the individual base learners that make them up[3] and recently different theoretical explanations have been proposed to justify the effectiveness of some commonly used ensemble methods[4]. Currently, there are two main

**Corresponding Author:** S.B. Kotsiantis, Educational Software Development Laboratory, Department of Mathematics, University of Patras, Hellas

approaches to model combination. The first is to produce a set of learned models by applying an algorithm repeatedly to different training sample data; the second applies various learning algorithms to the same sample data. The predictions of the models are then combined according to an averaging/voting scheme or a stacking algorithm[3].

Starting with bagging[5], we will say that this method samples the training set, generates random independent bootstrap replicates, constructs a learner on each of these and aggregates them by a simple majority vote (for classification problems) or averaging procedure (for regression problems). Therefore, taking a bootstrap replicate one can sometimes avoid or get less misleading training objects in the bootstrap training set. Consequently, a learner constructed on such a training set may have a better performance[6].

Another method that uses different subset of training data with a single learning method is the boosting approach[7]. It assigns weights to the training instances and these weight values are changed depending upon how well the associated training instance is learned by the learner; the weights for misclassified instances are increased. Thus, re-sampling occurs based on how well the training samples are classified by the previous model. Since the training set for one model depends on the previous model, boosting requires sequential runs and thus is not readily adapted to a parallel environment. After several cycles, the prediction is performed by taking a weighted vote of the predictions of each learner, with the weights being proportional to each learner's accuracy on its training set. AdaBoost is a practical version of the boosting approach for classification problems[7].

The AdaBoost.R algorithm[8] attacks the regression problem by reducing it to a classification problem. Friedman has also explored regression using the gradient descent approach[9]. In each iteration, the Additive Regression algorithm constructs goal values for each data-point xi equal to the (negative) gradient of the loss of its current master hypothesis on xi. The base learner then finds a function in a class minimizing the squared error on this constructed sample.

Another approach for building ensembles of classifiers is to use a variety of data mining algorithms on all of the training data and combine their predictions. Among the combination techniques, majority vote is the simplest to implement, since it requires no prior training[10]. If we have a dichotomic classification problem and L hypotheses whose error is lower than 0.5, then the resulting majority voting ensemble has an error lower than the single classifier, as long as the error of the base learners are uncorrelated. A similar approach for building ensembles of regression models is to use a variety of learning algorithms on all of the training data and combine their predictions. Among the combination techniques, averaging is the simplest to implement, since it requires no prior training[11]. Other approaches were based on the linear combination of the base models according to the function: $\sum_{i=1}^{n} a_i f_i(x)$ where $\alpha_i$ are the weight assigned to the base models prediction $f_i(x)$. The simplest approach to determining the values of $\alpha_i$ I is to set them to the same value. This is known as the Base Ensemble Method (BEM)[12]. More advanced approaches try to set the weights so as to minimize the mean square error of the training data.

**Locally application of combining techniques:** The proposed algorithm builds a model for each point to be estimated, taking into account only a subset of the training points. This subset is chosen on the basis of the preferable distance metric between the testing point and the training point in the input space. For each testing point, an ensemble of learners is thus learned using only the training points lying close to the current testing point. Generally, the proposed ensemble consists of the four steps Fig. 1. The proposed ensemble has some free parameters such as the distance metric. In our experiments, we used the most well known-Euclidean similarity function-as distance metric. For two data points, $X = \langle x_1, x_2, x_3, \ldots, x_{n-1} \rangle$ and $Y = \langle y_1, y_2, y_3, \ldots, y_{n-1} \rangle$, the Euclidean similarity function is defined as

$$d2(X,Y) = \sqrt{\sum_{i=1}^{n-1} (x_i - y_i)^2}$$

For the experiments, a number of combining techniques were used such as bagging, boosting, voting and averaging.

The proposed algorithm also requires choosing the value of K. There are several ways to do this. A first, simple solution is to fix K a priori before the beginning of the learning process. However, the best K for a specific dataset is obviously not the best one for another dataset. A second, more time-consuming solution is therefore to determine this best K automatically through the minimization of a cost criterion. One way to do that is to evaluate the estimation error on a test set and thus keep as K the value for which the error is the least. In the current implementation we decided to use a fixed value for K (=50): a) in order to keep the training time low and b) about this size of instances is appropriate for a simple algorithm, to build a precise model according to[13].

- **Determine a suitable distance metric.**
- **Find the k nearest neighbors using the selected distance metric.**
- **Apply combining technique using as training instances the k instances**
- **The answer of the ensemble is the prediction for the testing instance.**

Fig. 1: Local ensemble

Our method shares the properties of other memory-based methods such as no need for training and more computational cost for the prediction. Besides, our method has some desirable properties, such as better accuracy and confidence interval.

## EXPERIMENTS

We performed comparisons of the locally application of the combining techniques with the globally application of the combining techniques on classification and regression problems.

**Using the proposed technique as classification method:** We experimented with 22 datasets from the UCI repository[14]. These datasets cover many different types of classification problems having discrete, continuous and symbolic variables. In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier 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. It must be mentioned that we used the free available source code for most of the algorithms by[15] for our experiments.

**Local voting vs. global voting:** For the first experiment, we compare Local Voting with Global Voting. For the experiment we used the three most common weak machine learning algorithms OneR[15], Decision stump[16] and Naïve Bayes[17]. Each classifier (NB, OneR, DS) generate a hypothesis h1, h2, h3, respectively. The a-posteriori probabilities generated by the individual classifiers are correspondingly denoted p1(i), p2(i), p3(i) for each output class i. Next, the class represented by the maximum sum value of the a-posteriori probabilities is taken as the voting hypothesis (h*). The predictive class is computed by the rule:

$$\text{Predictive class} = \operatorname{argmax} \sum_{i=1,\, j=1}^{i=\text{number of classes},\, j=3} p_j$$

As it is well-known, voting methods fail if the weak learners cannot achieve at least 50% accuracy for the specific dataset. For this reason, the proposed method reduces the multi-class problems to a set of binary problems. We have used the One Per Class (OPC) approach. The ith classifier is trained with all of the examples in the ith class with positive labels and all other examples with negative labels. The final output is the class that corresponds to the classifier with the highest output value.

During the experiment we compared the proposed ensemble with the plain classifier NB, DS, OneR and their local versions using 50 instances as local region as well as the simple voting using the same learning algorithm as base learners. In Table 1, we represent with "v" that the proposed ensemble looses from the specific algorithm. That is, the specific algorithm performed statistically better than the proposed according to t-test with $p<0.05$[18]. Furthermore, "*" indicates that proposed ensemble performed statistically better than the specific classifier 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 row of the table one can also see the aggregated results in the form (a/b/c). In this notation "a" means that the proposed ensemble is significantly less accurate than the compared algorithm in a out of 22 datasets, "c" means that the proposed algorithm is significantly more accurate than the compared algorithm in c out of 22 datasets, while in the remaining cases (b), there is no significant statistical difference.

In the last raw of the Table 1 one can see the aggregated results. The presented ensemble is significantly more accurate than single NB in 7 out of the 22 datasets, while it has significantly higher error rate in 4 datasets. What is more, the proposed ensemble is significantly more accurate than DS and OneR in 14 and 13 out of the 22 datasets, equivalently, whilst it has significantly higher error rate in none dataset. Likewise, the proposed ensemble is significantly more accurate than local DS and OneR in 6 and 4 out of the 22 datasets, equivalently, whilst it has significantly higher error rate in none dataset. Furthermore, the presented ensemble is significantly more accurate than local NB in 3 out of the 22 datasets, whilst it has significantly higher error rate in one dataset. Moreover, the proposed ensemble is significantly more accurate than simple voting in 9 out of the 22 datasets, while on 2 datasets, it hassignificantly higher error rate.

**Local boosting vs. Global boosting for classification problems:** Secondly, we compared the local application of boosting with other methods that are based on the same base learning algorithm-DS:

Table 1: Comparing local voting with the plain classifier NB, DS, OneR, their local versions as well as simple global voting

| Datasets | Local voting | Local NB | Local DS | Local OneR | NB | DS | OneR | Voting |
|---|---|---|---|---|---|---|---|---|
| Autos | 81.07 | 75.93* | 69.86* | 76.49 | 57.41* | 44.9* | 61.77* | 66.45* |
| Breast-cancer | 73.59 | 72.82 | 73.38 | 72.69 | 72.7 | 69.27 | 66.91* | 67.26 |
| Breast-w | 96.45 | 96.32 | 96.22 | 96.27 | 96.07 | 92.33* | 92.01* | 94.48 |
| Colic | 82.39 | 81.77 | 81.3 | 81.17 | 78.7 | 81.52 | 81.52 | 81.52 |
| Credit-rating | 85.39 | 82.87* | 82.35* | 83.59 | 77.86* | 85.51 | 85.51 | 85.51 |
| Diabetes | 71.9 | 71.84 | 72.36 | 69.76* | 75.75v | 71.8 | 71.98 | 72.72 |
| Glass | 75.48 | 72.66 | 69.44* | 68.52* | 49.45* | 44.89* | 56.84* | 60.21* |
| Haberman | 69.89 | 71.3 | 70.65 | 68.88 | 75.06v | 71.57 | 72.53 | 72.56 |
| Heart-c | 80.53 | 81.36 | 78.68 | 79.23 | 83.34 | 72.93* | 72.53* | 73.19 |
| Heart-h | 79.68 | 80.66 | 78.31 | 79.14 | 83.95 | 81.78 | 80.69 | 81.58v |
| Heart-statlog | 78.85 | 80.41 | 74.63 | 76.3 | 83.59v | 72.3* | 71.26* | 72.67v |
| Hepatitis | 84.71 | 86.18 | 83.76 | 82.02 | 83.81 | 77.62* | 82.05 | 82.18 |
| Ionosphere | 88.24 | 81.91* | 87.56 | 88.24 | 82.17* | 82.57* | 82.59* | 90.03* |
| Iris | 94.53 | 95.67 | 93.8 | 94 | 95.53 | 66.67* | 93.53 | 93.53 |
| Lymphotherapy | 82 | 82.95 | 75.61* | 79.6 | 83.13 | 75.31 | 74.77 | 75.91 |
| Monk3 | 92.4 | 91.66 | 92.64 | 92.4 | 93.45 | 76.01* | 77.88* | 78.45* |
| Primary-tumor | 44.9 | 44.46 | 43.28 | 43.8 | 49.71v | 28.91* | 27.74* | 28.21* |
| Sonar | 82.45 | 86.6 | 77.81* | 73.57* | 67.71* | 72.25* | 62.12* | 68.98* |
| Titanic | 78.95 | 78.95 | 78.94 | 78.95 | 77.85 | 77.6* | 77.6* | 77.6* |
| Vehicle | 72.01 | 74.94v | 69.12* | 67.37* | 44.68* | 39.81* | 52.36* | 46.37* |
| Vote | 96 | 95.49 | 95.88 | 95.47 | 90.02* | 95.63 | 95.63 | 95.63* |
| Wine | 97.07 | 98.92 | 95.67 | 94.83 | 97.46 | 57.91* | 77.93* | 86.02 |
| Average Accuracy | 81.29 | 81.16 | 79.14 | 79.19 | 77.24 | 69.95 | 73.53 | 75.04 |
| *W/D/L* | | 1/18/3 | 0/16/6 | 0/18/4 | 4/11/7 | 0/8/14 | 0/9/13 | 2/10/9 |

- Simple DS algorithm
- Local weighted DS using 50 instances
- Global Boosting DS (using 25 sub-classifiers)

In the last raw of the Table 2 one can see the aggregated results. The presented ensemble is significantly more accurate than single DS in 14 out of the 22 datasets, while it has significantly higher error rate in none dataset. Likewise, the proposed ensemble is significantly more accurate than local weighted DS in 2 out of the 22 datasets, whilst it has significantly higher error rate in none dataset, respectively. Adaboost DS has significantly lower error rates in 1 out of the 22 datasets than local boosting, whereas it is significantly less accurate in 6 datasets.

**Local bagging vs. global bagging for classification problems:** Thirdly, we compared the local application of bagging with other methods that are based on the same learning algorithm-DS:

- Simple DS algorithm
- Local weighted DS using 50 local instances. This method differs from the proposed technique since it has no bagging process.
- Global Bagging DS (using 25 sub-classifiers).

In the last raw of the Table 3 one can see the aggregated results. The proposed ensemble is significantly more accurate than simple Bagging DS in 11 out of the 22 datasets, whilst it has significantly higher error rate in none dataset. In addition, the presented ensemble is significantly more accurate than single Local DS in 2 out of the 22 datasets, while it has significantly higher error rate in none dataset. The presented ensemble is significantly more accurate than single DS in 14 out of the 22 datasets, while it has significantly higher error rate in none dataset.

**Using the proposed technique as regression method:** We experimented with 16 datasets from the UCI repository[14]. The most well known measure for the degree of fit for a regression model to a dataset is the correlation coefficient. In order to calculate the regression models' correlation coefficient, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the regression model 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. It must be mentioned that we used the free available source code for most of the algorithms by[15] for our experiments.

**Local averaging vs. global averaging:** For the comparison of local averaging with the global averaging we used the three most common algorithms Linear Regression (LR)[12], RepTree[15] and Decision Table (DT)[19] as learners. These weak regression models combine rapid learning processes and acceptable performance. During the experiment we also compared the proposed ensemble with the plain regression models LR, RepTree, DT and their local versions using 50 instances as local region. In Table 4, we

Table 2: Comparing Local Boosting DS for classification problems

| Datasets | Local boost DS | Local DS | DS | Boost DS |
|---|---|---|---|---|
| Autos | 75.49 | 74.82 | 44.9* | 44.9* |
| Breast-cancer | 72.75 | 72.68 | 69.27 | 71.55 |
| Breast-w | 95.9 | 96.4 | 92.33* | 95.28 |
| Colic | 78.91 | 80.87 | 81.52 | 82.72 |
| Credit-rating | 84.94 | 83.61 | 85.51 | 85.57 |
| Diabetes | 73.77 | 73.2 | 71.8 | 75.37 |
| Glass | 70.39 | 70.58 | 44.89* | 44.89* |
| Haberman | 68.9 | 69.81 | 71.57 | 74.06v |
| Heart-c | 80.4 | 78.29 | 72.93* | 83.11 |
| Heart-h | 79.06 | 79.17 | 81.78 | 82.42 |
| Heart-statlog | 78.15 | 76.33 | 72.3 | 81.81 |
| Hepatitis | 84.45 | 83.04 | 77.62* | 81.5 |
| Ionosphere | 90.01 | 88.24 | 82.57* | 92.34 |
| Iris | 94.47 | 94 | 66.67* | 95.07 |
| lymphography | 84.6 | 76.67* | 75.31* | 75.44* |
| Monk3 | 90.68 | 93.44 | 76.01* | 90.92 |
| Primary-tumor | 43.22 | 43.22 | 28.91* | 28.91* |
| Sonar | 83.89 | 76.62* | 72.25* | 81.06 |
| Titanic | 79.05 | 79.05 | 77.6* | 77.83 |
| Vehicle | 70.98 | 69.58 | 39.81* | 39.81* |
| Vote | 96.02 | 95.4 | 95.63 | 96.41 |
| Wine | 97.47 | 96.79 | 57.91* | 91.57* |
| *Average accuracy* | *80.61* | *79.62* | *69.95* | *76.02* |
| *W/D/L* | | *0/20/2* | *0/8/14* | *1/15/6* |

Table 3: Comparing Local Bagging DS for classification problems

| Datasets | Local bagging DS | Local DS | DS | Bagging DS |
|---|---|---|---|---|
| Autos | 76.37 | 74.82 | 44.9* | 44.95* |
| Breast-cancer | 73.66 | 72.68 | 69.27 | 73.38 |
| Breast-w | 96.45 | 96.4 | 92.33* | 92.56* |
| Colic | 81.77 | 80.87 | 81.52 | 81.52 |
| Credit-rating | 85.01 | 83.61 | 85.51 | 85.51 |
| Diabetes | 74.54 | 73.2 | 71.8 | 72.45 |
| Glass | 70.96 | 70.58 | 44.89* | 45.08* |
| Haberman | 71.18 | 69.81 | 71.57 | 73.07 |
| Heart-c | 80.97 | 78.29 | 72.93* | 75.26 |
| Heart-h | 79.69 | 79.17 | 81.78 | 81.41 |
| Heart-statlog | 78.07 | 76.33 | 72.3* | 75.33 |
| Hepatitis | 84.59 | 83.04 | 77.62* | 80.61 |
| Ionosphere | 88.89 | 88.24 | 82.57* | 82.66* |
| Iris | 93.87 | 94 | 66.67* | 68.87* |
| lymphography | 80.78 | 76.67* | 75.31* | 74.5 |
| Monk3 | 93.45 | 93.44 | 76.01* | 82.41* |
| Primary-tumor | 43.98 | 43.22 | 28.91* | 28.91* |
| Sonar | 82.21 | 76.62* | 72.25* | 73.21* |
| Titanic | 78.99 | 79.05 | 77.6 | 77.6* |
| Vehicle | 71.47 | 69.58 | 39.81* | 40.14* |
| Vote | 95.72 | 95.4 | 95.63 | 95.63 |
| Wine | 97.8 | 96.79 | 57.91* | 86.27* |
| Average accuracy | 80.92 | 79.62 | 69.95 | 72.33 |
| W/D/L | | 0/20/2 | 0/8/14 | 0/11/11 |

represent as "v" that the specific algorithm performed statistically better than the proposed ensemble according to t-test with $p<0.05$. On the other hand, "*" indicates that proposed ensemble performed statistically better than the specific algorithm 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 row of the Table 4 one can see the aggregated results. The presented ensemble has significantly higher correlation coefficient than single LR in 9 out of the 16 datasets, while it has significantly lower correlation coefficient in 4 datasets. What is more, the proposed ensemble has significantly higher correlation coefficient than RepTree and DT in 14 and 12 out of the 16 datasets, equivalently, whilst it has significantly lower correlation coefficient in one dataset. Likewise, the proposed ensemble has significantly higher correlation coefficient than local RepTree and DT in 10 and 12 out of the 16 datasets, equivalently, whilst it has significantly lower correlation coefficient in none dataset. Furthermore, the presented ensemble has significantly higher correlation coefficient thanlocal LR in 6 out of the 16

Table 4: Comparing local averaging with the plain regression model DT, Rep tree, LR, their local versions as well as global averaging

| Datasets | Local averaging | Local rep tree | Local DT | Local LR | LR | Rep tree | DT | Averaging |
|---|---|---|---|---|---|---|---|---|
| Auto93 | 0.79 | 0.71* | 0.76 | 0.62* | 0.83v | 0.23* | 0.68* | 0.83v |
| AutoHorse | 0.93 | 0.92 | 0.93 | 0.83* | 0.95v | 0.83* | 0.85* | 0.93 |
| AutoMpg | 0.92 | 0.90* | 0.88* | 0.91 | 0.93 | 0.88* | 0.90* | 0.93 |
| AutoPrice | 0.93 | 0.90* | 0.90* | 0.93 | 0.89* | 0.88* | 0.81* | 0.91* |
| Bodyfat | 0.99 | 0.98* | 0.98* | 0.99 | 0.99 | 0.98* | 0.97* | 0.98 |
| Cleveland | 0.65 | 0.65 | 0.55* | 0.58* | 0.71v | 0.54* | 0.52* | 0.66 |
| Cpu | 0.98 | 0.93* | 0.93* | 0.99v | 0.95* | 0.90* | 0.92* | 0.97* |
| Fishcatch | 0.98 | 0.96* | 0.97* | 0.98 | 0.97* | 0.95* | 0.94* | 0.98 |
| Housing | 0.92 | 0.87* | 0.87* | 0.94v | 0.85* | 0.85* | 0.81* | 0.89* |
| Lowbwt | 0.76 | 0.76 | 0.74* | 0.73* | 0.79v | 0.78v | 0.78v | 0.79v |
| PwLinear | 0.90 | 0.82* | 0.85* | 0.93v | 0.86* | 0.89 | 0.83* | 0.91 |
| Quake | 0.10 | 0.09 | 0.05* | 0.10 | 0.06* | 0.07* | 0.09 | 0.10 |
| Servo | 0.93 | 0.86* | 0.93 | 0.93 | 0.85* | 0.86* | 0.80* | 0.90* |
| Stock | 0.99 | 0.99 | 0.99 | 0.99 | 0.93* | 0.98* | 0.97* | 0.98* |
| Triazines | 0.45 | 0.41 | 0.39* | 0.34* | 0.39* | 0.27* | 0.47 | 0.54v |
| Veteran | 0.45 | 0.35* | 0.31* | 0.41* | 0.48 | 0.23* | 0.41 | 0.47 |
| Average correlation coefficient | 0.79 | 0.75 | 0.75 | 0.76 | 0.77 | 0.6 | 0.73 | 0.79 |
| *W/D/L* | | 0/6/10 | 0/4/12 | 3/7/6 | 4/3/9 | 1/1/14 | 1/3/12 | 3/8/5 |

Table 5: Comparing Local Additive Regression DS for regression problems

| Dataset | Local additive regression DS | Local DS | Additive regression DS | DS |
|---|---|---|---|---|
| Auto93 | 0.77 | 0.72* | 0.77 | 0.59* |
| AutoHorse | 0.92 | 0.92 | 0.86* | 0.72* |
| AutoMpg | 0.89 | 0.89 | 0.87* | 0.74* |
| AutoPrice | 0.92 | 0.89* | 0.90* | 0.81* |
| Bodyfat | 0.95 | 0.94* | 0.95 | 0.82* |
| Cleveland | 0.57 | 0.63v | 0.65v | 0.40* |
| Cpu | 0.96 | 0.92* | 0.95* | 0.31* |
| Fishcatch | 0.96 | 0.94* | 0.94* | 0.83* |
| Housing | 0.87 | 0.84* | 0.84* | 0.60* |
| Lowbwt | 0.73 | 0.78v | 0.78v | 0.78v |
| PwLinear | 0.89 | 0.84* | 0.83* | 0.68* |
| Quake | 0.10 | 0.09 | 0.10 | 0.09 |
| Servo | 0.93 | 0.89* | 0.85* | 0.79* |
| Stock | 0.99 | 0.99 | 0.94* | 0.78* |
| Triazines | 0.51 | 0.47* | 0.37* | 0.04* |
| Veteran | 0.35 | 0.28* | 0.40v | 0.15* |
| Average correlation coefficient | 0.77 | 0.75 | 0.75 | 0.57 |
| *W-D-L* | | *2/4/10* | *3/3/10* | *1/1/14* |

datasets, whilst it has significantly lower correlation coefficient in 3 datasets. Moreover, the proposed ensemble has significantly higher correlation coefficient than simple averaging in 5 out of the 16 datasets, while it has significantly lower correlation coefficient on 3 datasets.

**Local boosting vs. global boosting for regression problems:** We compared the local application of boosting with other methods that are based on the same learning algorithm-DS:

- Simple DS algorithm
- Local DS using 50 local instances. This method differs from the proposed technique since it has no boosting process.
- Additive regression DS (using 10 sub-models).

In the last raw of the Table 5 one can see the aggregated results. The proposed ensemble is significantly more accurate than simple DS in 14 out of the 16 datasets, whilst it has significantly lower correlation coefficient in one dataset. In addition, the presented ensemble is significantly more accurate than Local DS in 10 out of the 16 datasets, while it has significantly lower correlation coefficient in 2 datasets. Furthermore, global additive regression DS have significantly higher correlation coefficient in 3 datasets than the proposed ensemble, whereas it is significantly less accurate in 10 datasets.

**Local bagging vs. Global bagging for regression problems:** We compared the local application of bagging with other methods that are based on the same learning algorithm-DS:

Table 6: Comparing Local bagging DS for regression problems

| Dataset | Local bagging DS | Local DS | Bagging DS | DS |
|---|---|---|---|---|
| Auto93 | 0.82 | 0.72* | 0.74* | 0.59* |
| AutoHorse | 0.94 | 0.92 | 0.80* | 0.72* |
| AutoMpg | 0.92 | 0.89 | 0.78* | 0.74* |
| AutoPrice | 0.91 | 0.89 | 0.82* | 0.81* |
| Bodyfat | 0.96 | 0.94 | 0.84* | 0.82* |
| Cleveland | 0.68 | 0.63* | 0.61* | 0.40* |
| Cpu | 0.93 | 0.92 | 0.87* | 0.31* |
| Fishcatch | 0.95 | 0.94 | 0.85* | 0.83* |
| Housing | 0.88 | 0.84* | 0.74* | 0.60* |
| Lowbwt | 0.79 | 0.78 | 0.78 | 0.78 |
| PwLinear | 0.86 | 0.84 | 0.68* | 0.68* |
| Quake | 0.11 | 0.09 | 0.09 | 0.09 |
| Servo | 0.89 | 0.89 | 0.79* | 0.79* |
| Stock | 0.99 | 0.99 | 0.79* | 0.78* |
| Triazines | 0.51 | 0.47* | 0.25* | 0.04* |
| Veteran | 0.38 | 0.28* | 0.33* | 0.15* |
| Average correlation coefficient | 0.78 | 0.75 | 0.67 | 0.57 |
| *W-D-L* | | *0/11/5* | *0/2/14* | *0/2/14* |

- Simple DS algorithm
- Local DS using 50 local instances. This method differs from the proposed technique since it has no bagging process.
- Bagging DS (using 25 sub-models).

In the last raw of the Table 6 one can see the aggregated results. The proposed ensemble is significantly more accurate than simple DS in 14 out of the 16 datasets, whilst it has significantly lower correlation coefficient in one dataset. In addition, the presented ensemble is significantly more accurate than Local DS in 5 out of the 16 datasets, while it has significantly lower correlation coefficient in none dataset. Furthermore, global bagging DS have significantly higher correlation coefficient in none dataset than the proposed ensemble, whereas it is significantly less accurate in 14 datasets.

## CONCLUSION

Local algorithms defer processing of the dataset until they receive request for information (e.g. classification or prediction of target value). A database of observed input-output data is always kept and the estimate for a new operating point is derived from an interpolation based on a neighborhood of the query point. Local techniques are an old idea in time series prediction[14].

Lazy learners are particularly useful for prediction on data streams. In data streams, new data keep arriving, so building a new learner each time can be very expensive. In addition, the multidimensional data is sometimes feature-space heterogeneous so that different features have different importance in different sub-areas of the whole space.

Local learning can reduce the complexity of component learners and improve the generalization performance although the global complexity of the system can not be guaranteed to be low. In this study, we propose the locally application of ensembles' techniques. We performed a comparison of the locally application of the combining techniques with the globally application of the combining techniques, on standard benchmark datasets and the locally application of the ensembles gave better accuracy. Due to the encouraging results obtained from our experiments, we can expect that the proposed combining method can be successfully applied to the classification and regression task in the real world case with more accurate results than the traditional data mining approaches.

The benefit of allowing multiple local models is somewhat offset by the cost of storing and querying the training dataset for each test set example which means that lazy learners do not scale well for the large amount of data associated with several applications. Local weighted learning algorithms must often decide what instances to store for use during generalization in order to avoid excessive storage and time complexity. By removing a set of instances from a database the response time for the predictions will decrease, as fewer instances are examined when a query instance is presented. This objective is primary when we are working with large databases and have limited storage.

In a following work we will focus on the problem of reducing the size of the stored set of instances while trying to maintain or even improve generalization accuracy by avoiding noise and overfitting. In[20,21] can be found numerous instance selection methods that can be combined with local boosting technique.

**REFERENCES**

1. Vapnik, V.N., 1998. Statistical Learning Theory, Wiley, New York.
2. Atkeson, C.G., A. W. Moore and S. Schaal, 1997. Locally weighted learning for control. Artificial Intelligence Review, 11: 75-113.
3. Dietterich, T.G., 2001. Ensemble Methods in Machine Learning. In Kittler, J., Roli, F., (Eds.) Multiple Classifier Systems. LNCS, pp: 1-15.
4. Kittler, J., M. Hatef, R.P.W. Duin and J. Matias, 1998. On combining classifiers, IEEE T-PAMI, pp: 226-239.
5. Breiman, L., 1996. Bagging Predictors. Machine Learning, Kluwer Academic Publishers, 24: 123-140.
6. Breiman, L., 2001. Using iterated bagging to debias regressions, Machine Learning, pp: 261-277.
7. Freund, Y. and R. Schapire, 1996. Experiments with a New Boosting Algorithm, Proceedings: ICML, pp: 148-156.
8. Duffy, N. and D. Helmbold, 2002. Boosting Methods for Regression, Machine Learning, 3 : 153-200.
9. Friedman, J., 1992. Stochastic Gradient Boosting, Computational Statistics and Data Analysis W. Iba and P. Langley (Eds.), Induction of one-level Decision Trees. Proc. of the Ninth International Machine Learning Conference. Aberdeen, Scotland: Morgan Kaufmann, 38: 367-378.
10. Van Erp, M., L.G. Vuurpijl and L.R.B. Schomaker, 2002. An Overview and Comparison of Voting Methods for Pattern Recognition. In Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition. Niagara-on-the-Lake, Canada, pp: 195-200.
11. Brown, G., J. Wyatt and P. Tino, 2005. Managing diversity in regression ensembles. J. Machine Learning Res., pp: 6.
12. Fox, J., 1997. Applied Regression Analysis, Linear Models and Related Methods, ISBN: 080394540X, Sage Pubns.
13. Frank, E., M. Hall and B. Pfahringer, 2003. Locally Weighted Naive Bayes. Proc. of the 19th Conference on Uncertainty in Artificial Intelligence. Acapulco, Mexico. Morgan Kaufmann.
14. Blake, C. and C. Merz, 1998. UCI Repository of machine learning databases. Irvine, CA: Uni-versity of California, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.h tml].
15. Witten, I. and E. Frank, 2000. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, CA.
16. Iba, W. and P. Langley, 1992. Induction of one-level Decision Trees. Proc. of the Ninth International Machine Learning Conference. Aberdeen, Scotland: Morgan Kaufmann.
17. Domingos, P. and M. Pazzani, 1997. On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning, 29: 103-130.
18. Nadeau, C. and Y. Bengio, 2003. Inference for the Generalization Error. Machine Learning 52: 239-281.
19. Kohavi, R., 1995. The Power of Decision Tables. In Proc European Conference on Machine Learning.
20. Brighton, H. and C. Mellish, 2002. Advances in instance selection for instance-based learning algorithms, Data Mining and Knowledge Discovery, 6: 153-172.
21. Wilson, D. and T. Martinez, 2000. Reduction Techniques for Instance-Based Learning Algorithms, Machine Learning, 38: 257-286.