LOCAL BOOSTING OF WEAK CLASSIFIERS

S. B. Kotsiantis
Educational Software Development Laboratory
Department of Mathematics
University of Patras
Greece
sotos@math.upatras.gr

P. E. Pintelas
Educational Software Development Laboratory
Department of Mathematics
University of Patras
Greece
pintelas@math.upatras.gr

Abstract. **Many data analysis problems involve an investigation of relationships between attributes in heterogeneous databases, where different prediction models can be more appropriate for different regions. We propose a technique of boosting localized weak classifiers; rather than having constant weights attached to each of the classifiers (as in standard boosting approaches), we allow weights to be functions over the input domain. In order to determine these functions, we identify local regions having similar characteristics and then build local classification experts on each of these regions describing the relationship between the data characteristics and the target class. The idea of local boosting is that although no single function works well globally, in any local region a function should be capable of doing the classification. We performed a comparison with other well known combining methods using weak classifiers as based learners, on standard benchmark datasets and we took better accuracy.**

I. INTRODUCTION

Instance-based learners classify an instance by comparing it to a database of pre-classified examples. The fundamental assumption is that similar instances will have similar classifications. The corresponding components of an instance-based learner are the distance function which determines how similar two instances are, and the classification function which specifies how instance similarities yield a final classification for the new instance [1].

The nearest neighbor rule becomes less appealing with finite training samples. This is due to the curse of dimensionality. Severe bias can be introduced in the nearest neighbor rule in a high dimensional input feature space with finite samples. As such, the choice of a distance measure becomes crucial in determining the outcome of nearest neighbor classification. The commonly used Euclidean distance measure, while simple computationally, implies that the input space is isotropic or homogeneous. However, the assumption for isotropy is often invalid and generally undesirable in many practical applications [1].

Local learning [2] can be understood as a general principle that allows to extend learning techniques designed for simple models, to the case of complex data for which the model'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 with rich data. Yet any classification 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.

We propose a technique of boosting localized weak classifiers. Usual boosting algorithms are well known to be sensitive to noise [5]. In the case of local boosting, the algorithm should handle reasonable noise, and be at least as good as boosting, if not better. For the experiments, two weak algorithms of two well known machine learning techniques: decision trees and rule learners, were used. We performed a comparison with other well known combining methods, on standard benchmark datasets and we took better accuracy for local boosting.

In the next section, we discuss the localized experts, while current ensemble approaches and work are described in section 3. In Section 4 we describe the proposed method and investigate its advantages and limitations. In Section 4, we evaluate the proposed method on several UCI datasets by comparing it with standard boosting and other lazy methods. Finally, section 5 concludes the paper and suggests further directions in current research.

II. LOCAL WEIGHTED LEARNING

When all training examples are considered when classifying a new test case, the algorithm works as a global method, while when the nearest training examples are considered, the algorithm works as a local method, since only data local to the area around the testing instance contribute to the class probabilities. Local methods have significant advantages when the probability measure defined on the space of symbolic objects for each class is very complex, but can still be described by a collection of less complex local approximations.

When the size of the training set is small compared to the complexity of the classifier, the learning machine usually overfits the noise in the training data. Thus effective control of complexity of a classifier plays a key role in achieving good generalization. Some theoretical results and experimental results [18] indicate that a local learning algorithm (that is learning machine trained on the training subset) provides a feasible solution to this problem.

Learning based on the training subset has been an exciting research topic and some important theoretical and experimental results have been obtained. In fact, local learning is not a new concept and it has appeared in the early years of pattern recognition. The obvious example is the k-nearest neighbor method: given a testing pattern, we estimate its class from the closest pattern in the training set.

A list of objections to k-nearest neighbor algorithms includes the following: a) voting used to combine the classes of the nearest k instances, b) uniform neighborhood shape

(spherical) regardless of instance location and c) uniform weight given to all features, instances and neighbors

Our ultimate goal is not to improve the nearest neighbor algorithm, but to improve classification accuracy by combining local classifiers. The authors of [7] proposed a theoretical model of a local learning algorithm and obtained bounds for the local risk minimization estimator for pattern recognition and regression problems using structural risk minimization principle.

In local learning, each local model is trained entirely 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 - complexity can only be controlled by the level of adaptability of each local model. This property avoids overfitting if a robust learning scheme exists for training the individual local model.

The authors of [9] extended the idea of constructing local simple base learners for different regions of input space, searching for appropriate architectures that should be locally used and for a criterion to select a proper unit for each region of input space. They proposed a hybrid MLP/RBF network by combining RBF and Perceptron units in the same hidden layer and using a forward selection to add units until an error goal is reached. Although the resulting Hybrid Perceptron/Radial Network is not in a strict sense an ensemble, the way by which the regions of the input space and the computational units are selected and tested could be in principle extended to ensembles of learning machines.

III. ENSEMBLES OF CLASSIFIERS

Empirical studies showed that classification problem ensembles are often much more accurate than the individual base learner that make them up [5], and recently different theoretical explanations have been proposed to justify the effectiveness of some commonly used ensemble methods [15]. In this work we propose a generative combining method and for this reason this section presents the most well-known generative methods for building ensembles of classifiers in the literature. Generative ensemble methods generate sets of base learners acting on the base learning algorithm or on the structure of the dataset and try to actively improve diversity and accuracy of the base learners.

Starting with bagging [8], we will say that this method samples the training set, generating random independent bootstrap replicates, constructs the classifier on each of these, and aggregates them by a simple majority vote in the final decision rule. Therefore, taking a bootstrap replicate one can sometimes avoid or get less misleading training objects in the bootstrap training set. Consequently, a classifier constructed on such a training set may have a better performance.

Another method that uses different subset of training data with a single learning method is the boosting approach [12]. 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 classifier; 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 classifier, with the weights being proportional to each classifier's accuracy on its training set.

AdaBoost is a practical version of the boosting approach [12]. There are two ways that Adaboost can use these weights to construct a new training set to give to the base learning algorithm. In boosting by sampling, examples are drawn with replacement with probability proportional to their weights. The second method, boosting by weighting, can be used with base learning algorithms that can accept a weighted training set directly. With such algorithms, the entire training set (with associated weights) is given to the base-learning algorithm.

MultiBoosting [19] is another method of the same category that can be considered as wagging committees formed by AdaBoost. Wagging is a variant of bagging; bagging uses resampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses reweighting for each training example, pursuing the effect of bagging in a different way.

In [17] another meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) is presented that uses a learner (one that provides high accuracy on the training data) to build a diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that disagree with the current decision of the committee, thereby directly increasing diversity when a new classifier is trained on the augmented data and added to the committee.

IV. PROPOSED ALGORITHM

It is known that boosting is an effective technique for improving prediction accuracy in many real life datasets [5]. However, previous researches indicated that in heterogeneous databases, where several homogeneous regions exist, boosting does not enhance the prediction capabilities as well as for homogeneous databases [16]. In such cases our experiments indicate that it is more useful to have several local experts responsible for each region of the dataset.

Local learning typically depends on the notion of "neighborhood". The neighborhood can be based on some a-priori measure of locality such as the Euclidean distance in input space.

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, a boosting ensemble of a weak classifier is thus learned using only the training points lying close to the current testing point.

Generally, the proposed ensemble consists of the four steps (see Fig 1)

1) Determine a suitable distance metric.
2) Find the k nearest neighbors using the selected distance metric.
3) Apply boosting to the used simple classifier using as training instances the k instances
4) The answer of the boosting ensemble is the prediction for the testing instance.

Fig. 1. Local Boosting ensemble

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, $X = <x_1, x_2, x_3, \ldots, x_{n-1}>$ and $Y = <y_1, y_2, y_3, \ldots, y_{n-1}>$, the Euclidean similarity function is defined as

$$d_2(X,Y) = \sqrt{\sum_{i=1}^{n-1}(x_i - y_i)^2}\ .$$

We also used k=50 since about this size of instances is appropriate for a simple algorithm to built a precise model [11]. The performance of ADABoost.M1 has been shown to exceed or meet that of various other boosting algorithms [12], thus making it a good choice for this research. We used 10 iterations for the boosting process in order to reduce the time need for classification of a new instance.

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

V. EXPERIMENTS

We have experimented with 27 datasets from the UCI repository [4]. These datasets cover many different types of problems having discrete, continuous, and symbolic variables. Some datasets have missing values, and some have a mixture of all the above. The datasets are listed in Table 1.

TABLE 1. The used datasets

Datasets	Instances	Categ. features	Numer. features	Classes
audiology	226	69	0	24
autos	205	10	15	6
Balance	625	0	4	3
breast-cancer	286	9	0	2
breast-w	699	0	9	2
Colic	368	15	7	2
Credit-rating	690	9	6	2
Diabetes	768	0	8	2
glass	214	0	9	6
haberman	306	0	3	2
heart-c	303	7	6	5
heart-h	294	7	6	5
heart-statlog	270	0	13	2
hepatitis	155	13	6	2
ionosphere	351	34	0	2
iris	150	0	4	3
labor	57	8	8	2
lympho-therapy	148	15	3	4
monk1	124	6	0	2
monk2	169	6	0	2
monk3	122	6	0	2
primary-tumor	339	17	0	21
sonar	208	0	60	2
titanic	2201	3	0	2
vehicle	846	0	18	4
vote	435	16	0	2
wine	178	0	13	3

For the experiments we used the two most common weak machine learning algorithms OneR [13] and Decision stump [14]. 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 median 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 [21] for our experiments.

A. Using DS as base classifier

Decision stump (DS) are one level decision trees that classify instances by sorting them based on feature values [14]. Each node in a decision stump represents a feature in an instance to be classified, and each branch represents a value that the node can take. Instances are classified starting at the root node and sorting them based on their feature values. At worst a decision stump will reproduce the most common sense baseline, and may do better if the selected feature is particularly informative.

Subsequently, we compare the proposed ensemble methodology with:

- K-nearest neighbors using k=3 (most common used number of neighbors), as well as k=50 because the proposed algorithm uses 50 neighbors. In addition, we tested Kstar: another instance-based learner which uses entropy as distance measure [10].
- Local weighted DS using 50 instances
- Bagging DS, Boosting DS and MultiBoost DS (using 25 sub-classifiers)
- DECORATE DS

In the following tables, 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 $p<0.05$. 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 rows of the tables 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 27 datasets, "c" means that the proposed algorithm is significantly more accurate than the compared algorithm in c out of 27 datasets, while in the remaining cases (b), there is no significant statistical difference between the results.

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 18 out of the 27 datasets, while it has significantly higher error rate in none dataset. What is more, the proposed ensemble is significantly more accurate than 3NN and 50NN in 4 and 11 out of the 27 datasets, respectively, whilst it has significantly higher error rate in one and 5 datasets. Likewise, the proposed ensemble is significantly more accurate than local weighted DS and Kstar in 3 and 6 out of the 27 datasets, whilst it has significantly

higher error rate in none and two datasets, respectively. In addition, local boosting is significantly more accurate than Bagging DS in 14 out of the 27 datasets, whilst it has significantly higher error rate in none dataset. Furthermore, Multiboost DS and Decorate DS have significantly lower error rates in 2 and none out of the 27 datasets than the proposed ensemble, respectively whereas they are significantly less accurate in 11 datasets. Adaboost DS has significantly lower error rates in 1 out of the 27 datasets than local boosting, whereas it is significantly less accurate in 9 datasets.

TABLE 2. Comparing Local boosting DS with instance based classifiers and other ensembles that use DS as base learner

Datasets	Local Boost DS	Kstar	3NN	Local DS	DS	50NN	Boost DS	Bagging DS	Multiboost DS	Decorate DS
audiology	73.12	80.32 v	67.97 *	72.68	46.46 *	35.95 *	46.46 *	46.46 *	46.46 *	46.46 *
autos	75.49	72.01	67.23 *	74.82	44.9 *	48.18 *	44.9 *	44.95 *	44.9 *	51.81 *
balance-scale	84.82	88.72 v	86.74 v	84.16	56.72 *	89.01 v	71.77 *	68.21 *	71.77 *	81.25 *
breast-cancer	72.75	73.73	73.13	72.68	69.27	70.75	71.55	73.38	71.76	75.18
breast-w	95.9	95.35	96.61	96.4	92.33 *	95.9	95.28	92.56 *	95.05	95.78
colic	78.91	75.71	80.95	80.87	81.52	84.04 v	82.72	81.52	82.9	82.03
credit-rating	84.94	79.1 *	84.96	83.61	85.51	86.16	85.57	85.51	85.39	85.28
diabetes	73.77	70.19 *	73.86	73.2	71.8	74.68	75.37	72.45	75.19	76.09
Glass	70.39	75.31	70.02	70.58	44.89 *	56.16 *	44.89 *	45.08 *	44.89 *	53.12 *
haberman	68.9	70.27	69.77	69.81	71.57	72.91	74.06 v	73.07	73.8 v	71.61
heart-c	80.4	75.18 *	81.82	78.29	72.93 *	81.58	83.11	75.26	83.54	72.43 *
heart-h	79.06	77.83	82.33	79.17	81.78	83.98 v	82.42	81.41	81.91	81.78
heart-statlog	78.15	76.44	79.11	76.33	72.3	83.74 v	81.81	75.33	82.89 v	81.48
hepatitis	84.45	80.17	80.85	83.04	77.62 *	79.38	81.5	80.61	82.21	80.02
ionosphere	90.01	84.64 *	86.02 *	88.24	82.57 *	71.65 *	92.34	82.66 *	90	90.4
Iris	94.47	94.67	95.2	94	66.67 *	90.53	95.07	68.87 *	94.73	93.93
Labor	90.47	92.03	92.83	85.3	78.77 *	64.67 *	90.57	81.97	89.97	91.07
lymphography	84.6	85.08	81.74	76.67 *	75.31 *	80.59	75.44 *	74.5 *	74.96 *	72.25 *
monk1	86.38	80.27	78.97	77.22 *	73.41 *	59.8 *	69.79 *	73.41 *	70.37 *	70.94 *
monk2	54.08	58.35	54.74	48.75	59.58	62.13 v	53.99	61.13	54.19	61.95
monk3	90.68	86.22	86.72	93.44	76.01 *	82.46 *	90.92	82.41	92.3	93.45
Primary-tumor	43.22	38.02	44.98	43.22	28.91 *	39.26	28.91 *	28.91 *	28.91 *	29.09 *
sonar	83.89	85.11	83.76	76.62 *	72.25 *	68.25 *	81.06	73.21 *	77.54 *	72.91 *
Titanic	79.05	77.56 *	78.9	79.05	77.6 *	77.56 *	77.83	77.6 *	77.62 *	77.6 *
Vehicle	70.98	70.22	70.21	69.58	39.81 *	63.47 *	39.81 *	40.14 *	39.81 *	47.35 *
Vote	96.02	93.22 *	93.08 *	95.4	95.63	90.41 *	96.41	95.63	95.63	95.59
Wine	97.47	98.72	95.85	96.79	57.91 *	96.46	91.57 *	86.27 *	91.22 *	96.45
W/D/L		2/19/6	1/22/4	0/24/3	0/9/18	5/11/11	1/17/9	0/13/14	2/14/11	0/16/11
Average accuracy	80,09	79,05	79,20	78,52	68,67	73,69	74,26	71,20	74,07	75,09

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the DS algorithm. The proposed ensemble can achieve an increase in classification accuracy about 17% compared to simple DS. The average relative accuracy improvement of the proposed methodology is from 1.5% to 12.5% in relation to the remaining methods. This indicates that it is possible to obtain a feasible solution to problems of pattern classification in the real world by local learning because approximating global target function is hard given that usually not enough training samples are available.

B. Using ONER as base classifier

OneR [13] is a simple classifier that extracts a set of rules based upon a single attribute. OneR shows that it is easy to get reasonable performance on a variety of classification problems by examining only one attribute.

Subsequently, we compare the proposed ensemble methodology with:

- K-nearest neighbors using k= 3, 50 as well Kstar: another instance-based learner which uses entropy as distance measure [10].

- Local weighted OneR using 50 instances
- Bagging OneR, Boosting OneR and MultiBoost OneR (using 25 sub-classifiers)
- DECORATE OneR

In the last raw of the Table 3 one can see the aggregated results.

TABLE 3. Comparing Local boosting OneR with instance based classifiers and other ensembles that use OneR as base learner

Datasets	Local Boost OneR	Kstar	3NN	Local OneR	OneR	50NN	Boost OneR	Bagging OneR	Multiboost OneR	Decorate OneR
audiology	74.53	80.32 v	67.97 *	72.28	46.46 *	35.95 *	46.46 *	46.46 *	46.46 *	46.46 *
autos	80.87	72.01 *	67.23 *	76.49	61.77 *	48.18 *	65.47 *	65.26 *	66.06 *	70.51 *
balance-scale	88.3	88.72	86.74	87.36	57.09 *	89.01	72.81 *	67.95 *	72.84 *	60.94 *
breast-cancer	72.58	73.73	73.13	72.16	66.91 *	70.75	69.49	68.45	70	66.87 *
breast-w	96	95.35	96.61	96.3	92.01 *	95.9	95.55	92.81 *	95.45	92.38 *
colic	80.05	75.71	80.95	81.17	81.52	84.04	81.17	81.52	80.32	78.15
credit-rating	82.45	79.1	84.96	83.59	85.51	86.16 v	81.86	85.51	81.87	85.51
diabetes	70.2	70.19	73.86 v	69.76	71.98	74.68 v	69.91	72.08	69.95	71.98
Glass	71.7	75.31	70.02	68.52	56.84 *	56.16 *	56.56 *	60.85 *	56.81 *	58.76 *
haberman	65.73	70.27	69.77	68.39	72.53 v	72.91 v	71.65 v	72.19 v	71.74 v	71.97 v
heart-c	77.33	75.18	81.82	79.23	72.53	81.58	72.61	76.08	76.24	72.83
heart-h	78.04	77.83	82.33	79.14	80.69	83.98 v	76.59	80.49	77.67	80.36
heart-statlog	77.78	76.44	79.11	76.3	71.26	83.74 v	73.59	75.59	75.26	74.93
hepatitis	82.62	80.17	80.85	82.02	82.05	79.38	77.41	82.33	78.73	80.94
ionosphere	90.09	84.64 *	86.02 *	88.21	82.59 *	71.65 *	88.43	85.04 *	88.91	85.59
Iris	93.33	94.67	95.2	94	93.53	90.53	93.67	93.6	94.53	93.47
Labor	90.2	92.03	92.83	82.47 *	72.97 *	64.67 *	89.37	77.93 *	87.53	79.57 *
lymphography	84.63	85.08	81.74	78.31 *	74.77 *	80.59	79.83	74.57 *	79.47	70.66 *
monk1	85.35	80.27	78.97	80.78	73.41 *	59.8 *	70.35 *	73.41 *	70.44 *	74.35 *
monk2	62.36	58.35	54.74 *	56.74	58.35	62.13	55.36	58.76	54.73	57.22
monk3	90.07	86.22	86.72	92.15	77.88 *	82.46 *	90.67	77.57 *	91.82	93.45
Primary-tumor	44.04	38.02 *	44.98	44.57	27.74 *	39.26	27.38 *	27.05 *	27.25 *	27.88 *
sonar	77.85	85.11	83.76	73.57	62.12 *	68.25 *	65.43 *	70.59	65.69 *	62.44 *
Titanic	78.97	77.56 *	78.9	79.03	77.6 *	77.56 *	77.74	77.6 *	77.64 *	77.6 *
Vehicle	70.89	70.22	70.21	67.37 *	52.36 *	63.47 *	51.63 *	52.85 *	51.45 *	52.71 *
Vote	95.79	93.22 *	93.08 *	95.52	95.63	90.41 *	96.62	95.63	95.91	95.63
Wine	96.46	98.72	95.85	94.83	77.93 *	96.46	92.07	79.38 *	89.7 *	79.15 *
W/D/L		1/21/5	1/21/5	0/24/3	1/10/16	5/11/11	1/18/8	1/12/14	1/16/10	1/12/14
Average accuracy	79,93	79,05	79,20	78,53	71,33	73,69	73,69	73,02	73,87	72,68

The presented ensemble is significantly more accurate than single OneR in 16 out of the 27 datasets, while it has significantly higher error rate in one dataset. What is more, the proposed ensemble is significantly more accurate than 3NN and 50NN in 5 and 11 out of the 27 datasets, respectively, whilst it has significantly higher error rate in one and 5 datasets. Likewise, the proposed ensemble is significantly more accurate than Kstar and local weighted DS in 5 and 3 out of the 27 datasets, whilst it has significantly higher error rate in 1 and none dataset, respectively. In addition, the Bagging OneR is significantly more accurate than local boosting in 1 out of the 27 datasets, whilst it has significantly higher error rate in 14 datasets. Furthermore, Adaboost OneR and Multiboost OneR have significantly lower error rates in 1 out of the 27 datasets than proposed ensemble, whereas they are significantly less accurate in 8 and 10 datasets, respectively. Decotate OneR has significantly lower error rates in 1 out of the 27 datasets than local boosting, whereas it is significantly less accurate in 14 datasets.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles that use only the OneR algorithm. The proposed ensemble can achieve an increase in classification about 12% compared to simple OneR. The average relative accuracy improvement of the proposed methodology is from 1% to 10% in relation to the remaining methods.

VI. CONCLUSION

Local memory-based algorithms defer processing of the dataset until they receive request for information (e.g. prediction or local modeling). 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 [3].

Lazy classifiers are particularly useful for classification on data streams. In data streams, new data keep arriving, so building a new classifier 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 classifiers and improve the generalization performance although the global complexity of the system can not be guaranteed to be low. In this paper we propose local boosting and our experiment for some real datasets shows that the proposed combining method outperforms other well known combining methods as well as any individual classifier. Usual boosting algorithms are well known to be sensitive to noise [5]. In the case of local boosting, the algorithm should handle reasonable noise, and be at least as good as boosting, if not better. Due to the encouraging results obtained from these experiments, we can expect that the proposed combining method can be successfully applied to the classification task in the real world case with more accuracy than the traditional data mining approaches.

The benefit of allowing multiple local models is offset by the cost of storing and querying the training dataset for each test set instance which means that lazy learners do not scale well for the large amount of data associated with many 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 classification decisions 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 [6] and [20] can be found numerous instance selection methods that can be combined with local boosting technique.

VII. REFERENCES

[1] D. Aha, *Lazy Learning*. Dordrecht: Kluwer Academic Publishers, 1997.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal, Locally weighted learning for control. *Artificial Intelligence Review*, 11:75–113, 1997.

[3] C.G. Atkeson, A.W. Moore & S. Schaal, Locally weighted learning. *Artificial Intelligence Review*, 11(1-5), 11-73, 1997.

[4] C. Blake & C. Merz, UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html]

[5] E. Bauer and R.. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):525–536, 1999.

[6] H. Brighton, C. Mellish, Advances in Instance Selection for Instance-Based Learning Algorithms, *Data Mining and Knowledge Discovery*, 6, 153–172, 2002.

[7] L. Bottou and V. Vapnik, Local learning algorithm, *Neural Computation*, vol. 4, no. 6, pp. 888-901, 1992.

[8] L. Breiman, Bagging Predictors. *Machine Learning*, 24(3), 123-140. Kluwer Academic Publishers, 1996.

[9] S. Cohen and N. Intrator. Automatic Model Selection in a Hybrid Perceptron/ Radial Network. In Multiple Classifier Systems. *2nd International Workshop*, MCS 2001, pages 349–358.

[10] C. John and L. Trigg, K*: An Instance- based Learner Using an Entropic Distance Measure", *Proc. of the 12th International Conference on ML*, pp. 108-114, 1995.

[11] E. Frank, M. Hall and B. Pfahringer, Locally weighted naive Bayes. *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence*. Acapulco, Mexico. Morgan Kaufmann, 2003.

[12] Y. Freund and R. Schapire, Experiments with a New Boosting Algorithm, *Proc. ICML'96*, p. 148-156, 1996

[13] R. C. Holte, Very simple classification rules perform well on most commonly used datasets, *Machine Learning* 11, 63-90, 1993.

[14] W. Iba, & P. Langley, Induction of one-level decision trees. *Proc. of the Ninth International Machine Learning Conference* (1992). Aberdeen, Scotland: Morgan Kaufmann.

[15] E.M. Kleinberg. A Mathematically Rigorous Foundation for Supervised Learning. In J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science, pages 67–76. Springer-Verlag, 2000.

[16] T. Lazarevic, Z. Obradovic, Adaptive Boosting for Spatial Functions with Unstable Driving Attributes, *In Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2000, pp. 329-340.

[17] P. Melville and R. Mooney, Constructing Diverse Classifier Ensembles using Artificial Training Examples, *Proc. of the IJCAI-2003*, pp.505-510, Acapulco, Mexico, August 2003

[18] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[19] G. I. Webb, MultiBoosting: A Technique for Combining Boosting and Wagging, *Machine Learning*, 40, 159–196, Kluwer Academic Publishers, 2000.

[20] D. Wilson, and T. Martinez, Reduction Techniques for Instance-Based Learning Algorithms, *Machine Learning*, 38, 257–286, 2000.

[21] I. Witten, and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Mateo, CA, 2000.