Boosting Localized Classifiers in Heterogeneous Databases[*]

Aleksandar Lazarevic[†] and Zoran Obradovic[†]

Abstract. Combining multiple global models (e.g. back-propagation based neural networks) is an effective technique for improving classification accuracy. This technique reduces variance by manipulating the distribution of the training data. In many large scale data analysis problems involving heterogeneous databases with attribute instability, standard boosting methods can be improved by coalescing multiple classifiers. Each classifier uses different germane attribute information that is identified through the attribute selection process. We propose a new technique of boosting localized classifiers when heterogeneous data sets contain more homogeneous data distributions. Instead of a single global classifier for each boosting round, we have localized classifiers responsible for each homogeneous region. The number of regions is identified through a clustering algorithm performed at each boosting iteration. A new boosting method applied to real life spatial data and synthetic spatial data shows improvements in prediction accuracy when unstable driving attributes and heterogeneity are present in the data. In addition, boosting localized experts significantly reduces the number of iterations needed for achieving the maximal prediction accuracy.

1 Introduction

Many large-scale data analysis problems involve an investigation of relationships between attributes in heterogeneous databases, where different prediction models can be responsible for different regions. In addition, large data sets very often exhibit attribute instability, such that the set of relevant attributes is not the same through the entire data

[*] Partial support provided by National Institute of Health; Grant Number: 1R01LM06916.
[†] Center for Information Science and Technology, College of Science and Technology, Temple University, Room 303, Wachman Hall (038-24), 1805 N. Broad St., Philadelphia, PA 19122, USA, aleks@astro.temple.edu, zoran@joda.cis.temple.edu

space. This is especially true in spatial databases, where different spatial regions may have completely different characteristics [1].

One of the most effective recent techniques for improving prediction accuracy in machine learning theory and pattern classification is combining multiple classifiers. There are many general combining algorithms such as bagging [2], boosting [3], or Error Correcting Output Codes (ECOC) [4] that significantly improve global classifiers like decision trees, rule learners, and neural networks. These algorithms may manipulate the training patterns that individual classifiers use (bagging, boosting) or the class labels (ECOC). In most of the algorithms the weights of different classifiers are the same for all the patterns within the data set to which they are applied.

In order to improve the global accuracy of the whole, an ensemble of classifiers must be both accurate and diverse. In heterogeneous databases there usually exist several more homogeneous regions. To improve the accuracy of the ensemble of classifiers for these databases, instead of applying a global classification model across entire data sets, the models are varied to better match site-specific needs thus improving prediction capabilities [5]. Therefore, in such an approach there is a local classification expert responsible for each region that strongly dominates the others from the pool of local experts.

Diversity of the ensemble is also required to ensure that all the classifiers do not make the same errors. In order to increase the diversity of combined classifiers for spatial heterogeneous databases with attribute instability, one cannot assume that the same set of attributes is appropriate for each single classifier. For each training sample, drawn in a bagging or boosting iteration, a different set of attributes is relevant and therefore the appropriate attribute set should be used by local classification experts built at each iteration.

In this paper, we extend the framework for the construction of composite classifiers through the AdaBoost algorithm [3]. Work by several authors [6, 7, 8, 9] has provided a rather general approach to boosting, through an incremental greedy minimization of some empirical cost function. In our approach, in each boosting round we try to maximize the local information for a drawn sample by allowing the weights of the different weak classifiers to depend on the input. Rather than having constant weights attached to each of the classifiers (as in standard approaches), we allow weights to be functions over the input domain. In order to determine these weights, at each boosting iteration we identify local regions having similar characteristics using a clustering algorithm and then build local classification experts on each of these regions describing the relationship between the data characteristics and the target class [1]. Therefore, instead of a single classifier built on a sample drawn in each boosting iteration, there are several local classification experts responsible for each of the regions identified through the clustering process. All data points belonging to the same region and hence to the same classification expert will have the same weights when all classification experts are combined. In addition, the local information is also emphasized with changing attribute representation through attribute selection methods at each boosting iteration [10].

In the next section, we discuss current ensemble approaches and work related to localized experts and changing attribute representations of combined classifiers. In Section 3 we describe the proposed method and investigate its advantages and limitations. In Section 4, we evaluate the proposed method on real-life and synthetic data sets by comparing it with standard boosting and other methods for dealing with heterogeneous databases. Finally, section 5 concludes the paper and suggests further directions in current research.

2 Related Work

Recently, researchers have begun experimenting with general algorithms for improving classification accuracy by combining multiple versions of a single classifier, also known as a multiple model or an ensemble approach [2, 3, 4]. Unfortunately, it seems that none of these combining methods can be very successful in improving the prediction accuracy for heterogeneous databases [11]. Several recent approaches for analyzing heterogeneous data are based on changing attribute representation for each of the coalesced classifiers.

FeatureBoost [12] is a recently proposed variant of boosting where attributes are boosted rather than examples. While standard boosting algorithms alter the distribution by emphasizing particular training examples, FeatureBoost alters the distribution by emphasizing particular attributes. The goal of FeatureBoost is to search for alternate hypotheses amongst the attributes. A distribution over the attributes is updated at each boosting iteration by conducting a sensitivity analysis on the attributes used by the model learned in the current iteration. The distribution is used to increase the emphasis on unused attributes in the next iteration in an attempt to produce different sub-hypotheses.

Only a few months earlier, a considerably different algorithm exploring a similar idea for an adaptive attribute boosting technique was published [11]. The technique coalesces multiple local classifiers each using different relevant attribute information. The related attribute representation is changed through attribute selection, attribute extraction and attribute weighting processes performed at each boosting round. In addition, a modification of the boosting method is developed for heterogeneous spatial databases with unstable driving attributes by drawing spatial blocks of data at each boosting round. This method was mainly motivated by the fact that standard combining methods do not improve local classifiers (e.g. k-nearest neighbors) due to their low sensitivity to data perturbation, although the method was also used with global classifiers like neural networks.

In addition to the previous method, there were a few more experiments in selecting different feature subsets as an attempt to force the neural network classifiers to make different and hopefully uncorrelated errors. Although there is no guarantee that using different attribute sets will decorrelate error, Tumer and Ghosh [13] found that with neural networks, selectively removing attributes could decorrelate errors. Unfortunately, the error rates in the individual classifiers increased, and as a result there was little or no improvement in the ensemble. Cherkauer [14] was more successful, and was able to combine neural networks that used different hand selected attributes to achieve human expert level performance in identifying volcanoes from images.

Opitz [15] has investigated the notion of an ensemble feature selection with the goal of finding a set of attribute subsets that will promote disagreement among the component members of the ensemble. A genetic algorithm approach was used for searching an appropriate set of attribute subsets for ensembles. First, an initial population of classifiers is created, where each classifier is generated by randomly selecting a different subset of attributes. Then, the new candidate classifiers are continually produced, by using the genetic operators of crossover and mutation on the attribute subsets. The algorithm defines the overall fitness of an individual to be a combination of accuracy and diversity.

Unlike the approaches that change attribute representation, there is another group of methods for analyzing heterogeneous databases based on building different local classification experts, each responsible for a particular data region. Our recent approach [5] belongs to this category and is designed for analysis of spatially heterogeneous databases. It first clusters the data in the space of observed attributes, with an objective of identifying similar spatial regions. This is followed by local prediction aimed at learning relationships between driving attributes and the target attribute inside each cluster. The method was also extended for learning when the data are distributed at multiple sites.

A similar method is based on a combination of classifier selection and fusion by using statistical inference to switch between these two [16]. Selection is applied in regions of the attribute space where one classifier strongly dominates the others from the pool (clustering-and-selection step), and fusion is applied in the remaining regions. Decision templates (DT) are adopted for classifier fusion, where all classifiers are trained over the entire attribute space and thereby considered as competitive rather than complementary.

Some researchers also have tried to combine boosting techniques with building single classifiers in order to improve prediction in heterogeneous databases. One such approach is based on a supervised learning procedure, where outputs of predictors are trained on different distributions followed by a dynamic classifier combination [17]. This algorithm applies principles of both boosting and the mixture of experts [18] and shows high performance on classification or regression problems. The proposed algorithm may be considered either as a boost wise initialized Mixture of Experts, or as a variant of Boosting which uses a dynamic model for combining the output of the classifiers. The main characteristic of boosting included in this scheme is the ability to initialize a split of the training set to different experts. This split is based on a difficulty criterion. Unlike standard boosting where this difference depends on the errors of the first classifier or the disagreement between the first two classifiers, this method uses a confidence measure as the difficulty criterion. The algorithm is designed for an arbitrary number of experts as the ensemble is constructed gradually by adding a new expert and re partitioning the data. The first expert is trained on the entire training set. The patterns on which the current experts are not confident are assigned to the initial training set of a new expert and used for its learning. This procedure is repeated until no more experts are required. When all experts are constructed, the entire training data set is repartitioned according to the current confidence level of each expert on each pattern.

3 Boosting Localized Experts

It is known that boosting is an effective technique for improving prediction accuracy in many real life data sets [2, 7, 19]. However, our previous research indicated that in heterogeneous databases, where several more homogeneous regions exist, boosting does not enhance the prediction capabilities as well as for homogeneous databases [11]. In such cases it is more useful to have several local experts responsible for each region of the data set. A possible way to approach this problem is to cluster the data first and then to assign a single classifier to each discovered cluster. In this paper we try to combine this approach with the standard boosting technique in order to further improve generalization capabilities of local classification models.

We follow the generalized analysis of AdaBoost.M2 algorithm [3]. Our boosting extension, described in Figure 1, models a scenario in which the relative significance of each expert advisor is a function of the attributes from the specific input patterns. This extension seems to better model real life situations where particularly complex tasks are split among experts, each with expertise in a small spatial region.

In this work as in many boosting algorithms, the final composite hypothesis is constructed as a weighted combination of base classifiers. The coefficients of the combination in the standard boosting, however, do not depend on the position of the point x whose label is of interest. Since the boosting procedure filters data successively through re-weighting, it is possible that some of the classifiers $h_t(x)$ were not exposed during training to any data in the vicinity of the point x. Moreover, greater flexibility can be achieved by having each classifier operate only in a localized region. Therefore, it would seem more suitable to weight each classifier h_t at point x by a local weight $\beta_t(x)$ depending on the point x.

- Given: Set S = $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, k\}$
- Let B = $\{(i, y): i \in \{1,2,3,4,\ldots m\}, y \neq y_i\}$
- Initialize the distribution D_1 over the examples, such that $D_1(i) = 1/m$.
- While (t < T) or (global accuracy on set S starts to decrease)

1. Find relevant attribute information for distribution D_t.
2. Obtain c distributions $D_{t,j}$, j = 1, … c and corresponding sets $S_j = \{(x_{1,j}, y_{1,j}), \ldots, (x_{m_j,j}, y_{m_j,j})\}$ $x_{i,j} \in X_j$, with labels $y_{i,j} \in Y_j = \{1, \ldots, k\}$ from clusters discovered in an unsupervised w rapper approach around clustering performed in step 1. Clustering was performed using the most relevant attributes also identified in step 1. Let $B_j = \{(i^j, y^j): i^j \in \{1,2,3,4,\ldots m^j\}, y^j \neq y_i^j\}$.
3. For j = 1 … c (For each of c clusters)

   3.1. Find relevant attri bute representation for distribution $D_{t,j}$ using supervised feature selection
   3.2. Train a weak learner using distribution $D_{t,j}$
   3.3. Compute weak hypothesis $h_{t,j}$: $X_j \times Y_j \rightarrow [0, 1]$
   3.4. Compute convex hulls $H_{t,j}$ for each of c clusters from the entire set S
   3.5. Compute the pseudo-loss of hypothesis $h_{t,j}$:

   $$\varepsilon_{t,j} = \frac{1}{2} \sum_{(i^j, y^j) \in B_j} D_{t,j}(i^j, y^j)(1 - h_{t,j}(x_{i,j}, y_{i,j}) + h_{t,j}(x_{i,j}, y^j))$$

   3.6. Set $\beta_{t,j} = \varepsilon_{t,j} / (1 - \varepsilon_{t,j})$
   3.7. Determine clusters on the entire training set according to the convex hull mapping. All points inside the convex hull $H_{t,j}$ belong to the *j-th* cluster $T_{t,j}$ from iteration t.

4. Merge all $h_{t,j}$, $j = 1, \ldots c$ into a unique weak hypothesis h_t and all $\beta_{t,j}$, $j = 1, \ldots c$ into an unique β_t according to convex hull belonging (example fitting in the *j-th* convex hull has the hypothesis $h_{t,j}$ and the value $\beta_{t,j}$).
5. Update D_t : $D_{t+1}(i, y) = (D_t(i, y) / Z_t) \cdot \beta_t(i, y)^{(1/2)\cdot(1 + h_t(x_i, y_i) - h_t(x_i, y))}$
   where Z_t is a normalization constant chosen such that D_{t+1} is a distribution.
6. Output the final hypothesis: $h_{fn} = \arg\max_{y \in Y} \sum_{t=1}^{T} \bigcup_{j=1}^{c} (\log \frac{1}{\beta_{tj}(i^j, y^j)}) \cdot h_{t,j}(x^j, y^j)$

Figure 1. ***The scheme for boosting localized classifiers with performing attribute selection (step 1) in each boosting iteration***

The algorithm proceeds in a series of T rounds. In each round , the entire weighted training set is given to the set of local weak learners to compute a unique weak hypothesis h_t. The distribution is updated to give wrong classifications higher weights than correct classifications.

Since at each boosting iteration t we have different training samples drawn according to the distribution D_t, at the beginning of the "for loop" in Figure 1 we include **step 1**, wherein we choose different attribute subsets for each sample. Different attribute representations are realized through a feature selection process in the boosting iterations. Regression-based attribute selection was carried out through performance feedback [10] forward selection and backward elimination search based on linear regression mean square error (MSE) minimization. The r most relevant attributes are chosen according to the selection criterion at each round of boosting, and are used by the clustering algorithm and classification models. Thus, for each round of boosting we have different relevant attribute subsets representing the drawn sample, in an attempt to force the single global classifiers to make different and hopefully uncorrelated errors.

In addition to attribute instability in a sample drawn from a heterogeneous database there are usually several more homogeneous regions. Therefore, at each boosting iteration we perform clustering in order to find those homogeneous regions. As a result of the clustering, we obtain several distributions $D_{t,j}$ (j = 1,…, c), where c is the number of discovered clusters. For each of c clusters discovered in the data sample, we first identify relevant attributes using supervised feature selection procedure. Then, we train a weak learner using the corresponding data distribution and compute a weak hypothesis $h_{t,j}$. Furthermore, for every cluster from the data sample, we identify its convex hull in the attribute space used for clustering, and map these convex hulls to the entire training set in order to find the corresponding clusters where the local classifiers will be applied (Figure 2) [20]. All data points inside the convex hull $H_{t,j}$ belong to the *j-th* cluster discovered at iteration t. Data points outside the convex hulls are attached to the cluster containing the closest data pattern. Therefore, instead of a single global classifier constructed in every iteration by the standard boosting approach, there are c classifiers and each of them is applied to the corresponding mapped cluster.

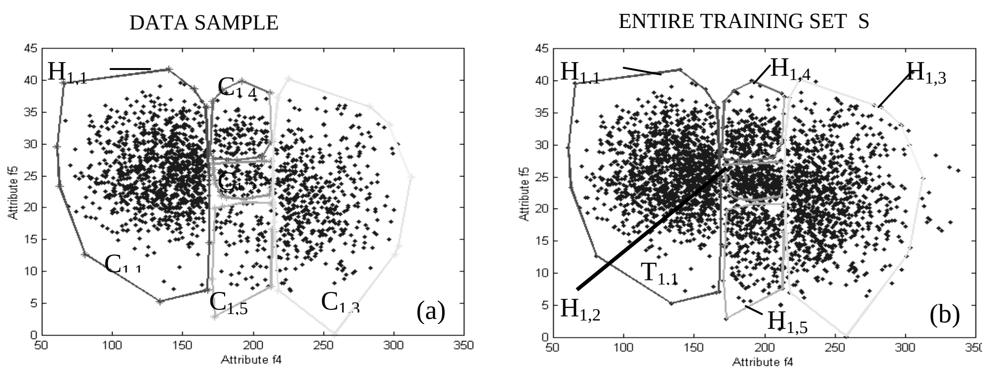

Figure 2. ***Mapping convex hulls $H_{1,j}$ of clusters $C_{1,j}$, j = 1, …,c, (discovered in the data sample), to the entire training set in order to find corresponding clusters. For example, all data points inside the contours of the convex hull $H_{1,1}$ (corresponding to the cluster $C_{1,1}$ discovered on the data sample) belong to the new cluster $T_{1,1}$ identified on the entire training set.***

In standard boosting all data points have the same pseudo-loss ε_t and the parameter β_t when combining the classifiers from the boosting iterations. In our approach data points from different clusters have different pseudo-loss values and different parameter values β_t. For each cluster j, (j =1,…,c) from iteration t, defined with the convex hull $H_{t,j}$, there is a pseudo-loss $\varepsilon_{t,j}$ and the corresponding parameter $\beta_{t,j}$. Each pseudo-loss value $\varepsilon_{t,j}$ is computed independently for each cluster where a particular classifier is responsible. The value of the parameter $\beta_{t,j}$ is also computed separately for each cluster using the corresponding pseudo-loss value $\varepsilon_{t,j}$. Before updating the distribution D_t, the parameters $\beta_{t,j}$ for *c* clusters are merged into a unique vector β_t such that the *i*-th pattern from the data set that belongs to the *j-th* cluster specified by the convex hull $H_{t,j}$, corresponds to the parameter $\beta_{t,j}$ at the *i*-th position in the vector β_t. Analogously, the hypotheses $h_{t,j}$ are merged into a single hypothesis h_t. Since we merged $\beta_{t,j}$ and $h_{t,j}$ into β_t and h_t respectively, the updating of the distribution D_t can be performed as in the standard boosting algorithm. However, in making the final hypothesis h_{fn} the local classifiers from each iteration are first applied to the corresponding clusters and integrated into a composite classifier responsible for that iteration. These composite classifiers are then combined using the standard AdaBoost.M2 algorithm.

The clustering technique is an important part of the proposed algorithm. Using attributes derived from feature selection at step 0 of each boosting iteration, two clustering algorithms were employed to partition the spatial data set into “similar” regions. The first one called DBSCAN relies on a density-based notion of clusters and was designed to discover clusters of an arbitrary shape efficiently [21]. The key idea of density-based clustering is that for each point of a cluster its *Eps*-neighborhood for a given *Eps* > 0 has to contain at least a minimum number of points (*MinPts*), (i.e. the density in the *Eps*-neighborhood of points has to exceed some threshold). Furthermore, the typical density of points inside clusters is considerably higher than outside of clusters. DBSCAN uses a simple but effective heuristic for determining the parameters *Eps* and *MinPts* for the smallest cluster in the database.

The second clustering algorithm used in our proposed method is the standard *k*-means algorithm [22]. Here, data set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, $x_i \in X$, is partitioned into *k* clusters by finding *k* points $\{m_j\}_{j=1}^{k}$ such that

$$\frac{1}{n}\sum_{x_i \in X}\left(\min_j d^2(x_i, \mu_j)\right)$$

is minimized, where $d^2(x_i, m_j)$ usually denotes the Euclidean distance between x_i and m_j, although other distance measures can be used. The points $\{\mu_j\}_{j=1}^{k}$ are known as *cluster centroids*.

When performing clustering during boosting iterations, it is possible that some of the discovered clusters are relatively small and therefore there is an insufficient number of data points needed for training a local classifier. Several techniques for handling these scenarios were considered.

The first technique denoted as ***simple*** halts the boosting process when a cluster with a small number of data points is detected. This number of data patterns is defined as a function of the number of patterns in the entire training set. When the boosting procedure is terminated, only the classifiers from the previous iterations are combined in order to create the final hypothesis h_{fn}.

A more sophisticated technique for addressing small clusters does not stop the boosting process, but instead of training the local classifier on the detected cluster with insufficient amount of the data, it employs the local classifiers constructed in previous iterations. When a cluster with an insufficient number of data points is identified, its corresponding cluster from previous iterations is detected using the convex hull matching

(Figure 2) and the model constructed on the corresponding cluster is applied on the cluster discovered in the current iteration. The most effective method for determining the model that should be applied is to take the classification model constructed in the iteration where the *local* prediction accuracy for the corresponding cluster was maximal. This technique represented as ***best_local*** will be compared to the ***simple*** method as well as to two similar techniques: ***previous*** and ***best_global***. The ***previous*** method always takes the classifiers constructed on the corresponding cluster from the *previous* iteration, while the ***best_global*** technique uses the classification models constructed on the corresponding cluster from the iteration where the *global* prediction accuracy, achieved by applying final hypothesis h_{fn}, was maximal. In all these sophisticated techniques, the boosting procedure ceases when the prespecified number of iterations is reached or there is a significant drop in the prediction accuracy for the training set.

We used multilayer (2-layered) feedforward neural network classification models with the number of hidden neurons equal to the number of input attributes. We also experimented with different numbers of hidden neurons. The neural network classification models had the number of output nodes equal to the number of classes (3 in our experiments), where we predicted the class given by the output with largest response. We used two learning algorithms: resilient propagation [23] and Levenberg-Marquardt [24].

To further experiment with attribute stability properties, miscellaneous attribute selection algorithms [10] were applied to the entire training set and the most stable attributes were selected. The standard boosting method was applied to the global and local classifiers using the identified fixed set of attributes at each boosting iteration. When boosting is applied with attribute selection at each boosting round, the attribute occurrence frequency is monitored in order to identify the most stable selected attributes. The hypothesis considered in the next section was that when attribute subsets selected through boosting iterations become stable, it is appropriate to stop the boosting process.

4 Experimental Results

Our experiments were first performed on two synthetic data sets corresponding to 5 homogeneous data distributions made using our spatial data simulator [25]. The attributes f4 and f5 were simulated to form five clusters in their attribute space (f4, f5) using the technique of feature agglomeration [25]. Furthermore, instead of using one model for generating the target attribute on the entire spatial data set, a different data generation process using different relevant attributes was applied per each cluster, such that the distributions of generated data resembled the distributions of real life data. The degree of relevance was also different for each distribution. Both data sets had 6561 patterns with 5 relevant (f1, ..., f5) and 5 irrelevant attributes (f6, ..., f10), where one was used for training, and another one for out of sample testing. The histograms of all 5 attributes for all 5 distributions are shown in Figure 3.

We also performed experiments using spatial data from a 220 ha field located near Pullman, WA. All attributes were interpolated to a 10x10 m grid resulting in 24,598 patterns. The Pullman data set contained x and y coordinates (attributes 1-2), 19 soil and topographic attributes (attributes 3-21) and the corresponding crop yield. The field was spatially partitioned into training and test set (left half of the field was the training set, while right half served as the test set). The attributes used were: baresoil, soil type, elevation, primal sketch, solar radiation, compound topographic index, aspect east-west, aspect north-south, distance to long flow, flow direction, flow width, slope, plan curvature, profile curvature, tangent curvature, average upslope slope, average upslope plan curvature, average upslope profile curvature, and average upslope tangent curvature.

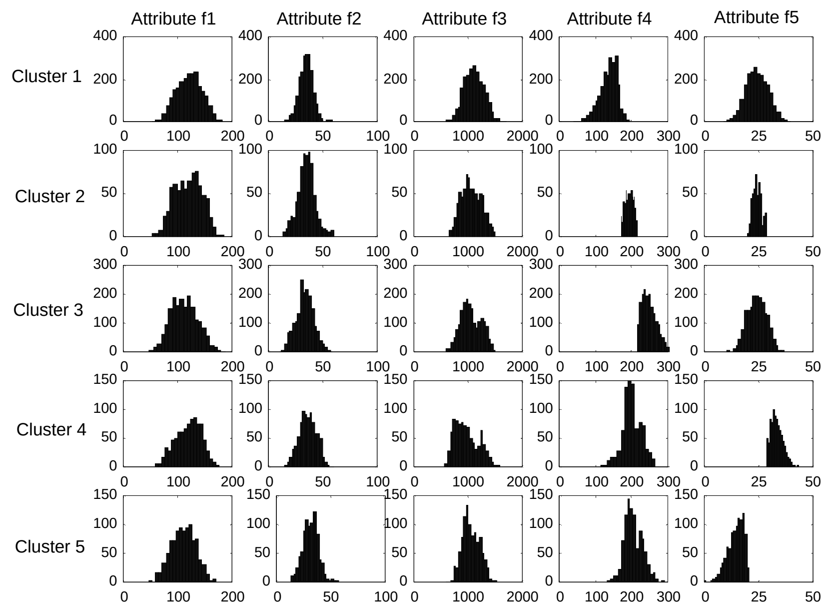

Figure 3. ***Histograms of all 5 relevant attributes for all 5 clusters of a synthetic data set***

For the synthetic data set we performed standard boosting, adaptive attribute boosting (boosting with attribute selection at each iteration) and all proposed variants of boosting localized experts (boosting with clust ering). For each of these methods, the reported classification accuracies for 3 equal size classes were obtained by averaging over 10 trials of all proposed boosting algorithms applied to neural network classifiers (Figure 4 and Table 1). For all reported results, the best prediction accuracies were achieved when using the Levenberq-Marquardt algorithm for training neural networks.

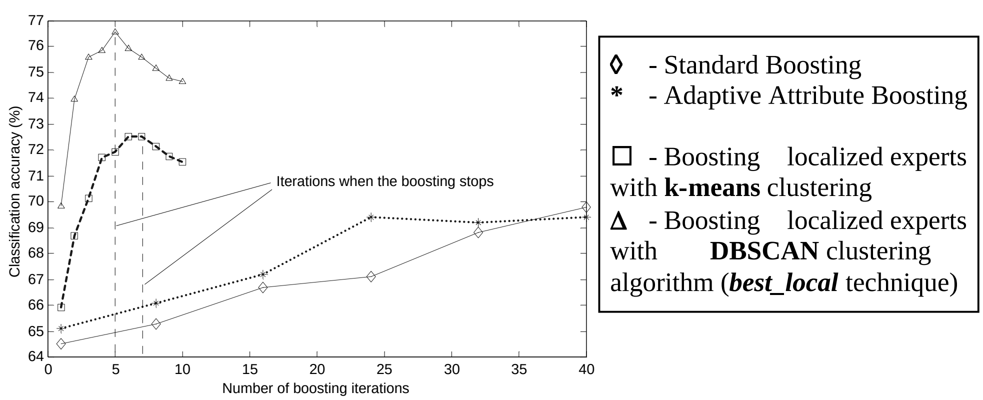

Figure 4. ***Overall classification accuracies for 3-class predictors on out of sample (test) synthetic data set with 5 relevant and 5 irrelevant attributes and five clusters defined by 2 of 5 relevant attributes.***

Table 1. ***Final classification accuracies for the 3-class problems. Different boosting algorithms are applied on out of sample synthetic data with 5 relevant and 5 irrelevant attributes and 5 clusters.***

<table>
<tr><th colspan="3">Method</th><th>Classification accuracy (%)</th></tr>
<tr><td colspan="3">Global Approach</td><td>61.0 ± 2.2</td></tr>
<tr><td colspan="3">DBSCAN Clustering with specialized classifiers</td><td>71.3 ± 0.9</td></tr>
<tr><td colspan="3">Standard Boosting</td><td>69.8 ± 1.1</td></tr>
<tr><td colspan="3">Adaptive Attribute Boosting</td><td>69.4 ± 1.1</td></tr>
<tr><td rowspan="5">Boosting Localized Experts with Clustering</td><td colspan="2">k-means clustering</td><td>72.6 ± 1.1</td></tr>
<tr><td rowspan="4">DBSCAN clustering</td><td>simple</td><td>73.9 ± 1.7</td></tr>
<tr><td>previous</td><td>74.4 ± 1.5</td></tr>
<tr><td>best_global</td><td>74.9 ± 1.4</td></tr>
<tr><td>best_local</td><td>76.6 ± 1.2</td></tr>
</table>

Analyzing the data in Table 1 and the charts in Figure 4, the method of adaptive attribute boosting was not significantly better than the standard boosting model, but the all variants of boosting localized experts considerably outperformed both the standard boosting and the adaptive attribute boosting.

Observe that the adaptive attribute boosting results showed no improvements in prediction accuracy. This was due to properties of the synthetic data set, where each spatial region had not only different relevant attributes related to yield class but also a different number of relevant attributes. In such a scenario with uncertainty regarding the number of relevant attributes for each region, we needed to select at least the 4 or 5 most important attributes at each boosting round, since selecting 3 most relevant attributes may be insufficient for successful learning. However, the total number of relevant attributes in the data set was 5 as well, and therefore it was meaningless to select 5 attributes during the boosting rounds since we cannot achieve any attribute instability. Therefore, we were selecting the 4 most relevant attributes for adaptive attribute boosting, knowing that for some drawn samples we would lose beneficial information. In the standard boosting method we used all 5 relevant attributes from the data set. Nevertheless, we obtained similar classification accuracies for both the adaptive attribute boosting and the standard boosting method, but adaptive attribute boosting reached the “bounded” final prediction accuracy in fewer boosting iterations. This property could be useful for reducing the time needed for the latest boosting rounds. Instead of post-pruning the boosted classifiers [26] we can try to set the appropriate number of boosting iterations at the beginning of the procedure.

All methods of boosting localized experts resulted in improved generalization of approximately 10 % as compared to standard and adaptive attribute boosting. It was also evident that the boosting of localized experts required fewer iterations in order to reach the maximal prediction accuracy. After the prediction accuracy was maximized, the overall prediction accuracy on the training set, as well as the total classification accuracy on the test set, started to decline. This phenomenon was probably due to the fact that in the later iterations only data points that were difficult for learning were drawn and therefore the prediction accuracy of the local models built in those iterations began to deteriorate. As a consequence, the total prediction accuracy decreased too.

The data distribution of discovered clusters was monitored at each boosting iteration by performing DBSCAN clustering algorithm (Figure 5). Unlike the previous adaptive attribute boosting method when around 30 boosting iterations were needed to achieve good generalization results, here typically only a few iterations (5 – 10) were sufficient for reaching the maximum prediction accuracy on the training set. As could be observed in Figure 5, data samples drawn in initial iterations (iteration 1) clearly included data points from all five clusters while samples drawn in later iterations (iterations 4, 5) contained very small number of data points from the clusters where the prediction accuracy was good. Therefore, as one of the criteria for stopping boosting early, we accepted the following rule: the boosting procedure stops when the size of any of the discovered clusters is less than some predefined number (usually less than 50).

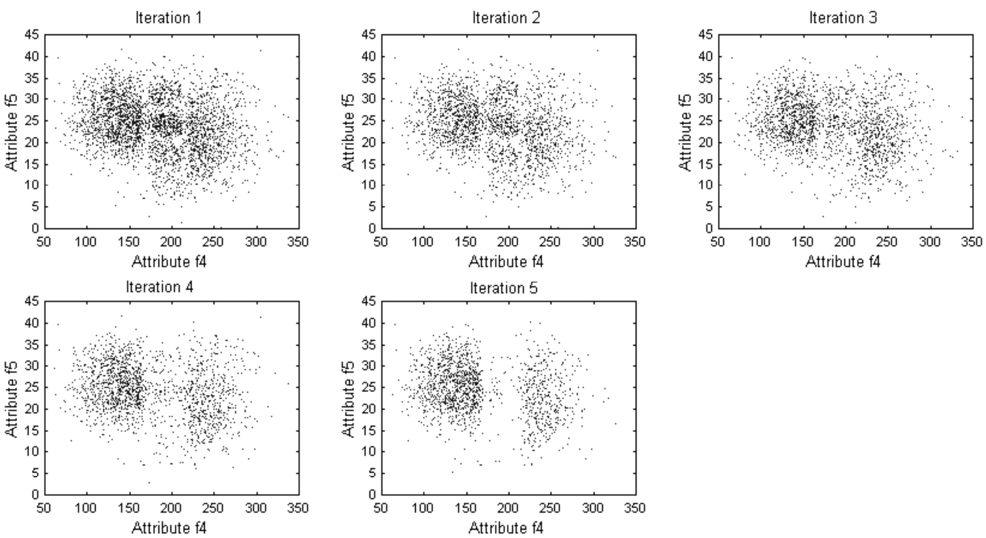

Figure 5. ***Changing the distributions of drawn samples during boosting on the neural network classifier. Samples from initial iterations contain points from all clusters, while samples from later iterations contain a small number of points from the central clusters where the accuracy was good.***

An additional criterion for stopping the boosting algorithm early is to observe the classification accuracy on the entire training set and to stop the procedure when it starts to decline. Figure 4 shows the iterations when we stop the boosting procedure. This is the moment when the classification accuracy on the training set starts to decline. Although in practice the prediction accuracy on the test set does not necessarily start to drop in the same iteration, this difference is usually up to two boosting iterations and does not significantly affect the total generalizability of the proposed method.

However, when using the k-means clustering algorithm during the boosting procedure, we did not notice the phenomenon of reducing the number of data points in discovered clusters. Therefore, for the k-means variant of boosting localized experts we did not perform the modifications of the proposed algorithm. In addition, it was evident that boosting localized experts when using k-means clustering algorithm was not as successful as boosting localized experts with the DBSCAN algorithm, due to better quality clusters identified by DBSCAN which was designed to discover spatial clusters of arbitrary shape.

Nevertheless, when using the DBSCAN algorithm at each boosting round, the ***best_local*** technique provided the best prediction accuracy (Table 1), while the other methods were not significantly better than the boosting localized experts with k-means clustering. The ***simple*** technique failed to achieve improved prediction results, since it did not reach enough boosting iterations to develop the most appropriate classifiers for each cluster that need to be combined. On the other hand, the ***previous*** method had boosting cycle that was long enough, but did not combine appropriate models. Therefore, both methods coalesced the classifiers that could not generalize well or they were built on clusters without enough training data. Finally, the ***best_global*** and ***best_local*** combined the most accurate models for each cluster taken in some of the earlier iterations, and hence achieved the best generalizability. However, the prediction accuracy of all models deteriorated in later boosting iterations, due to drawing only data points that were difficult to learn.

Experiments with all proposed boosting modifications were repeated for training and test sets of real life spatial data. The goal was to predict 3 equal size classes of wheat yield as a function of soil and topographic attributes. For real life data (Pullman data set) 17 miscellaneous attribute selection methods were used to identify the 4 most relevant attributes on the training data set (Table 2) and the histograms for the most stable attributes (4, 7, 9, 20) are shown in Figure 6. These attributes were used for the global prediction method when a single model is learned on the entire training set and applied on the test data set, for the standard boosting method, and for variants of the boosting localized experts without performing attribute selection at each boosting round.

Table 2. ***Attribute selection methods used to identify 4 most stable attributes on training data set***

Attribute Selection Methods			Selected attributes
Branch & Bound methods	Probabilistic distance	Mahalanobis distance	7, 9, 11, 20
		Bhatacharya distance	4, 7, 10, 14
		Patrick-Fisher distance	13,17, 20, 21
Forward Selection methods	Inter-class distance	Minkowski (order = 1)	7, 9, 10, 11
		Minkowski (order = 3)	3, 4, 5, 7
		Euclidean distance	3, 4, 5, 7
		Chebychev distance	3, 4, 5, 7
	Probabilistic distance	Bhatacharya distance	3, 4, 8, 9
		Mahalanobis distance	7, 9, 11, 20
		Divergence distance metric	3, 4, 8, 9
		Patrick-Fisher distance	13,16, 20, 21
	Minimal Error Probability, k-NN with resubstitution		4, 7, 11, 19
	Linear regression performance feedback		5, 9, 7, 18
Backward Elimination methods	Probabilistic distance	Mahalanobis distance	7, 9, 11, 20
		Bhatacharya distance	4, 7, 9, 14
		Patrick-Fisher distance	13,17, 20,21
	Linear regression performance feedback		7, 9, 11, 20

When performing attribute selection during boosting, the selected attributes were monitored and their frequency was computed. The frequency of selected attributes during the boosting rounds, when the adaptive attribute boosting without performing clustering at each iteration was applied to neural network classification models, is presented in Figure 7.

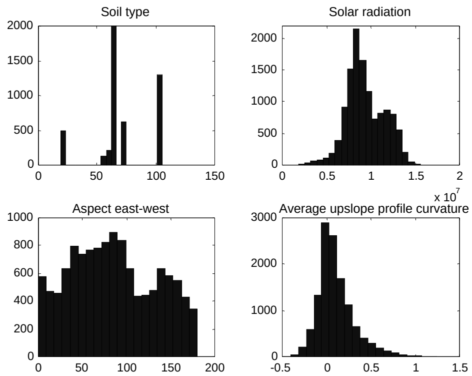

Figure 6. ***Histograms of 4 most relevant attributes of real life data set***

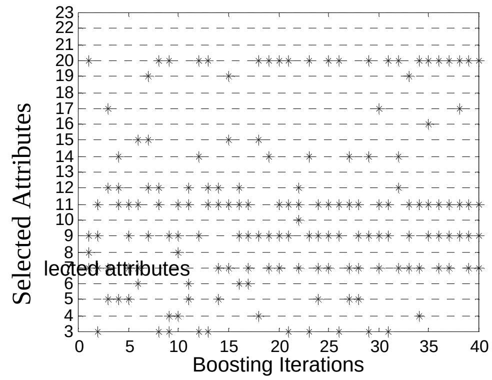

Figure 7. ***Attribute stability during boosting on the Levenberq-Marquardt algorithm on real life data (* denotes that the attribute is selected in boosting iteration; - denotes that the attribute is not selected)***

The results in Figure 8 were obtained by the backward elimination attribute selection technique using the Levenberq-Marquardt algorithm for optimizing neural network parameters. When using the method of boosting localized experts, the best experimental results were achieved again with the ***best_local*** technique and the Levenberq-Marquardt algorithm and only these results are reported in Figure 8 and Table 3. The same stopping criteria for the boosting procedure, as for the synthetic data sets, were used. In these experiments adaptive attribute boosting outperformed the standard boosting model, while all 4 variants of boosting localized experts with clustering through iterations were more successful than the standard boosting, the adaptive attribute boosting and the method of building specialized classifiers on clusters identified using DBSCAN algorithm (Table 3).

Table 3. ***Final classification test accuracies for the 3 -class problems. Different boosting algorithms are applied to the out of sample real life data set with 19 soil and topographic attributes.***

<table>
<tr><th colspan="3">Method</th><th>Classification accuracy (%)</th></tr>
<tr><td colspan="3">Global Approach</td><td>42.4 ± 2.2</td></tr>
<tr><td colspan="3">DBSCAN Clustering with specialized classifiers</td><td>49.7 ± 0.9</td></tr>
<tr><td colspan="3">Standard Boosting</td><td>45.5 ± 1.1</td></tr>
<tr><td colspan="3">Adaptive Attribute Boosting</td><td>48.8 ± 1.1</td></tr>
<tr><td rowspan="4">Boosting Localized Experts with Clustering</td><td rowspan="2">k-means clustering</td><td>without attribute selection</td><td>50.3 ± 1.2</td></tr>
<tr><td>WITH attribute selection</td><td>$\mathbf{50.6 \pm 1.1}$</td></tr>
<tr><td rowspan="2">DBSCAN clustering</td><td>without attribute selection</td><td>52.2 ± 1.3</td></tr>
<tr><td>WITH attribute selection</td><td>$\mathbf{52.4 \pm 1.4}$</td></tr>
</table>

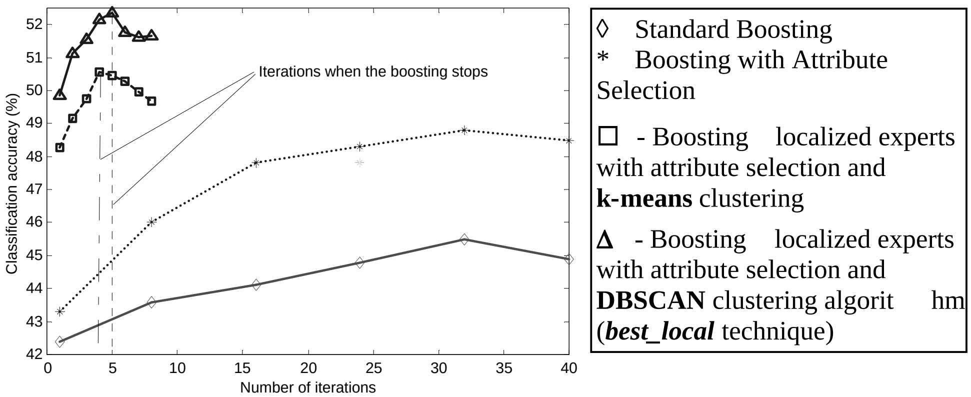

Figure 8. ***Overall classification accuracies for the 3 -class predictors on out of sample (test) real life data set***

It appeared that for pure adaptive attribute boosting with only attribute selection, monitoring selected attributes could be a good criterion for stopping boosting early, since after the selected attribute subsets had became stable, no significant improvements in prediction accuracy were noticed. The results indicate that 30 boosting rounds were usually sufficient to maximize prediction accuracy. During the boosting iterations we were selecting the 4 and 5 most important attributes, and the number of hidden neurons in a 2-layer feedforward neural network was equal to the number of input attributes. We noticed that further increasing the number of hidden neurons did not improve prediction accuracy probably due to overfitting.

The boosting localized experts on a real life heterogeneous data set is not as superior to the adaptive attribute boosting as for the synthetic data set, since higher attribute instability was apparently beneficial for the adaptive attribute boosting. Similar to experiments on synthetic data, the ***best_local*** technique of boosting localized experts was the most successful among all the proposed methods.

5 Conclusion

Results from two spatial data sets indicate that the proposed algorithm for combining multiple classifiers can result in significantly better predictions over existing classifier ensembles, especially for heterogeneous data sets with attribute instabilities. First, this study provides evidence that by manipulating the attribute representation used by individual classifiers at each boosting round, classifiers could be more decorrelated thus leading to higher prediction accuracy. The attribute stability test also served as a good indicator for stopping further boosting iterations properly. Second, boosting localized experts with applied clustering at each boosting round further significantly improved the achieved prediction accuracy on highly heterogeneous databases. Boosting localized experts also significantly reduces the number of boosting iterations needed for achieving maximal prediction accuracy.

Although boosting localized experts required order of magnitude less boosting rounds to achieve the maximum prediction accuracy than the standard and adaptive attribute boosting, the number of constructed prediction models increases drastically through the iterations. This number depends on the number of discovered clusters and on the number of boosting rounds needed for making the final classifier. In our case, this drawback was alleviated by the fact that we were experimenting with small numbers of clusters (4, 5) and that only a few boosting iterations were sufficient to maximize the prediction accuracy. Therefore, the memory needed for storing all prediction models is comparable or even less than for the standard boosting technique.

In addition to the prediction accuracy of the proposed method, the time required for building the model is also an important issue when developing a novel algorithm. Albeit the number of learned classifiers per iteration for the proposed method was much larger than for the standard boosting, the cluster data sets on which the classification models were built were smaller. The computation time for learning by the proposed model therefore was comparable to learning the models on the entire training data. Hence, the total computation time depends only on the number of iterations, and is much smaller for the proposed boosting localized experts than for the standard boosting or the adaptive attribute boosting.

Although the performed experiments provide evidence that the proposed approaches can improve predictions of classifier ensembles, further work is needed to examine the method for more heterogeneous data sets with more diverse attributes. We are currently working on extending the combining of the adaptive attribute boosting and the boosting localized experts such that other attribute representation methods (attribute extraction, attribute weighting) are applied on each cluster discovered during the boosting rounds. Furthermore, identifying attributes using supervised learning may not be appropriate for performing clustering algorithm. Therefore, finding the smallest attribute subsets that best uncover “natural” groupings (clusters) from the data according to some criterion is needed [27]. We are also investigating modifying the proposed algorithm for spatial data sets in which observations close to each other are more likely to be similar than observations widely separated in space.

The other classification models (C4.5 decision trees, k-Nearest Neighbors) will also be examined in order to further improve the generalization capabilities of the proposed method. In addition, we are working to extend the method to regression based problems.

6 References

1. A. LAZAREVIC, X. XU, T. FIEZ, Z. OBRADOVIC, *Clustering-Regression-Ordering Steps for Knowledge Discovery in Spatial Databases*, In Proceedings of IEEE/INNS International Conference on Neural Networks, 1999, No. 345, Session 8.1B.
2. L. BREIMAN, *Bagging predictors*, Machine Learning 24 (1996), pp. 123-140.
3. Y. FREUND, AND R. E. SCHAPIRE, *Experiments with a new boosting algorithm*, In Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 325-332.
4. E. B. KONG, T. DIETTERICH, *Error-correcting output coding corrects bias and variance*, In Proceedings of the twelfth National Conference on Artificial Intelligence, 1996, pp. 725-730.
5. A. LAZAREVIC, AND Z. OBRADOVIC, *Knowledge Discovery in Multiple Spatial Databases*, submitted to Journal of Neural Computing and Applications, 2000.
6. L. BREIMAN, *Arcing the edge*, Technical Report 486, Statistics Department, University of California, 1997.
7. G. RATSCH, T. ONODA, AND K. R. MULLER, *Regularizing AdaBoost*. In M. KEARNS, A. SMOLLA AND COHN (Eds.), Advances in Neural Information Processing Systems, 11 (1998), MIT Press, pp. 564-570.

8. J. FRIEDMAN, T. HASTIE, R. TIBSHIRANI, *Additive Logistic Regression: A Statistical View of Boosting*, The Annals of Statistics, 38(2000), pp. 337-374.
9. L. MASON, J. BAXTER, P. BARTLETT, AND M. FREAN, *Function Gradient Techniques for combining hypotheses*, In A. SMOLA, P. BARTLETT, B. SCHOLKOPF, AND D. SCHUURMANS, (Eds.), Advances in Large Margin Classifiers, MIT Press, 2000.
10. L. LIU, AND H. MOTODA, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.
11. A. LAZAREVIC, T. FIEZ, Z. OBRADOVIC, *Adaptive Boosting for Spatial Functions with Unstable Driving Attributes*, In Proceedings of Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 2000, pp. 329-340.
12. J. O'SULLIVAN, J. LANGFORD, R. CARUNA, A. BLUM, *FeatureBoost: A Meta-Learning Algorithm that Improves Model Robustness*, In Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 703-710.
13. K. TUMER, AND J. GHOSH, *Error correlation and error reduction in ensemble classifiers*, Connection Science 8(1996), pp. 385-404.
14. K. J. CHERKAUER, *Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks*, In P. CHAN (Ed.): Working Notes of the AAAI Workshop on Integrating Multiple Learned Models, 1996, pp. 15-21.
15. D. OPITZ, *Feature Selection for Ensembles*, In Proceedings of Sixteenth National Conference on Artificial Intelligence (AAAI), 1999, pp. 379-384.
16. L. KUNCHEVA, J. BEZDEK, R. DUIN, *Decision Templates for Multiple Classifier Fusion: An Experimental Comparison*, Pattern Recognition, 34(2001), pp. 299-314.
17. R. AVNIMELECH, N. INTRATOR, *Boosting Mixture of Experts: An Ensemble Learning Scheme*, Neural Computation, 11(1999), pp. 475-490.
18. M. JORDAN, R. JACOBS, *Hierarchical Mixture of Experts and the EM Algorithm*, Neural Computation, 6(1994), pp. 181-214.
19. H. SCHWENK, Y. BANGIO, *Boosting Neural Networks*, Neural Computation, 12(1999), pp. 1869-1887.
20. A. LAZAREVIC, D. POKRAJAC, AND Z. OBRADOVIC, *Distributed Clustering and Local regression for Knowledge Discovery in Multiple Spatial Databases*, In Proceedings of 8th European Symposium on Artificial Neural Networks, 2000, pp. 129–134.
21. J. SANDER, M. ESTER, H-P. KRIEGEL, X. XU, *Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications*, Data Mining and Knowledge Discovery, Kluwer Academic Publishers, 2(1998), pp. 169-194.
22. L. KAUFMAN, P. J. ROUSSEEUW, *Finding groups in data: an introduction to cluster analysis*, John Willey, New York, 1990.
23. M. RIEDMILLER, H. BRAUN, *A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm*, In Proceedings of the IEEE International Conference on Neural Networks, 1993.
24. M. HAGAN, M. MENHAJ, *Training feedforward networks with the Marquardt algorithm*, IEEE Transactions on Neural Networks 5(1994), pp. 989-993.
25. D. POKRAJAC, T. FIEZ, Z. OBRADOVIC, *A Spatial Data Simulator for Agriculture Knowledge Discovery Applications*, in review.
26. D. MARGINEANTU, AND T. DIETTERICH, *Pruning adaptive boosting*, In Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 211-218.
27. J. DY, AND C. BRODLEY, *Feature Subset Selection and Order Identification for Unsupervised Learning*, In Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 247-254.