Adaptive Boosting Techniques in Heterogeneous and Spatial Databases

Aleksandar Lazarevic, Zoran Obradovic
Center for Information Science and Technology,
Temple University, Room 303, Wachman Hall (038-24),
1805 N. Broad St., Philadelphia, PA 19122, USA,
aleks@ist.temple.edu, zoran@ist.temple.edu
phone: (215) 204 – 6265, fax: (215) 204 - 5082

Abstract. *Combining multiple classifiers is an effective technique for improving classification accuracy by reducing the variance through manipulating the training data distributions. In many large-scale data analysis problems involving heterogeneous databases with attribute instability, however, standard boosting methods do not improve local classifiers (e.g. k-nearest neighbors) due to their low sensitivity to data perturbation. Here, we propose an adaptive attribute boosting technique to coalesce multiple local classifiers each using different relevant attribute information. To reduce the computational costs of k-nearest neighbor (k-NN) classifiers, a novel fast k-NN algorithm is designed. We show that the proposed combining technique is also beneficial when boosting global classifiers like neural networks and decision trees. 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. Finally, when heterogeneous data sets contain several homogeneous data distributions, we propose a new technique of boosting specialized classifiers, where instead of a single global classifier for each boosting round, there are specialized classifiers responsible for each homogeneous region. The number of regions is identified through a clustering algorithm performed at each boosting iteration. New boosting methods applied to synthetic spatial data and real life spatial data show improvements in prediction accuracy for both local and global classifiers when unstable driving attributes and heterogeneity are present in the data. In addition, boosting specialized experts significantly reduces the number of iterations needed for achieving the maximal prediction accuracy.*

Keywords: Adaptive attribute boosting, Spatial boosting, Clustering, Boosting specialized experts, Heterogeneous spatial databases

1. Introduction

In contemporary data mining, many real world knowledge discovery problems involve the investigation of relationships between attributes in heterogeneous data sets where rules identified among the observed attributes in certain subsets do not apply elsewhere. A heterogeneous data set can be partitioned into homogeneous subsets such that learning a local model separately on each of them results in improved overall prediction accuracy. In addition, many large-scale data sets very often exhibit attribute instability, which means that the set of relevant attributes that describes data examples is not the same through the entire data space. This is especially true in spatial databases, where different spatial regions may have completely different characteristics [18].

It is well known in machine learning theory that a combination of many different predictors can be an effective technique for improving prediction accuracy. There are many general combining algorithms such as bagging [5], boosting [9], or Error Correcting Output Codes (ECOC) [15] that significantly improve global classifiers like decision trees, rule learners, and neural networks. These algorithms may manipulate the training patterns used by individual classifiers (bagging, boosting) or the class labels (ECOC). In most cases, the importance of different classifiers is the same for all of 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. To make the ensemble of classifiers for heterogeneous databases more accurate, instead of applying a global classification model across entire data sets, the models are varied to better match specific needs of the subsets thus improving prediction capabilities [21]. In such an approach, there is a specialized classification expert responsible for each region which strongly dominates the others from the pool of specialized experts.

On the other hand, diversity is required to ensure that all the classifiers do not make the same errors. In order to increase the diversity of combined classifiers for heterogeneous spatial 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 for constructing single classifiers in every iteration. In addition, the application of different classifiers on spatial databases, where the data are highly spatially correlated, may produce spatially correlated errors [19]. In such situations, standard combining methods might require different schemes for manipulating the training instances in order to maintain classifier diversity.

In this paper, we extend the framework for constructing multiple classifier system using the AdaBoost algorithm [9]. In our approach, we first try to maximize local specific information for a drawn sample by changing the attribute representation using attribute selection, attribute extraction and appropriate attribute weighting methods [22] at each boosting iteration. Second, in order to exploit the spatial data knowledge, a modification of the boosting method appropriate for heterogeneous spatial databases is proposed, where, at each boosting round, spatial data blocks are drawn instead of sampling single instances like in the standard approach. Finally, the maximal gain by emphasizing local information, especially for highly heterogeneous data sets, was achieved by allowing the weights of the different weak classifiers to depend on the input. Rather than having constant weights of the classifiers for all data patterns (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 specialized classification experts on each of these regions which describe the relationship between the data characteristics and the target class [18]. Instead of a single classifier built on a sample drawn in each boosting iteration, there are several specialized classification experts responsible for each of the local 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 combining the classification experts.

The influence of all of these adjustments is not the same, however, for local classifiers [4] (e.g. k–nearest neighbors, radial basis function networks) and global classifiers (e.g. decision trees and artificial neural networks). It is known that standard combining methods do not improve simple local classifiers due to correlated predictions across the outputs from multiple combined classifiers [5, 15]. We show that, by selecting different attribute representations for each sample, prediction of combined nearest neighbor as well as global classifiers can be considerably decorrelated. Our experimental results indicate that sampling spatial data blocks during boosting iterations is beneficial only for local but not for global classifiers. Further significant improvements in prediction accuracy obtained by building specialized classifiers responsible for local regions show that this method seems to be slightly more beneficial for k-nearest neighbor algorithms than for global classifiers, although the total prediction accuracy was significantly better when combining global classifiers.

The nearest neighbor classifier is often criticized for slow run-time performance and large memory requirements, and using multiple nearest neighbor classifiers could further worsen the problem. Therefore, we used a novel fast method for k-nearest neighbor classification to speed up the boosting process.

In Section 2, we discuss current ensemble approaches and work related to specialized experts and changing attribute representations of combined classifiers. Section 3 describes the proposed methods and investigates their advantages and limitations. In Section 4, we evaluate the proposed methods on three synthetic and one real-life data set comparing it with standard boosting and other methods for dealing with heterogeneous spatial databases. Finally, section 5 concludes the paper and suggests further directions in current research.

2. Classifier Ensembles

2.1. Ensembles of Local Learning Algorithms

One of the oldest and simplest methods for performing general, non-parametric classification that belongs to the family of local learning algorithms [4] is a k-nearest neighbor classifier (k-NN) [7]. Despite its simplicity, the k-NN classifier can often provide similar accuracy to more sophisticated methods such as decision trees or neural networks. Its advantages include the ability to learn from a small set of examples, and to incrementally add new information at runtime.

Many general algorithms for combining multiple versions of a single classifier do not improve the k-NN classifier at all. For example, when experimenting with bagging, Breiman [5] found no difference in accuracy between the bagged k-NN classifier and the single model approach. Kong and Dietterich [15] also concluded that ECOC would not improve classifiers that use local information due to high error correlation.

A popular alternative to bagging is boosting, which uses adaptive sampling of patterns to generate the ensemble. In boosting [9], the classifiers in the ensemble are trained serially, with the weights on the training instances set adaptively according to the performance of the previous classifiers. The main idea is that the classification algorithm should concentrate on the difficult instances. Boosting can generate more diverse ensembles than bagging does, due to its ability to manipulate the input distributions. However, it is not clear how one should apply boosting to the k-NN classifier for the following reasons: (1) boosting stops when a classifier obtains 100% accuracy on the training set, but this is always true for the k-NN classifier, (2) increasing the weight on a hard to classify instance does not help to correctly classify that instance as each prototype can only help classify its neighbors, not itself. Freund and Schapire [9] applied a modified version of boosting to the k-NN classifier that worked around these problems by limiting each classifier to a small number of prototypes. However, their goal was not to improve accuracy, but to improve speed while maintaining current performance levels.

Although there is a large body of research on multiple model methods for classification, very little specifically deals with combining k-NN classifiers. Ricci and Aha [31] applied ECOC to the k-NN classifier (NN-ECOC). Normally, applying ECOC to k-NN would not work since the errors would be correlated across the binary learning problems. However, they found that applying attribute selection to the two-class problems decorrelated errors if different attributes were selected. Unlike this approach, Bay's Multiple Feature Subsets (MFS) method [3] uses random attributes when combining individual classifiers by simple voting. Each time a pattern is presented for classification, a new random subset of attributes is selected for each classifier. He used two different sampling functions: sampling with replacement, and sampling without replacement. Each of the k-NN classifiers uses the same number of attributes.

Some researchers developed techniques for reducing memory requirements for k-NN classifiers by their combining. In combining condensed nearest neighbor (CNN) classifiers [1], the size of each classifier's prototype set is drastically reduced in order to destabilize the k-NN classifier. Bootstrap or disjoint data set partitioning was used in combination with CNN classifiers to edit and reduce the prototypes. In Voting nearest neighbor subclassifiers [16], three small groups of examples are selected such that each k-NN subclassifier, when used on them, errs in a different part of the instance space. Simple voting may then correct many failures of individual subclassifiers.

2.2. Ensemble of Global Learning Algorithms

There has been a very significant movement during the past decade to combine the decisions of global classifiers (e.g. decision trees, neural networks), and a significant body of literature on this topic has been produced. All combining methods are results of two parallel lines of study: (1) multiple classifier systems that attempt to find an optimal combination of the decisions from a given set of carefully designed global classifiers; and (2) specialized classifier systems that build mutually complementary classification experts, each responsible for a particular data subset, and then merge them together. Although it is known that multiple classifier systems work well with global classifiers like neural networks, there have been several experiments in selecting different attribute subsets as an attempt to force the classifiers to make different and hopefully uncorrelated errors when analyzing heterogeneous databases.

FeatureBoost [26] 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 [19]. The technique coalesces multiple local classifiers each using different relevant attribute information. The related attribute representation is changed through attribute selection, extraction and weighting processes performed at each boosting round. This method was mainly motivated by the fact that standard combining methods do not improve local classifiers (e.g. k-NN) 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 selecting different attribute 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 [35] 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 [6] 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.

Motivated by the problem of how to avoid overfitting a set of training data when using decision trees for classification, Ho [12] proposed a "*decision forest*", an ensemble of decision trees constructed systematically by autonomously and pseudorandomly selecting a small number of dimensions from a given attribute space. The decisions of individual trees are combined by averaging the conditional probability of each class at the leaves. The method maintains high accuracy on the training data and, compared with single tree classifiers, improves on the generalization accuracy as it grows in complexity.

Opitz [25] 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 the combination of accuracy and diversity.

DynaBoost [24] is an extension of the AdaBoost algorithm that allows an input-dependent combination of the base hypotheses. A separate weak learner is used for determining the input dependent weights of each hypothesis. The error function minimized by these additional weak learners is a margin cost function that is also minimized by AdaBoost. Although the weights depend on the input, there is still a single hypothesis per iteration that needs to be combined.

Several approaches belonging to specialized classifier systems have also appeared lately. Our recent approach [21] 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 [17]. 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 [2]. This algorithm applies principles of both boosting and Mixture of Experts [13] 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 the Boosting algorithm. As a variant of the Mixture of Experts, it can be made appropriate for general classification and regression problems, by initializing the partition of the data set to different experts in a boosting like manner. If viewed as a variant of the Boosting algorithm, it uses a dyn amic model for combining the outputs of the classifiers.

3. Methodology

3.1 Adaptive Attribute Boosting

The adaptive attribute boosting algorithm we present here is a variant of the AdaBoost.M2 procedure [9]. The proposed algorithm, shown in Figure 1, proceeds in a series of T rounds. In every round a weak learning algorithm is called and presented with a different distribution D_t altered not only by emphasizing particular training examples, but also by emphasizing particular attributes. The distribution is updated to give wrong classifications higher weights than correct classifications. The entire weighted training set is given to the weak learner to compute the weak hypothesis h_t. At the end, the different hypotheses are combined into a final hypothesis h_{fn}.

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 modify the standard algorithm by adding **step 0**, wherein we choose a different attribute representation for each sample. Different attribute representations are realized through attribute selection, attribute extraction and attribute weighting processes through boosting iterations. This is an attempt to force individual classifiers to make different and hopefully uncorrelated errors.

- Given: Set S $\{(x_1, y_1), \dots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \dots, k\}$
  - Let $B = \{(i, y): i \in \{1,2,3,4,\dots m\}, y \neq y_i\}$
  - Initialize the distribution D_1 over the examples, such that $D_1(i) = 1/m$.
- For t = 1, 2, 3, 4, … T

  0. *Find relevant feature information for distribution D_t using supervised attribute selection*
  1. Train weak learner using distribution D_t
  2. Compute weak hypothesis h_t: $X \times Y \rightarrow [0, 1]$
  3. Compute the pseudo-loss of hypothesis h_t:

     $$\varepsilon_t = \frac{1}{2} \cdot \sum_{(i,y)\in B} D_t(i, y)(1 - h_t(x_i, y_i) + h_t(x_i, y))$$

  4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$
  5. Update D_t : $D_{t+1}(i, y) = (D_t(i, y) / Z_t) \cdot \beta_t^{(1/2)\cdot(1 - h_t(x_i, y) + h_t(x_i, y_i))}$
     where Z_t is a normalization constant chosen such that D_{t+1} is a distribution.

- Output the final hypothesis: $h_{fn} = \arg\max_{y\in Y} \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) \cdot h_t(x, y)$

Figure 1. The adaptive attribute boosting with performing attribute selection at step 0 in each boosting iteration

To eliminate irrelevant and highly correlated attributes, regression-based attribute selection is performed through performance feedback forward selection and backward elimination search techniques [22] based on linear regression mean square error (MSE) minimization. The r most relevant attributes are selected according to the selection criterion at each round of boosting, and are used by the single classifiers. In addition, attribute extraction procedure is performed through Principal Components Analysis (PCA) [10]. Each of the single classifiers uses the same number of new transformed attributes. Another possibility is to choose an appropriate number of newly transformed attributes that will retain some predefined part of the variance.

The attribute weighting method for the proposed technique is used only for local classifiers (k -NN) and is based on a 1-layer feedforward neural network. First, we try to perform target value prediction for the drawn sample with a defined 1-layer feedforward neural network using all attributes. It turns out that this kind of neural network can discriminate relevant from irrelevant attributes. Therefore, the neural networks interconnection weights are taken as attribute weights for the k-NN classifier.

To further experiment with attribute stability properties, miscellaneous attribute selection algorithms [22] are applied to the entire training set and the most stable attributes are selected. These attributes are then used by the standard boosting method. When applying adaptive attribute boosting, in order to compare the most stable selected attributes, the attribute occurrence frequency is monitored at each boosting round. When attribute subsets selected through boosting rounds become stable, this is an indication to stop the boosting process.

3.1.1 Adaptive Attribute Boosting for k-NN Classifier

Nearest neighbors are stable to the data perturbation, so bagging and boosting generate poor k -NN ensembles. However, they are extremely sensitive to the attributes used. Our approach attempts to use this instability to generate a diverse set of local classifiers with uncorrelated errors. At each boosting round, we perform one of the methods for changing attribute representation, explained above, to determine a suitable attribute space for use in classification. When determining the least distant instances, we consider standard Euclidean distance and Mahalanobis distance.

To speed up the long-lasting boosting process, a fast k-NN classifier is proposed. For n training examples and d attributes our approach requires preprocessing which takes O($d \cdot n \cdot log\ n$) steps to sort each attribute separately. However, this is performed only once, and we trade off this initial time for later speedups.

Initially, we form a hyper-rectangle with boundaries defined by the extreme values of the k closest values for each attribute (Figure 2 – small dotted lines). If the number of training instances inside the identified hyper-rectangle is less than k, we compute the distances from the test point to all of $d \cdot k$ data points which correspond to the k closest values for each of d attributes, and sort them into a non-decreasing array sx. We take the nearest training example cdp with the distance dst_{min}, and form a hypercube with boundaries defined by this minimum distance dst_{min} (Figure 2 - larger dotted lines). If the hypercube does not contain enough data, i.e. k training points, form the hypercube of a side $2 \cdot sx(k+1)$. Although this hypercube contains more than k training examples, we need to find the one which contains the minimal number of training examples greater than k. Therefore, if needed, we search for a minimal hypercube by binary halving the index in the non-decreasing array sx. This can be executed at most $log\ k$ times, since we are reducing the size of the hypercube from $2 \cdot sx(k+1)$ to $2 \cdot sx(1)$. Therefore the total time complexity of our algorithm is $O(d \cdot log\ k \cdot log\ n)$, under the assumption that $n > d \cdot k$, which is always true in practical problems.

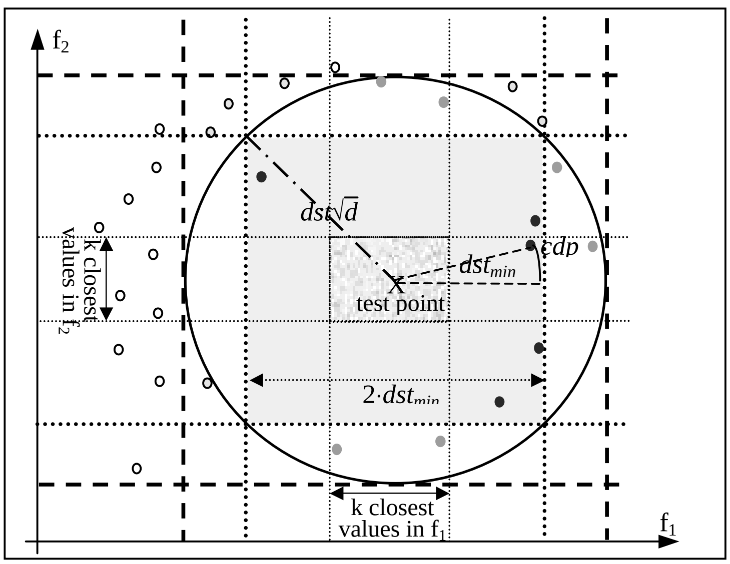


Figure 2. The used hyper-rectangle, hypersphere and hypercubes in the fast k-NN

If the number of training instances inside the identified hyper-rectangle (Figure 2 - small dotted lines) is larger than k, we also search for a minimal hypercube that contains at least k and at most $2 \cdot k$ training instances inside that hypercube. This is accomplished by binary halving or by incrementing a side of the hypercube. After each modification of the hypercube's side, we compute the number of enclosed training instances and modify the hypercube accordingly. Analogously to the previous case, it can be shown that binary halving or incrementing the hypercube's side will not take more than $log\ k$ time, and therefore the total time complexity is still $O(d \cdot log\ k \cdot log\ n)$.

When we find a hypercube which contains the appropriate number of points, it is not necessary that all k nearest neighbors are in the hypercube, since some of the closer training instances to the test points could be located in a hypersphere of identified radius $dst\sqrt{d}$ (Figure 2). Since there is no fast way to compute the number of instances inside the sphere without computing all the distances, we embed the hypersphere in a minimal hypercube (Figure 2 – dashed lines) and compute the number of training points inside this surrounding hypercube. The number of points inside the surrounding hypercube is much less than the total number of training instances and therefore speedups our algorithm.

3.1.2 Adaptive Attribute Boosting for Global Classifiers

Although standard boosting can increase the prediction accuracy of global classifiers like neural networks [34] and decision trees [30], we change attribute representation to see if adaptive attribute boosting can further improve accuracy of an ensemble of global classifiers. The most stable attributes used in standard boosting of k-NN classifiers are also used here for the same purpose.

We train multilayer (2-layered) feedforward neural network classification models with the number of hidden neurons equal to the number of input attributes. The neural network classification models have the number of output nodes equal to the number of classes, where the predicted class is from the output with the largest response. We used two learning algorithms: resilient propagation [32] and Levenberg-Marquardt [11]. For a decision tree model, we used the ID3 learning algorithm [29] which employs the information gain criterion to choose which attribute to place at the root of each decision tree and subtree. After the trees are fully grown, a pruning phase replaces subtrees with leaves using the same predefined pruning factor for all trees.

3.2 Spatial Boosting

Spatial data represent a collection of attributes whose dependence is strongly related to spatial location; observations close to each other are more likely to be similar than observations widely separated in space. Explanatory attributes, as well as the target attribute in spatial data sets are very often highly spatially correlated. As a consequence, applying different classification techniques on such data is likely to produce errors that are also spatially correlated [27]. Therefore, when applied to spatial data, the boosting method may require different partitioning schemes than simple weighted selection that does not take into account the spatial properties of the data.

The proposed spatial boosting method (Figure 3) starts with partitioning the spatial data set into the spatial data blocks (squares of size M points $\times$ M points). Rather than drawing n data points according to the distribution D_t (Figure 1), the proposed method draws only $\lfloor n/M^2 \rfloor$ data points according to the distribution P_t (Figure 3). Since each of drawn examples belongs exactly to one of the partitioned spatial data blocks, the proposed method defines $\lfloor n/M^2 \rfloor$ belonging spatial data blocks and merges them into a set used for learning a weak classifier. Like in standard boosting, the distribution P_t is also updated to give wrong classifications higher weights than correct classifications, but due to spatial correlation of data, at the end of each boosting round simple median M $\times$ M filtering is applied over the entire data distribution P_t. Using this approach we hope to achieve more decorrelated classifiers whose integration can further improve model generalization capabilities for spatial data. The spatial boosting technique was applied to both local (k-NN) and global (neural network, decision trees) classifiers

- Given set S $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \ldots, k\}$ is split into $\lfloor n/M^2 \rfloor$ squares of size M x M points. Let $B = \{(i, y): i \in \{1,2,3,4,\ldots m\}, y \neq y_i\}$
- Initialize the distribution P_1 over the examples, such that $P_1(i) = 1/m$.
- For t = 1, 2, 3, 4, … T
  1. According to distribution P_t draw $\lfloor n/M^2 \rfloor$ data points that uniquely determine belonging spatial data blocks.
  2. Train a weak learner on a set containing all belonging spatial data blocks.
  3. Compute weak hypothesis h_t: $X \times Y \to [0, 1]$
  4. Compute the pseudo-loss of hypothesis h_t: $\varepsilon_t = \frac{1}{2} \cdot \sum_{(i,y)\in B} P_t(i, y)(1 - h_t(x_i, y_i) + h_t(x_i, y))$
  5. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$
  6. Update P_t : $P_{t+1}(i, y) = (P_t(i, y) / Z_t) \cdot \beta_t^{(1/2)\cdot(1 - h_t(x_i, y) + h_t(x_i, y_i))}$ where Z_t is a normalization constant chosen such that D_{t+1} is a distribution. Apply median M $\times$ M filtering to the distribution P_t.
- Output the final hypothesis: $h_{fn} = \arg\max_{y \in Y} \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) \cdot h_t(x, y)$

Figure 3. The spatial boosting algorithm with drawing spatial data blocks at each boosting round

3.3 Boosting Specialized Classifiers

Although previous boosting modifications improve generalizability of final predictors, it seems that in heterogeneous databases where several more homogeneous regions exist boosting does not enhance the prediction capabilities as well as for homogeneous databases [19]. In such cases it is more useful to have several local experts responsible for each region of the data set. A possible approach to this problem is to cluster the data first and then to assign a single classifier to each discovered cluster. Boosting specialized classifiers, described in Figure 4, 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.

- 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 using unsupervised wrapper approach around clustering algorithm.
2. Obtain c distributions $D_{t,j}$, j = 1, …c and corresponding sets (clusters) $S_{t,j} = \{(x_{1,j}, y_{1,j}), \ldots, (x_{m_j,j}, y_{m_j,j})\}$ $x_{i,j} \in X_j$, with labels $y_{i,j} \in Y_j = \{1, \ldots, k\}$ by applying clustering with the most relevant attributes 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 attribute representation for clusters $S_{t,j}$ using supervised feature selection
   3.2. Train weak learners $L_{t,j}$ on the sets $S_{t,j}$, $j = 1,\ldots c$.
   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 $S_{t,j}$ 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.

- 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 4. The scheme for boosting specialized classifiers with performing attribute selection algorithm wrapped around clustering (step 1) in each boosting iteration.

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. The proposed boosting algorithm achieves greater flexibility by building classifiers that operate only in specialized regions and have local weights $\beta_t(x)$ that depend on the point x where they are applied.

In order to partition the spatial data set into these localized regions, two clustering algorithms are employed. The first is the standard k-means algorithm [14]. Here, data set S = {(x_1, y_1), … , (x_m, y_m)}, $x_i \in X$, is partitioned into k clusters by finding k points $\{m_j\}_{j=1}^{k}$ such that

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

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

The second clustering algorithm called DBSCAN relies on a density-based notion of clusters and was designed to discover clusters of an arbitrary shape [33]. The key idea of a density-based cluster 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), since the typical density of points inside clusters is considerably higher than outside of clusters. Unlike the cluster centroids in the k-means, here the centers of the clusters can be outside of the clusters due to their arbitrary shapes. Therefore, we define cluster medoids, the cluster *core* objects closest to the cluster centroids.

Since our boosting specialized experts involves clustering at step 1, there is a need to find a small subset of attributes that uncover "natural" groupings (clusters) from the data according to some criterion. For this purpose, we adopt the wrapper framework in unsupervised learning [8], where we apply the clustering algorithm to each attribute subset in the search space and then evaluate the attribute subset by a criterion function that utilizes the clustering result. If there are d attributes in a data set, an exhaustive search of the 2^d possible attribute subsets for the one that maximizes our selection criterion is computationally intractable. Therefore, in our experiments, fast sequential forward selection search is applied.

Like in [8] we also accept the scatter separability trace ($S_w^{-1} S_b$) for attribute selection criterion, where S_w is the within-class scatter matrix and S_b is the between scatter matrix. S_w measures the average covariance of each cluster and how scattered the samples are from their cluster medoids in the case of DBSCAN clustering, or from their cluster means in the case of k-means clustering. S_b measures how the cluster means or medoids are distant from the total mean. Larger the value of the trace ($S_w^{-1} S_b$) results in larger the normalized distance between the clusters and therefore in better cluster discrimination.

This procedure, performed at step 1 of every boosting iteration, results in r most relevant attributes for clustering. Thus, for each round of boosting, there are different relevant attribute subsets that are responsible for distinguishing among homogeneous regions existing in a drawn sample. As a result of the clustering, applied to find those homogeneous regions, several distributions $D_{t,j}$ ($j = 1,\ldots,c$) are obtained, where c is the number of discovered clusters. For each of c clusters $S_{t,j}$ discovered in the data sample, we identify its most relevant attributes, train a weak learner $L_{t,j}$ using a distribution $D_{t,j}$ and compute a weak hypothesis $h_{t,j}$. Furthermore, for every cluster $S_{t,j}$, we identify its convex hull $H_{t,j}$ in the attribute space used for clustering, and map these convex hulls to the entire training set in order to find the corresponding clusters $T_{t,j}$ (Figure 5) [20]. Data points inside the convex hull $H_{t,j}$ belong to cluster $T_{t,j}$, and 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 $L_{t,j}$ and each of them is applied to the corresponding cluster $T_{t,j}$.

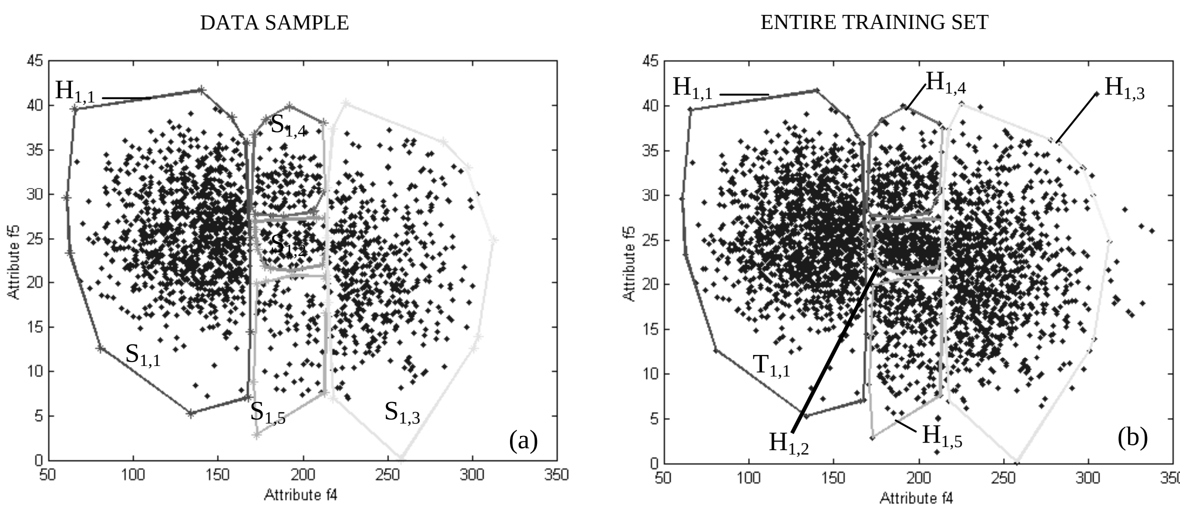


Figure 5. Mapping convex hulls $H_{1,j}$ of clusters $S_{1,j}$ discovered in the data sample to the entire training set in order to find corresponding clusters $T_{1,j}$. For example, all data points inside the contours of the convex hull $H_{1,1}$ (corresponding to the cluster $S_{1,1}$) belong to the new cluster $T_{1,1}$ identified on the entire training set.

In our boosting specialized classifiers, data points from different clusters have different pseudo-loss values and different parameter values β_t. For each cluster $T_{t,j}$, ($j = 1,\ldots,c$) (Figure 5) defined with the convex hull $H_{t,j}$, there is a pseudo-loss $\varepsilon_{t,j}$ and the corresponding parameter $\beta_{t,j}$. Both the pseudo-loss value $\varepsilon_{t,j}$ and parameter $\beta_{t,j}$ are computed independently for each cluster $T_{t,j}$ where a particular classifier $L_{t,j}$ is responsible. 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 $\beta_{t,j}$ into β_t and $h_{t,j}$ into h_t, updating the distribution D_t can be performed as in standard boosting. However, the local classifiers from each round are first applied to the corresponding clusters and integrated into a composite classifier responsible for that round. The composite classifiers are then combined into the final hypothesis using the AdaBoost.M2 algorithm.

When performing clustering during boosting iterations, it is possible that some of the discovered clusters have an insufficient number of data points needed for training a specialized classifier. This number of data patterns is defined as a function of the number of patterns in the entire training set. Several techniques for handling this scenario are considered.

The first technique denoted as ***simple*** halts the boosting process every time a cluster with an insufficient size is detected. When the boosting procedure is terminated, only the classifiers from the previous iterations are combined in order to create the final hypothesis h_{fn}. More sophisticated techniques do not stop the boosting process, but instead of training the specialized classifier on an insufficiently large cluster, they employ the specialized classifiers constructed in previous iterations. When an insufficiently large cluster is identified, its corresponding cluster from previous iterations is detected using the convex hull matching (Figure 5) and the model constructed on the corresponding cluster is applied to the cluster discovered in the current iteration. To determine this model, the most effective method (***best_local***) takes the classifier constructed in the iteration where the *local* prediction accuracy for the corresponding cluster was maximal. In two similar techniques, 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 from the iteration where the *global* prediction accuracy was maximal. In all of these sophisticated techniques, the boosting procedure ceases when the pre-specified number of iterations is reached or there is a significant drop in the prediction accuracy for the training set.

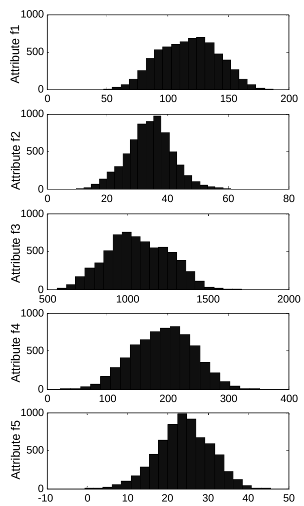
Furthermore, drawing spatial data blocks in boosting iterations, employed in the spatial boosting technique, is also integrated in boosting specialized classifiers.

4. Experimental Results

Our experiments were first performed on three synthetic data sets generated using our spatial data simulator [28] such that the distributions of generated data resembled the distributions of real life spatial data. All data sets had

6561 patterns with five relevant (f1,...f5) and five irrelevant attributes (f6,...,f10) and three equal size classes. The first data set stemmed from homogeneous distribution, while the second one was heterogeneous containing five homogeneous data distributions. In heterogeneous data set, the attributes f4 and f5 were simulated to form five clusters in their attribute space (f4, f5) using the technique of feature agglomeration [28]. Furthermore, instead of using a single 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. The degree of relevance was also different for each distribution. The histograms of all five attributes for homogeneous data set as well as for heterogeneous data set with five distributions are shown in Figures 6a and 6b respectively. When applying boosting specialized classifiers, we also experimented with the heterogeneous data set where the one of attributes relevant for clustering was missing only during clustering process, while all attributes were available when training specialized classifiers.

(a)



(b)

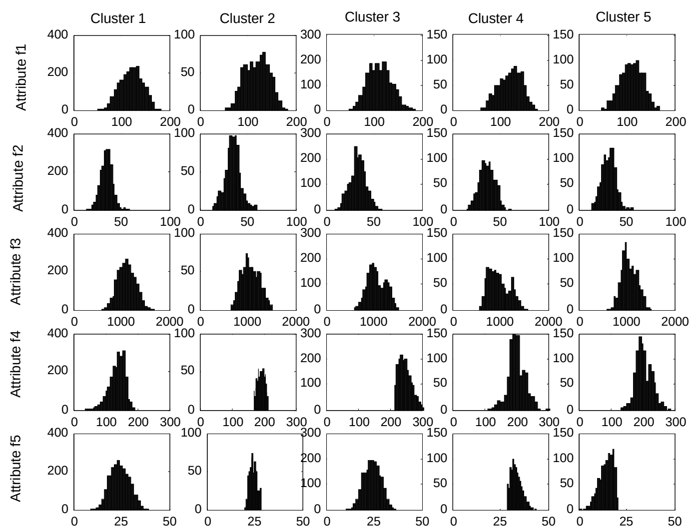


Figure 6. Histograms of all five relevant attributes for a) homogeneous synthetic spatial data set b) heterogeneous synthetic spatial data set with five clusters

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.

For all performed experiments, synthetic and real life data sets were split into training and test data sets. The all reported classification accuracies were achieved on test data by averaging over 10 trials of the boosting algorithm.

For synthetic data sets, we first performed standard boosting and adaptive attribute boosting (Figure 7) for both local (k -NN classifiers) and global (neural networks and decision trees) classifiers. For the k -NN classifier experiments, the value of k was set using cross validation performance estimates on the entire training set. For boosting neural network classifiers, we used the model defined in section 3.1.2., and the best prediction accuracies were achieved using the Levenberq-Marquardt algorithm for training neural networks. For boosting ID3 decision trees, we used a post-pruning with a small constant pruning factor such that the pruned trees were smaller than the original ones for approximately 20%.

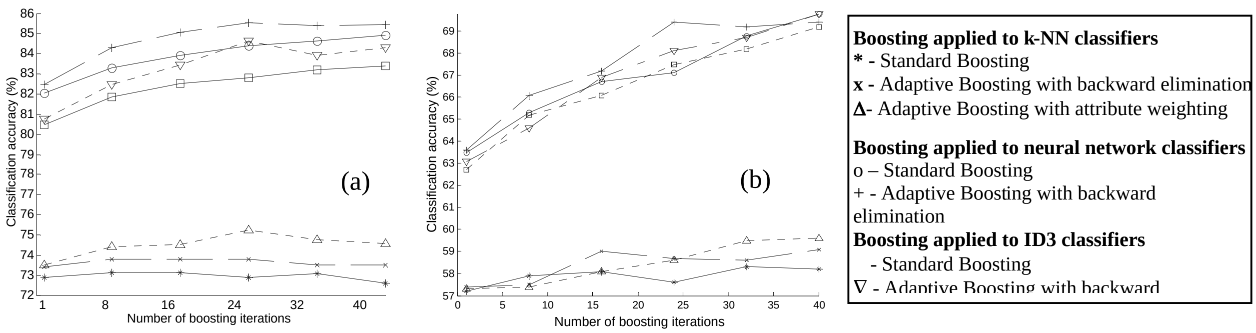


Figure 7. Overall averaged classification accuracies (%) for the 3 equal-size class problems on (a) homogeneous synthetic test data set (b) heterogeneous synthetic test data set with five clusters defined by 2 of 5 relevant attributes.

Analyzing the charts in Figure 7, it was evident that the method of adaptive attribute boosting applied to local and global classifiers showed only minor improvements in prediction accuracy for both synthetic data sets. For the homogeneous data set this was because there were no differences in relevant attributes through the training set, while for the heterogeneous data set this was due to the fact that each spatial region not only had 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 four or five most important attributes at each boosting round, since selecting the three most relevant attributes may be insufficient for successful learning. Since the total number of relevant attributes in the data set was five as well, we selected the four most relevant attributes for adaptive attribute boosting, knowing that for some drawn samples we would lose beneficial information. Due to these facts concerning deficient attribute instability, the selected attributes during the boosting iterations were not monitored.

In the standard boosting method, we used all five 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 total number of the boosting rounds. Instead of post-pruning the boosted classifiers [23] we can try to set the appropriate number of boosting iterations at the beginning of the procedure.

Applying the spatial boosting method to a k-NN classifier, we achieved much better prediction than with the adaptive attribute boosting methods on a k-NN classifier (Table 1). Furthermore, when applying spatial boosting with attribute selection at each round, the prediction accuracy was increased slightly as the size (M) of the spatial block was increased (Table 1). No such improvements were noticed for spatial boosting with fixed attributes or with the attribute weighting method, and therefore the classification accuracies for only M = 5 are given.

Applying spatial boosting on global classifiers (neural networks and decision tree) resulted in no enhancements in classification accuracies. Moreover, for pure spatial boosting without attribute selection we obtained slightly worse classification accuracies than using “non-spatial” boosting. This phenomenon was due to spatial correlation of our attributes, which means that data points that are close in the attribute space are probably close in real space, too. However, neural networks or decision trees do not consider spatial local information during the training, and unlike k-NN do not gain from sampling spatial data blocks.

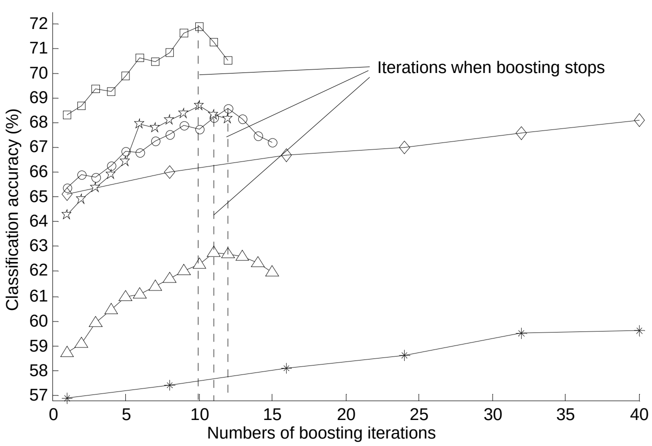
Table 1. Overall averaged classification accuracy (%) of spatial boosting for the 3 equal size classes on both synthetic test data sets using k-NN classifiers.

		Homogeneous data set					**Heterogeneous data set with 5 clusters**				
Number of Boosting Rounds		**8**	**16**	**24**	**32**	**40**	**8**	**16**	**24**	**32**	**40**
Fixed Attribute Set (M = 5)		79.1	79.6	80.1	80.7	80.6	65.6	65.5	65.8	66.0	66.1
Backward Elimination	M = 2	78.9	79.3	80.3	80.2	79.9	64.6	65.2	65.5	65.4	65.3
	M = 3	80.1	79.7	80.7	80.6	80.8	65.3	65.9	65.9	66.2	66.4
	M = 4	80.3	80.1	80.8	80.5	81.0	65.4	65.2	65.8	66.1	66.7
	M = 5	81.2	80.8	82.3	82.4	82.5	66.0	66.7	67.0	67.6	68.1
Attribute Weighting (M = 5)		79.4	78.8	80.1	80.7	80.3	64.2	64.7	65.4	66.3	65.9

When performing boosting specialized experts (Table 2, Figures 8 and 9) on heterogeneous data set with all attributes, instead of performing unsupervised feature selection around a clustering algorithm at each boosting iteration, we always applied clustering using the attributes f4 and f5, since we knew that these attributes determine homogeneous distributions. When one of two attributes responsible for clustering was missing, we performed clustering using available clustering attribute and the most relevant attribute obtained through the feature selection process. In addition, we always used all five relevant attributes for training specialized classifiers. The experiments performed on homogeneous data set showed similar performance like in heterogeneous data with missing clustering attribute and they are not reported here.

Table 2. Final averaged classification accuracies (%) for the 3 equal size classes. Different boosting algorithms are applied on both synthetic heterogeneous test data sets.

Heterogeneous data sets →			**Set with all relevant attributes**			**Set with missing clustering attribute**		
Method			k-NN	Neural Network	ID3	k-NN	Neural Network	ID3
Single Classifier			57.3	61.0 ± 2.2	63.3	57.3	61.0 ± 2.2	63.3
DBSCAN Clustering with single specialized classifiers			62.1	71.3 ± 0.9	67.7	58.2	63.1 ± 1.4	64.2
Standard Boosting			58.2 ± 0.7	69.8 ± 1.1	69.2 ± 0.6	58.2 ± 0.7	69.8 ± 1.1	69.2 ± 0.6
Adaptive Attribute Boosting			59.1 ± 0.6	69.4 ± 1.1	69.8 ± 0.6	59.1 ± 0.6	69.4 ± 1.1	69.8 ± 0.6
Spatial Boosting (M = 5)			68.1 ± 0.9	69.1 ± 1.2	68.2 ± 0.07	**68.1 ± 0.9**	69.1 ± 1.2	68.2 ± 0.07
Boosting Specialized Experts with Clustering	k-means clustering		**66.4** ± 1.1	**72.6** ± 1.1	**71.2** ± 0.8	61.8 ± 1.3	70.4 ± 1.5	69.9 ± 1.1
	DBSCAN clustering	*simple*	66.9 ± 1.4	73.9 ± 1.7	72.1 ± 1.0	62.1 ± 1.4	71.1 ± 1.8	70.4 ± 1.3
		previous	67.4 ± 1.3	74.4 ± 1.5	72.8 ± 1.2	63.3 ± 1.5	71.3 ± 1.9	70.5 ± 1.3
		best_global	67.9 ± 1.3	74.9 ± 1.4	73.4 ± 1.1	62.4 ± 1.4	71.6 ± 1.5	70.8 ± 1.1
		best_local	**68.6** ± 1.1	**76.6** ± 1.2	**74.5** ± 0.9	**62.7 ± 1.3**	**71.9 ± 1.4**	**71.1 ± 1.2**
Spatial Boosting Specialized Experts (DBSCAN + *best_local*)			**71.9 ± 1.0**	76.4 ± 1.3	74.4 ± 1.0	**68.6 ± 1.1**	71.4 ± 1.5	70.8 ± 1.3



Boosting applied to heterogeneous data set
* - Adaptive Attribute Boosting
◊ - Drawing spatial blocks
o - Boosting specialized experts with **DBSCAN** clustering (***best_local*** technique)
□ - Spatial Boosting specialized experts with **DBSCAN** clustering (***best_local***)

Boosting applied to heterogeneous data set with missing clustering attribute
Δ - Boosting specialized experts with **DBSCAN** clustering (***best_local*** technique)
★ - Spatial Boosting specialized experts with **DBSCAN** clustering (***best_local***)

Figure 8. Overall classification accuracies (%) of k-NN for the 3 equal-size class problems on heterogeneous synthetic test data sets with 5 relevant and 5 irrelevant attributes.

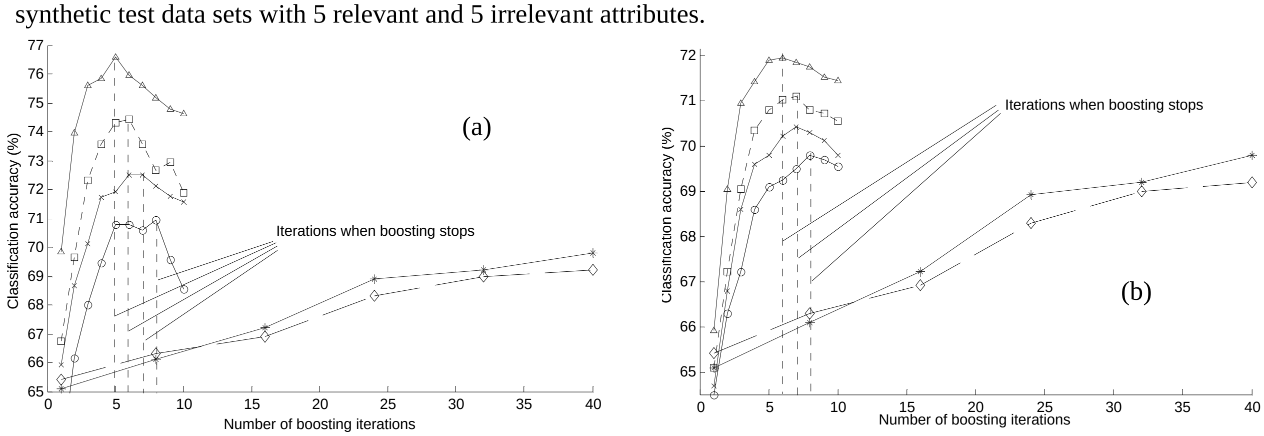


Figure 9. Overall classification accuracies (%) for the 3 equal size classes for global predictors applied on (a) heterogeneous synthetic test data set with all available attributes, (b) heterogeneous synthetic test data set with missing one clustering attribute. (*****- Adaptive Attribute Boosting with neural networks, Boosting specialized neural networks with **×** - k-means clustering, **Δ** - DBSCAN clustering (best_local), ◊ - Adaptive Attribute Boosting with ID3s, Boosting specialized ID3 classifiers with **o** - k-means clustering, □ -DBSCAN clustering (best_local))

All methods of boosting specialized experts resulted in improved generalizations for all synthetic spatial data sets. However, improvements for heterogeneous data set with all attributes (approximately 68 – 77%) were much more significant than for heterogeneous data set with missing clustering attribute (approximately 63 -72%) as compared to 57 – 63% obtained by single classifiers, specialized classifiers built on identified clusters, standard boosting and adaptive attribute boosting as shown at Table 2, Figures 8 and 9. Therefore, it is apparent that the prediction accuracy of all methods for boosting specialized experts directly depends on the quality of identified clusters during boosting iterations.

Boosting specialized experts is slightly more beneficial when boosting k-NN classifiers than global prediction models (Table 2), since the discovered clusters emphasize the local information, which is more helpful for local learning algorithms than for the global ones. Compared to the pure boosting specialized experts, the spatial boosting of global specialized classifiers again did not significantly affect the overall classification accuracy, while the influence of drawing spatial blocks when boosting specialized k-NN classifiers was reduced as compared to the improvements of pure spatial boosting over the standard and adaptive attribute boosting. This is due to the observed phenomenon that the smaller discovered clusters are not totally spatial, i.e. they contain scattered points in the spatial domain, and, in such cases, drawing spatial blocks does not help in reducing the total classification error.

It was also evident that the boosting of specialized experts required significantly fewer iterations in order to reach the maximal prediction accuracy. After 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 due to the fact that in the later iterations only data points that were difficult for learning were drawn. Therefore, there was not sufficient number of data examples in identified clusters needed for successful learning, and the prediction accuracy on these clusters begun to deteriorate thus causing the drop of the total prediction accuracy too.

The data distribution of clusters discovered by applying DBSCAN clustering algorithm to heterogeneous data set with all attributes was monitored at each boosting iteration (Figure 10). 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 - 8 for global classification models and 8 - 12 for k-NN classifiers) were sufficient. As observed in Figure 10, 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 a very small number of data points from the clusters where the prediction accuracy was good in previous iterations. As one of the criteria for stopping boosting early, we stop the boosting procedure when the size of any of the discovered clusters is less than some predefined number (usually less than 40). An additional stopping criterion is to observe the classification accuracy on the entire training set and to stop the procedure when it starts to decline. Figures 8 and 9 show the iterations when we stopped the boosting procedure. Although in practice the prediction accuracy on the test set does not necessarily start to drop in the same iteration, this difference is usually within two boosting iterations and does not significantly affect the total generalizability of the proposed method.

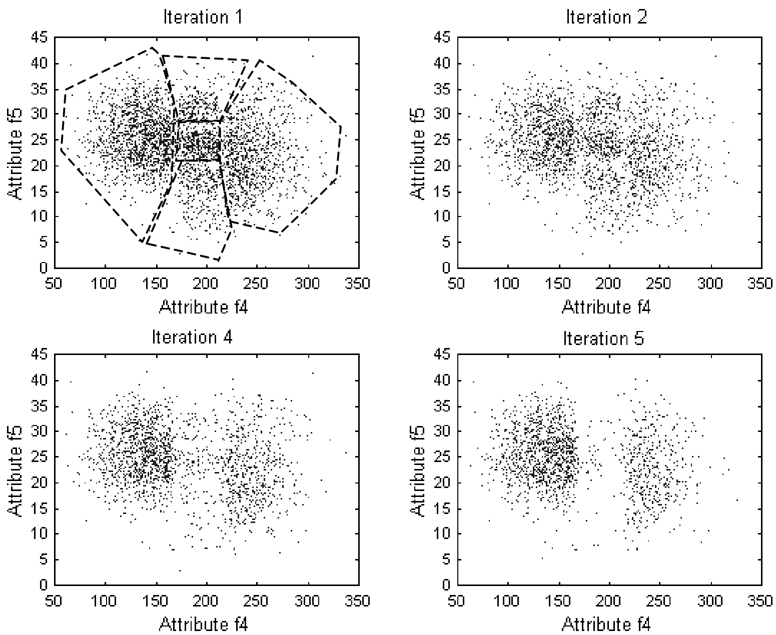


Figure 10. Changing the distributions of drawn samples during boosting on the neural network classifier. Samples from initial iterations contain points from all clusters, while samples from later iterations contain a small number of points from the central clusters where the accuracy was good.

When using the k-means clustering algorithm during the boosting procedure, we did not notice the phenomenon of reducing the size of discovered clusters and therefore we did not perform the modifications of the proposed algorithm. In addition, it was evident that boosting specialized experts when using the k -means clustering algorithm was not as successful as boosting localized experts with the DBSCAN algorithm, due to the better quality of the 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 2), while the ***simple*** and ***previous*** methods were not significantly better than boosting localized experts with k -means clustering. The ***simple*** technique failed to achieve improved prediction results, because it did not reach enough boosting iterations to develop the most appropriate classifiers for each cluster that needed to be combined, while the ***previous*** method had a boosting cycle that was long enough, but did not combine appropriate models. Finally, the ***best_global*** and ***best_local*** methods combined the most accurate models for each cluster taken in some of the earlier iterations, and hence achieved the best generalizability.

Experiments with all proposed boosting modifications were repeated for 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) 16 miscellaneous attribute selection methods (Table 3) were applied on the training data set in order to identify the four most relevant attributes that were used in the standard boosting method. Histograms for these most stable attributes (4, 7, 9, 20) are shown in Figure 11.

Table 3. Attribute selection methods used to identify the 4 most stable attributes on train 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 substitution		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 on real life data set, the four and five attributes were selected and monitored and their frequency was computed. The frequency of selected attributes during the boosting rounds, when boosting was applied to k-NN classifiers, neural network and decision tree classification models, is presented in Figures 12, 13 and 14 respectively. When PCA was used with boosting k-NN classifiers, projections to four dimensions explained most of the variance and there was little improvement from additional dimensions. For the attribute weighting method in boosting k-NN predictors, we used the attribute synaptic weights between input nodes and the output node of a 1-layer neural network constructed for each drawn sample. When boosting was applied to global classifiers (neural network classifiers and decision trees), only attribute selection procedures for changing attribute representation were considered. The achieved classification accuracies for both local and global classifiers are given in Table 4.

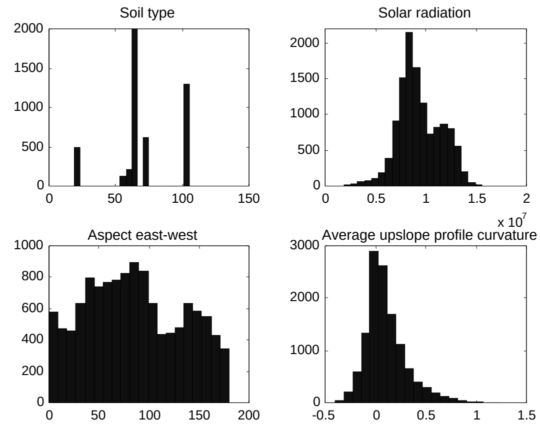


Figure 11. Histograms of 4 most relevant attributes of real life data set

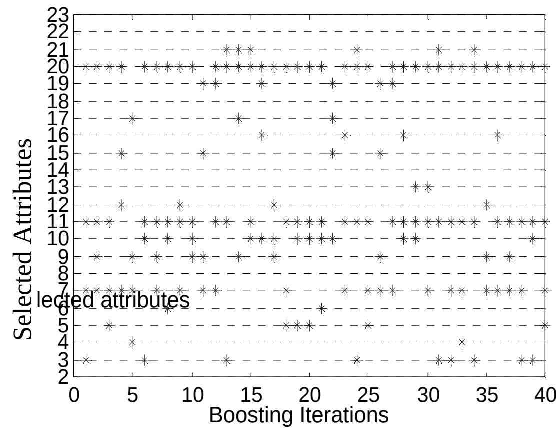


Figure 12. Attribute stability during boosting on k-NN classifiers (* denotes that attribute is selected in boosting round, - denotes that attribute is not selected)

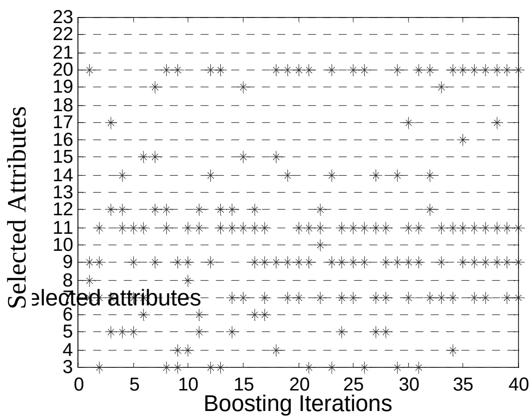


Figure 13. Attribute stability during boosting on neural network with Levenberg-Marquardt algorithm

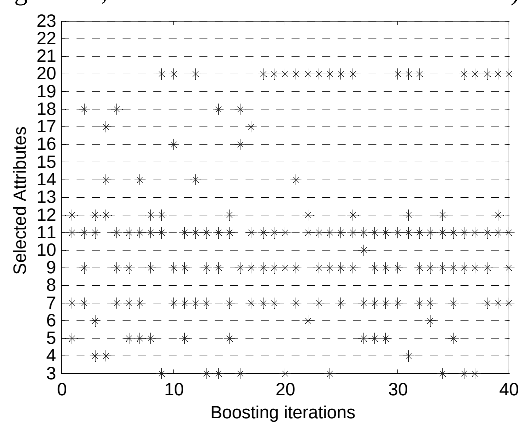


Figure 14. Attribute stability during boosting on ID3 decision tree algorithm

Table 4. Comparative analysis of overall classification accuracies (%) for the 3 equal-size class problems on real life test data with 19 soil and topographic attributes.

Number of Boosting Rounds	k-NN classifier: Standard Boosting	k-NN: Adaptive Attribute Boosting with Forward Selection	k-NN: Adaptive Attribute Boosting with Backward Elimination	k-NN: Adaptive Attribute Boosting with PCA	k-NN: Adaptive Attribute Boosting with Attribute Weighting	Levenberg-Marquardt neural networks: Standard Boosting	Levenberg-Marquardt neural networks: Backward Elimination	ID3 Decision Trees: Standard Boosting	ID3 Decision Trees: Backward Elimination
8	38.2	40.9	38.5	42.4	43.0	43.6	47.5	43.3	46.9
16	39.5	41.3	38.8	42.4	43.9	44.1	47.8	43.7	47.3
24	38.8	41.9	42.1	44.5	44.8	44.8	48.3	44.3	47.8
32	38.5	41.8	43.5	45.1	46.1	45.5	48.8	45.0	48.2
40	39.3	42.1	42.8	43.4	44.3	44.9	48.5	45.2	48.4

Results from Table 4 show that the methods of adaptive attribute boosting outperformed the standard boosting technique for both local and global classifiers. The results indicate that 30 boosting rounds were usually sufficient to maximize prediction accuracy and to somewhat stabilize the selected attributes although attribute selection during boosting was less stable for k-NN (Figure 12) than for neural networks (Figure 13) or decision trees (Figure 14). For k-NN after approximately 30 boosting rounds the attributes became fairly stable with attributes 7, 11 and 20 obviously more stable than attributes 3 and 9, which also appeared in later iterations. The prediction accuracies when using k-NN classifier with Mahalanobis distance were worse than those using Euclidean distance, and are not reported here.

When boosting neural network classifiers we used models defined in section 3.1.2, and the best results were obtained using the applied backward elimination attribute selection and the Levenberq-Marquardt learning algorithm (Table 4). On the other hand, decision trees used all selected attributes for computing the splitting criterion, and after constructing they are pruned such that the number of nodes in pruned trees was reduced for 20%.

Classification accuracies of spatial boosting for k -NN classifiers on the real life data set were again much better than without using spatial information and comparable to boosting neural networks and decisi on trees (Table 5). Here, the classification accuracy improvements from increasing the size (M) of the spatial blocks were less apparent than for synthetic spatial data probably due to the higher spatial correlation of the synthetic data sets.

Table 5. Overall classification accuracy (%) of spatial boosting for the 3 equal -size class problems for real life test data using k-NN classifiers.

Number of Boosting Rounds	Spatial Boosting for k-NN with					
	Fixed Attribute Set	Backward Elimination Attribute Selection				Attribute Weighting
	M = 5	M = 2	M = 3	M = 4	M = 5	M = 5
8	46.4	45.8	47.7	48.1	47.8	45.2
16	46.6	46.2	47.6	48.1	47.7	45.6
24	46.7	46.7	47.9	48.2	48.2	45.8
32	46.9	46.9	48.3	48.4	47.9	46.3
40	47.0	47.2	48.3	47.9	47.8	45.9

When boosting specialized classifiers, all experiments were performed with the unsupervised wrapper procedure for identifying the most germane attributes for clustering and also with the supervised feature selection procedure for finding the most important attributes for each of t he discovered clusters. In order to reduce the computational cost of the unsupervised wrapper approach, we did not identify more than three most appropriate attributes for clustering, since our previous experiments with clustering on the entire training se t indicate that the best quality of clusters was obtained when using only two or three attributes [21]. The same experiments pointed out that modeling with four attributes results in the best prediction capability and therefore we were selecting only four attributes for constructing classifiers on discovered clusters. Figure 15 shows the overall classification accuracy when boosting k -

NN classifiers, while the results in Figure 16 were obtained using the Levenberq-Marquardt algorithm for optimizing neural network parameters and using the pruned ID3 trees with a relatively small pruning factor.

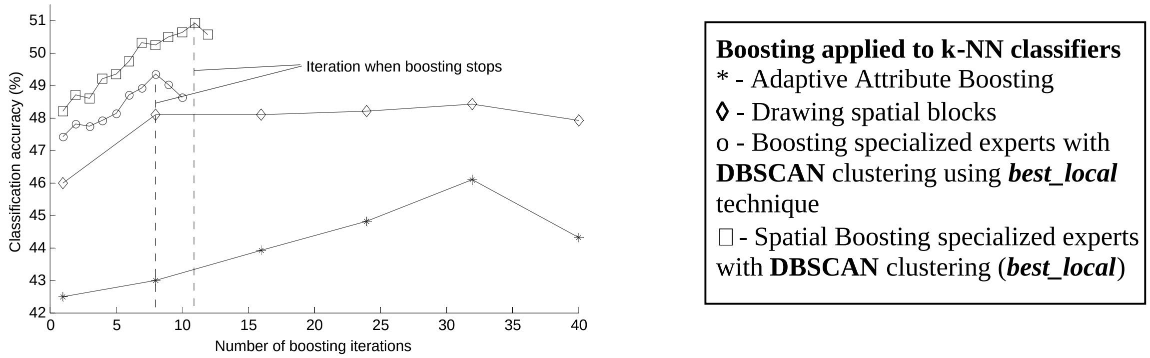


Figure 15. Overall classification accuracies of k-NN classifiers for the 3-class problems on real life test data.

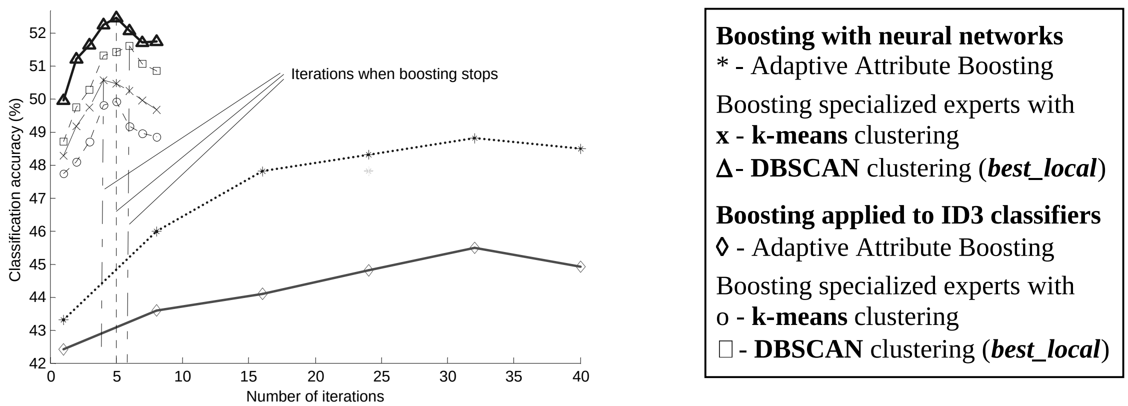


Figure 16. Overall classification accuracies of global predictors for the 3-class problems on real life test data.

Boosting specialized experts on a real life data set is not as superior to the adaptive attribute and spatial boosting methods as for the synthetic heterogeneous data set with all attributes. However, similar improvements in prediction accuracy were achieved for synthetic heterogeneous data set with missing clustering attribute. This indicates that in real life data, it is possible there is a lack of appropriate driving variables for explaining the variability of the target attribute. The discovered spatial clusters in real life data are not as distinct as the spatial clusters in synthetic heterogeneous data with all attributes, but the higher attribute instability was apparently beneficial for adaptive attribute boosting. Unlike synthetic heterogeneous data sets, for real life data the additional diversity of constructed classifiers is achieved by performing unsupervised attribute selection and by discovering clusters using the different attributes. Similar to experiments on synthetic data, the ***best_local*** technique of boosting localized experts was the most successful among all the proposed methods.

5. Conclusions and Future Work

Results from several spatial data sets indicate that the proposed techniques for combining multiple classifiers can result in significantly better predictions over existing classifier ensembles, especially for heterogeneous spatial 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. Second, our adaptive attribute boosting technique is more efficient than standard boosting, since a smaller number of iterations was sufficient to achieve the same final prediction accuracy. In addition, the attribute stability test served as a good indicator for properly stopping further boosting iterations. Third, the new boosting method proposed for spatial data showed promising results for k-NN classifiers making it competitive with powerful global classification models like neural networks and decision trees. Finally, boosting specialized experts with clustering performed at each boosting round further significantly improved both the prediction accuracy on highly heterogeneous databases and the efficiency of the algorithm by additional reducing the number of boosting iterations needed for achieving maximal prediction accuracy. However, for homogeneous data as well as for heterogeneous data sets with missing relevant attributes, the proposed method of boosting specialized classifiers showed only small improvements in achieved prediction accuracy.

Although boosting specialized experts required order of magnitude less boosting rounds to achieve the maximum prediction accuracy than the standard, adaptive attribute or spatial 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 and that only a few boosting iterations were sufficient to maximize 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 boosted specialized experts, 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 specialize experts was therefore 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 standard boosting or adaptive attribute boosting.

Despite the fact that the new fast k-NN classifier significantly reduces the computational requirements, an open research question is to further increase the speed of ensembles of k-NN classifiers for high-dimensional data. Although the performed experiments provide evidence that the proposed approach can improve predictions by ensembles of both local and global classifiers, further work is needed to examine the adaptation of global classifiers when boosting spatial data. In order to use the advantages from both local and non-linear prediction models, we are currently experimenting with a method of boosting radial basis functions. In addition, we are working to extend the method to regression-based problems.

Acknowledgments. The authors are grateful to Xiaowei Xu for providing the software for the DBSCAN clustering algorithm, to Dragoljub Pokrajac for providing simulated data and practical advice, to Tim Fiez for providing real life data, and to Celeste Brown for her useful comments.

6. References

1. Alpaydin, E., Voting over Multiple Condensed Nearest Neighbors, in *Lazy Learning*, (D. Aha, ed.), Kluwer, 115-132, 1997.
2. Avnimelech, R., Intrator, N., Boosting Mixture of Experts: An Ensemble Learning Scheme, *Neural Computation*, 11:475-490, 1999.
3. Bay, S., Nearest Neighbor Classification from Multiple Feature Subsets, *Intelligent Data Analysis*, 3(3):191-209, 1999.
4. Bottou, L., Vapnik, V., Local Learning Algorithms, *Neural Computation*, 4(6): 888-900, 1992.
5. Breiman, L.: Bagging predictors, *Machine Learning* 24, 123-140, 1996.
6. Cherkauer, K., Human Expert-level Performance on a Scientific Image Analysis Task by a System Using Combined Artificial Neural Networks, in *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, (P. Chan, ed.), 15-21, 1996.
7. Dasarathy, B. V., Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, *IEEE Computer Society Press*, 388-397, 1991.

8. Dy, J. and Brodley, C., Feature Subset Selection and Order Identification for Unsupervised Learning, in *Proceedings of the 17th International Conference on Machine Learning*, 247-254, 2000.

9. Freund, Y., and Schapire, R. E., Experi ments with a New Boosting Algorithm, in *Proceedings of the 13th International Conference on Machine Learning*, 325-332, 1996.

10. Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, 1990.

11. Hagan, M., Menhaj, M.B., Training f eedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks* 5, 989-993, 1994.

12. Ho, T. K., The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832-844, 1998.

13. Jordan M., Jacobs R., Hierarchical Mixture of Experts and the EM Algorithm. *Neural Computation*, 6(2): 181-214, 1994.

14. Kaufman, L., Rousseeuw, P., *Finding groups in data: an introduction to cluster analysis*, Willey, New York, 1990.

15. Kong, E. B., Diett erich, T. G., Error -Correcting Output Coding Corrects Bias and Variance, In *Proceedings of the 12th National Conference on Artificial Intelligence*, 725-730, 1996.

16. Kubat, M., Cooperson, M., Voting Nearest Neighbor Subclassifiers, in *Proceedings of the 17th International Conference on Machine Learning*, 503 – 510, 2000.

17. Kuncheva, L., Bezdek, J., Duin, R., Decision Templates for Multiple Classifier Fusion: An Experimental Comparison, *Pattern Recognition*, 34, 299-314, 2001.

18. Lazarevic A, Xu X, Fiez T, Obradovic Z ., Clustering-Regression-Ordering Steps for Knowledge Discovery in Spatial Databases, in *Proceedings of IEEE/INNS International Conference on Neural Networks*, No. 345, Session 8.1B, 1999.

19. Lazarevic, A., Fiez, T., Obradovic, Z., Adaptive Boosting for Spatia l Functions with Unstable Driving Attributes, in *Proceedings of Pacific -Asia Conference on Knowledge Discovery and Data Mining*, 329 –340, 2000.

20. Lazarevic, A., Pokrajac, D., and Obradovic, Z., Distributed Clustering and Local regression for Knowledge Discovery in Multiple Spatial Databases, in *Proceedings of 8th European Symposium on Artificial Neural Networks*, 129-134, 2000.

21. Lazarevic, A. and Obradovic, Z., Knowledge Discovery in Multiple Spatial Databases, in review.

22. Liu, L. and Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, Boston, 1998.

23. Margineantu, D., and Dietterich, T., Pruning adaptive boosting, in *Proceedings of the 14th International Conference on Machine Learning*, 211-218, 1997.

24. Moerland, P., Mayoraz, E., DynaBoost: Combining Boosted Hypotheses in a Dynamic Way, IDIAP Research Report 99-09, 1999.

25. Opitz, D., Feature Selection for Ensembles, in *Proceedings of 16th National Conference on Artificial Intelligence*, 379-384, 1999.

26. O'Sullivan, J., Langford, J., Caruna, R., Blum, A., FeatureBoost: A Meta-Learning Algorithm that Improves Model Robustness, in *Proceedings of the 17th International Conference on Machine Learning*, 703-710, 2000.

27. Pokrajac, D., Obradovic, Z., Combining Regressive and Auto-Regressive Models for Spatio-Temporal Prediction, in *Proceedings of 17th International Machine Learning Workshop on Spatial Knowledge*, 2000.

28. Pokrajac D, Fiez T, Obradovic Z., A Spatial Data Simulator for Agriculture Knowledge Discovery Applications, in review.

29. Quinlan, R., Induction of Decision Trees, *Machine Learning*, 1(1), 81 – 106, 1986.

30. Quinlan, R., Bagging, Boosting and C4.5, in *Proceedings of the 13th National Conference on Artificial Intelligence*, 725–730, 1996.

31. Ricci, F., and Aha, D. W., Error-Correcting Output Codes for Local Learners, in *Proceedings of the 10th European Conference on Machine Learning*, 280-291, 1998.

32. Riedmiller, M., Braun, H., A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, in *Proceedings of the IEEE International Conference on Neural Networks*, 586–591,1993.

33. Sander J., Ester M., Kriegel H-P, Xu X., Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications, *Data Mining and Knowledge Discovery*, 2(2): 169-194, 1998.

34. Schwenk, H., Bengio, Y., Boosting Neural Networks, *Neural Computation*, 12:1869-1887, 1999.

35. Tumer, K., and Ghosh, J., Error Correlation and Error Reduction in Ensemble Classifiers, *Connection Science* 8, 385-404, 1996.