Multiple Classifiers Applied to Multisource Remote Sensing Data

Gunnar Jakob Briem, *Associate Member, IEEE*, Jon Atli Benediktsson, *Senior Member, IEEE*, and Johannes R. Sveinsson, *Senior Member, IEEE*

***Abstract*—The combination of multisource remote sensing and geographic data is believed to offer improved accuracies in land cover classification. For such classification, the conventional parametric statistical classifiers, which have been applied successfully in remote sensing for the last two decades, are not appropriate, since a convenient multivariate statistical model does not exist for the data. In this paper, several single and multiple classifiers, that are appropriate for the classification of multisource remote sensing and geographic data are considered. The focus is on multiple classifiers: bagging algorithms, boosting algorithms, and consensus-theoretic classifiers. These multiple classifiers have different characteristics. The performance of the algorithms in terms of accuracies is compared for two multisource remote sensing and geographic datasets. In the experiments, the multiple classifiers outperform the single classifiers in terms of overall accuracies.**

***Index Terms*—Bagging, boosting, consensus theory, multiple classifiers, multisource remote sensing data.**

I. Introduction

DATA FUSION of multisource remote sensing and geographic data for classification purposes has been an important research topic for more than a decade. In such fusion, different types of information from several data sources, e.g., Landsat Thematic Mapper data, radar data, elevation data, and slope data, are used in order to improve the classification accuracy as compared to the accuracy achieved by single-source classification.

A major observation in previous research on multisource classification is that conventional parametric statistical pattern recognition methods are not appropriate in classification of such data, since in most cases they cannot be modeled by a convenient multivariate statistical model [1], [2]. Therefore, other methods have been looked at.

Here, we are interested in the use of an ensemble of classifiers or *multiple classifiers* (a schematic representation of which is shown in Fig. 1) for classification of multisource data. Traditionally, in pattern recognition, a single classifier is used to determine which class a given pattern belongs to. However, in many cases, the classification accuracy can be improved by using an ensemble of classifiers in the classification. In such cases, it is possible to have the individual classifiers support each other in making a decision. The aim is to determine an effective combination method that makes use of the benefits of each classifier but avoids the weaknesses.

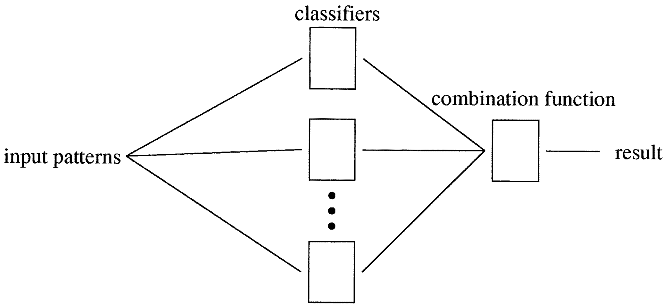

Fig. 1. Schematic diagram of a multiple classifier.

In this paper, the performance of three types of multiple classifiers are investigated in terms of classification of multisource remote sensing and geographic data. The paper is organized as follows. First, multiclassifier systems are discussed with a special emphasis on the recently proposed bagging and boosting algorithms and statistical consensus theory. Experimental results for the multisource remote sensing and geographic datasets are given in Section III. Finally, conclusions are drawn in Section IV.

II. Multiclassifier Systems

Several methods have been proposed to combine multiple classifiers [3]–[5]. Wolpert [3] introduced the general method of stacked generalization where outputs from classifiers are combined in a weighted sum with weights that are based on the individual performance of the classifiers. Tumer and Ghosh [4] have also shown that substantial improvements can be achieved in difficult pattern recognition problems by combining or integrating the outputs of multiple classifiers. Benediktsson *et al.* combined classifiers using neural networks [6], [7] and improved their overall accuracies as compared to the best results of the single classifiers involved in the classification.

The theory of multiclassifier systems can be traced back at least as far as 1965 [8], [9]. Currently, two of the most used multiclassification approaches are boosting [10], [11] and bagging [12]. Both these approaches are based on manipulating training samples. In contrast, statistical consensus theory is based on treating data sources separately, and it uses all the training data only once. All three approaches are discussed briefly below.

Manuscript received September 26, 2001; revised May 17, 2002. This research was supported in part by the Icelandic Research Council and the Research Fund of the University of Iceland.

G. J. Briem was with the Department of Electrical and Computer Engineering, University of Iceland, Reykjavik IS-107, Iceland. He is now with Dimon Software, Reykjavik IS-105, Iceland.

J. A. Benediktsson and J. R. Sveinsson are with the Department of Electrical and Computer Engineering, University of Iceland, Reykjavik IS-107, Iceland.

Digital Object Identifier 10.1109/TGRS.2002.802476

0196-2892/02$17.00 © 2002 IEEE

A. Boosting

Boosting is a general supervised method that is used to increase the accuracy of any classifier. Several versions of boosting have been proposed, but we will concentrate on AdaBoost [11], which was proposed in 1995. In particular, we will use the AdaBoost.M1 method, which can be used on classification problem with more than two classes. This version of the AdaBoost algorithm is shown as follows.

```
Input: A training set $S$ with $m$ samples, where each
sample $x_j$ is of class $\omega_j$, base classifier $\mathcal{I}$, and
number of classifiers $T$.
1. $S_1 = S$ and weight$(x_j) = 1$ for $j = 1 \ldots m$ $(x \in S_1)$
2. For $i = 1$ to $T$ {
3.     $C_i = \mathcal{I}(S_i)$
4.     $\epsilon_i = 1/m \sum_{x_j \in S_i : C_i(x_j) \neq \omega_j}$ weight$(x_j)$
5.     If $\epsilon_i > 0.5$, set $S_i$ to a bootstrap sample from
       $S$ with weight$(x) = 1 \, \forall \, x \in S_i$ and go to Step 3
       If $\epsilon_i$ is still $> 0.5$ after 25 iterations, abort!
6.     $\beta_i = \epsilon_i/(1 - \epsilon_i)$
7.     For each $x_j \in S_i${ if $C_i(x_j) = \omega_j$ then
         weight$(x_j)$ = weight$(x_j) \cdot \beta_i$}
8.     Norm weights such that the total weight
       of $S_i$ is $m$
9. }
10. $C^*(x) = \arg\max_{\omega \in \Omega} \sum_{i : C_i(x) = \omega} \log(1/\beta_i)$
Output: The multiple classifier $C^*$.
```

In the beginning of AdaBoost, all samples have the same weight, $1/m$, thus forming a uniform distribution D_1, and they are used to train the classifier C_1. Then, the samples are reweighted in such a way that the incorrectly classified samples have more weight than the correctly classified ones. Based on this new distribution, the classifier of the next iteration C_2 is trained. The classifier of iteration i, C_i, is therefore based on the distribution D_{i-1} calculated in iteration $i - 1$.

Iteration by iteration, the weight of the samples that are correctly classified goes down. Therefore, the algorithm starts concentrating on the difficult samples. At the end of the procedure, T weighted training sets and T base classifiers have been generated.

It is recognized that not all base classifiers accept a weighted set of samples. In such cases, instead of supplying the distribution D_i directly to the base classifier, a set of samples is chosen from S in accordance with D_i by replacement. This is similar to the bootstrapping performed in the bagging method discussed next, except that the distribution generally is not uniform.

If the classification error is greater than 0.5, an attempt is made to decrease it by bootstrapping. If the classification error stays above 0.5, then the procedure is stopped (in failure). A demand is made, therefore, on the minimum accuracy of the base classifier, which can be of considerable disadvantage in multiclass problems.

The main advantage of AdaBoost is that in many cases it increases the overall accuracy of the classification. Many practical classification problems include samples that are not equally difficult to classify, and AdaBoost is suitable for such problems. AdaBoost tends to exhibit virtually no overfitting when the data are noiseless. Other advantages of boosting include that the algorithm has a tendency to reduce both the variance and the bias of the classification [11], [13]. On the other hand, AdaBoost is computationally more demanding than other simpler methods. Therefore, it is dependent on the classification problem whether it is more valuable to get increased classification accuracy or to obtain a simple and fast classifier. Another problem with AdaBoost is that it usually does not perform well in terms of accuracies when there is noise in the data.

B. Bagging

Bagging is an abbreviation of ***b***ootstrap ***agg***regat***ing***. Bootstrap methods are based on randomly and uniformly collecting m samples with replacement from a sample set of size m. The bagging algorithm (which, like AdaBoost, is supervised) was proposed in 1994 [12] and constructs many different bags of samples by performing bootstrapping iteratively, classifying each bag, and computing some type of an average of the classifications of each sample via a vote. Bagging is in some ways similar to boosting, since both methods design a collection of classifiers and combine their conclusions with a vote. However, the methods are different. For example, because bagging always uses resampling instead of reweighting, it does not change the distribution of the samples (does not weight them), so all classifiers in the bagging algorithm have equal weights during the voting. It is also noteworthy that bagging can be done in parallel, i.e., it is possible to prepare all the bags at once. On the other hand, boosting is always done in series, and each sample set is based on the latest weights. The bagging algorithm can be written as follows.

```
Input: A training set $S$ with $m$ samples, where each
sample $x_j$ is of class $\omega_j$, base classifier $\mathcal{I}$, and
number of bootstrapped sets $T$.
1. For $i = 1$ to $T${
2.     $S_i =$ bootstrapped bag from $S$
3.     $C_i = \mathcal{I}(S_i)$
4. }
5. $C^*(x) = \arg\max_{\omega \in \Omega} \sum_{i : C_i(x) = \omega} 1$
Output: The multiple classifier $C^*$.
```

From the above, it can be seen that bagging is a very simple algorithm. Each classifier C_i is trained on a bootstrapped set of samples S_i from the original sample set S. After all classifiers have been trained, a simple majority vote is used, but if more than one class jointly receives the maximum number of votes, then the winner is selected using some simple mechanism, e.g., random selection. For a particular bag S_i, the probability that a sample from S is selected at least once in m tries is $1 - (1 - 1/m)^m$. For a large m, the probability is approximately $1 - 1/e \approx 0.632$, indicating that each bag only includes about 63.2% of the samples in S. If the base classifier is unstable, i.e., when a small change in training samples can result in a large change in classification accuracy, then bagging can improve the classification accuracy significantly. If the base classifier is stable, e.g., like a k-NN classifier, then bagging can reduce the classification accuracy because each classifier receives less of the training data.

The main advantage of the bagging algorithm is that it can increase the classification accuracy significantly if the base classifier is properly selected. The bagging algorithm is also not very sensitive to noise in the data. The algorithm uses the instability of its base classifier in order to improve the classification accuracy. Therefore, it is of great importance to select the base classifier carefully. This is also the case for boosting, since it is sensitive to small changes in the input signal. Bagging reduces the variance of the classification, just as boosting does, but in contrast to boosting, bagging has little effect on the bias of the classification.

C. Consensus Theory

Consensus theory is not based on manipulating the training data like bagging and boosting. Consensus theory [6], [14] involves general procedures with the goal of combining single probability distributions to summarize estimates from multiple experts, with the assumption that the experts make decisions based on Bayesian decision theory. The combination formula obtained is called a consensus rule. The consensus rules are used in classification by applying a maximum rule, i.e., the summarized estimate is obtained for all the information classes, and the pattern X is assigned to the class with the highest summarized estimate. Probably, the most commonly used consensus rule is the linear opinion pool (LOP), which is based on a weighted linear combination of the posterior probabilities from each data source. Another consensus rule, the logarithmic opinion pool (LOGP), is based on the weighted product of the posterior probabilities. The LOGP differs from the LOP in that it is unimodal and less dispersed. Also, the LOGP treats the data sources independently.

The weighting schemes in consensus theory should reflect the goodness of the input data. The simplest approach is to give all the data sources equal weights. Also, reliability measures that rank the data sources according to their goodness can be used as a basis for *heuristic weighting* [6]. Furthermore, the weights can be chosen to not only weight the individual sources but also the individual classes. For such a scheme, both linear and nonlinear optimization can be used.

III. Experimental Results

Two experiments were conducted on multisource remote sensing and geographic data using bagging, boosting, and consensus-theoretic classifiers. To obtain baseline results for the multiple classifiers, several single classifiers were applied to the data. These include the minimum Euclidean distance (MED) classifer, Gaussian maximum likelihood (ML) classifier, and conjugate-gradient backpropagation (CGBP) [6] with two and three layers. The base classifiers that were used for bagging and boosting were also trained as single classifiers on the data. These base classifiers were as follows:

1) decision table with a ten-fold cross validation for feature selection and termination of search for the best feature set after 15 nonimproving subsets [15];
2) j4.8 decision tree [16] (an implementation of the C4.5 revision eight-decision tree [17]);
3) simple classifier 1R with minimum bucket size of eight [18], which only uses one feature when it determines a class.

TABLE I
Training and Test Samples for Information Classes in the Experiment on the Colorado Dataset

Class #	Information Class	Training Size	Test Size
1	Water	301	302
2	Colorado Blue Spruce	56	56
3	Mountane/Subalpine Meadow	43	44
4	Aspen	70	70
5	Ponderosa Pine 1	157	157
6	Ponderosa Pine/Douglas Fir	122	122
7	Engelmann Spruce	147	147
8	Douglas Fir/White Fir	38	38
9	Douglas Fir/Ponderosa Pine/Aspen	25	25
10	Douglas Fir/White Fir/Aspen	49	50
Total		1008	1011

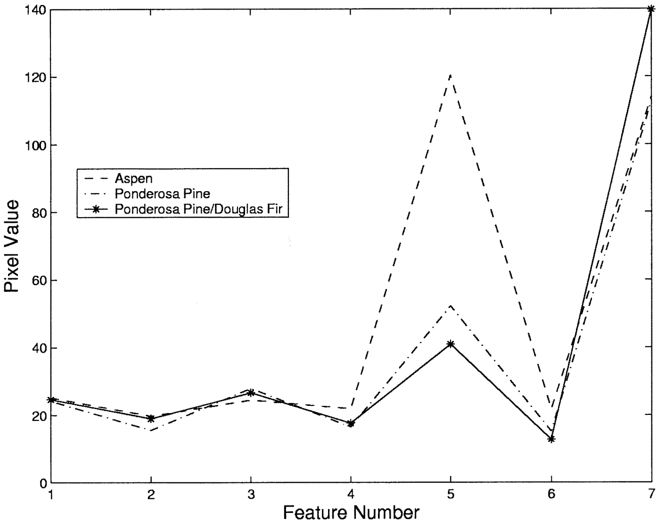

Fig. 2. Colorado data. Average spectra for three forest type classes. The first four features are the Landsat MSS data, followed by elevation, slope, and aspect respectively.

In summary, the single classifiers applied to the datasets were the following:

- minimum Euclidean distance (MED);
- maximum likelihood (ML);
- conjugate-gradient backpropagation (CGBP);
- decision table;
- j4.8 (an implementation of the C4.5 decision tree);
- 1R (classification based on one feature).

In the results below, the *average accuracy* is defined as the average of the classification accuracy of each class, regardless of the number of samples in each class. The *overall accuracy* is defined as the number of correctly classified samples, regardless of which class they belong to, divided by the total number of samples.

The results of the experiments are discussed below.

TABLE II
TRAINING ACCURACIES IN PERCENTAGE FOR THE DIFFERENT CLASSIFICATION METHODS APPLIED TO THE COLORADO DATASET

Method	Cl.	Cl.	Cl.	Cl.	Cl.	Cl.	Cl.	Cl.	Cl.	Cl.	**Avg.**	**Overall**
	1	2	3	4	5	6	7	8	9	10	**Acc.**	**Acc.**
MED	41.5	98.2	25.6	37.1	37.6	0.0	73.5	0.0	40.0	24.5	**37.8**	**40.3**
Decision Table	100.0	89.3	67.4	84.3	54.1	80.3	100.0	36.8	24.0	93.9	**73.0**	**82.8**
j4.8	100.0	83.9	79.1	87.1	68.8	88.5	99.3	57.9	52.0	95.9	**81.3**	**88.0**
1R	100.0	0.0	0.0	0.0	38.2	63.1	97.3	23.7	0.0	36.7	**35.9**	**60.3**
CGBP (40 hidden neurons)	100.0	96.7	95.7	99.5	90.3	89.5	100.0	87.3	96.7	100.0	**95.6**	**96.3**
LOP (equal weights)	100.0	0.0	0.0	92.9	38.9	49.2	100.0	0.0	12.0	100.0	**49.3**	**68.1**
LOP (heuristic weights)	100.0	25.0	16.3	91.4	36.3	90.2	99.3	0.0	0.0	100.0	**55.8**	**74.2**
LOP (optimal linear weights)	100.0	62.5	25.6	74.3	66.2	79.5	98.6	23.7	40.0	91.8	**66.2**	**80.3**
LOP (optimized with CGBP)	100.0	87.9	26.2	81.8	67.8	73.0	100.0	39.5	75.0	94.4	**74.6**	**83.5**
LOGP (equal weights)	99.7	96.4	20.9	87.1	60.5	46.7	100.0	44.7	44.0	91.8	**69.2**	**79.0**
LOGP (heuristic weights)	99.7	91.1	23.3	95.7	45.2	83.6	100.0	5.3	48.0	100.0	**69.2**	**80.5**
LOGP (optimal linear weights)	100.0	67.9	23.3	81.4	58.6	82.8	98.6	18.4	28.0	91.8	**65.1**	**79.7**
LOGP (optimized with CGBP)	100.0	80.4	69.8	99.6	78.3	82.8	100.0	80.3	100.0	100.0	**89.1**	**91.4**
Bagging (Decision Table)	100.0	69.6	76.7	95.7	81.5	82.0	100.0	31.6	60.0	98.0	**79.5**	**88.3**
Bagging (j4.8)	100.0	89.3	74.4	92.9	84.1	81.1	100.0	55.3	72.0	93.9	**84.3**	**90.4**
Bagging (1R)	100.0	41.1	60.5	61.4	50.3	68.0	97.3	21.1	32.0	83.7	**61.5**	**74.9**
Boosting (Decision Table)	100.0	82.1	88.4	98.6	75.2	83.6	100.0	68.4	100.0	100.0	**89.6**	**91.4**
Boosting (j4.8)	100.0	98.2	97.7	100.0	91.1	94.3	100.0	94.7	100.0	100.0	**97.6**	**97.5**
Boosting (1R)	100.0	94.6	79.1	95.7	78.3	76.2	100.0	63.2	100.0	100.0	**88.7**	**90.9**
Number of Samples	301	56	43	70	157	122	147	38	25	49		1008

A. Colorado Dataset

In the first experiment, classification was performed on a dataset consisting of the following four data sources:

1) Landsat MSS data (four spectral data channels);
2) elevation data (in 10-m contour intervals, one data channel);
3) slope data (0° to 90° in 1° increments, one data channel);
4) Aspect data (1° to 180° in 1° increments, one data channel).

The area used for classification is a mountainous area in Colorado. Ground reference data are available with ten ground-cover classes. One class is water; the others are forest types, as specified in Table I. It is very difficult to distinguish among the forest types using the Landsat MSS data alone, since the forest classes show very similar spectral response. A typical example of this is shown in Fig. 2 for the average spectra of three of the forest type classes. For these classes, it is clear that it helps to add the topographic data sources to the Landsat data in order to make the data more distinguishable.

The training accuracies for both single and multiple classifiers are given in Table II. The test accuracies are given in Table III. It was not possible to use the Gaussian maximum likelihood classifier for the Colorado dataset. The covariance matrix, which was computed using the standard unbiased estimation on the raw data, became singular for several of the forest type classes because of little variation in the topographic data. For the LOP and LOGP, ten data classes were defined in each data source. The multispectral remote sensing data sources were modeled to be Gaussian. On the other hand, the topographic data sources were modeled by Parzen density estimation with Gaussian kernels [19], which constructs an average of density functions, centered at each training sample. Several different weighting schemes were used for the LOP and LOGP [6].

Both bagging and boosting were applied on the raw data and run using the WEKA software provided by the University of Waikato, New Zealand [20]. In the case of bagging, 100 iterations were selected for the decision table, ten iterations for j4.8, and 200 iterations for 1R. For boosting, the Adaboost.M1 was employed, with 50 iterations for the decision table, 200 iterations for j4.8, and 60 iterations for 1R. The number of iterations was chosen after trying up to at least 300 iterations for j4.8 and 1R and 100 iterations for the decision table. A plot of the overall test accuracies as a function of the number of iterations is shown in Fig. 3. In each case, the ten-class problem was converted into multiple two-class problems, through the use of the *MultiClass-Classifier* [20], to make it more tractable for the base classifiers, especially in view of the stringent classification accuracy demand of the AdaBoost.M1 algorithm. As can be seen from Fig. 3, all plots have a knee at 10 to 20 iterations. After that, they improve only slightly.

At first glance, it may seem odd that in the experiments different numbers of iterations were used when bagging and boosting each base classifier (as discussed below). However, the

TABLE III
TEST ACCURACIES IN PERCENTAGE FOR THE DIFFERENT CLASSIFICATION METHODS APPLIED TO THE COLORADO DATASET

Method	Cl. 1	Cl. 2	Cl. 3	Cl. 4	Cl. 5	Cl. 6	Cl. 7	Cl. 8	Cl. 9	Cl. 10	**Avg. Acc.**	**Overall Acc.**
MED	40.1	100.0	34.1	30.0	32.5	0.8	69.4	0.0	28.0	20.0	**35.5**	**38.0**
Decision Table	100.0	80.4	50.0	74.3	47.1	73.0	96.6	28.9	4.0	80.0	**63.4**	**77.0**
j4.8	100.0	62.5	54.5	74.3	57.3	66.4	98.0	28.9	8.0	84.0	**63.4**	**77.4**
1R	100.0	0.0	0.0	0.0	30.6	65.6	95.9	15.8	0.0	24.0	**33.2**	**58.3**
CGBP (40 hidden neurons)	99.9	57.1	61.0	67.6	59.3	69.1	97.4	34.6	45.3	78.7	**67.0**	**78.4**
LOP (equal weights)	100.0	0.0	0.0	87.1	35.0	48.4	100.0	0.0	0.0	94.0	**46.5**	**66.4**
LOP (heuristic weights)	100.0	30.4	18.2	80.0	35.7	88.5	100.0	0.0	0.0	96.0	**54.9**	**73.4**
LOP (optimal linear weights)	100.0	80.4	25.0	77.1	66.3	75.4	99.3	15.8	32.0	92.0	**66.1**	**80.2**
LOP (optimized with CGBP)	100.0	90.2	39.2	75.3	61.0	74.6	99.3	34.9	58.0	96.5	**72.9**	**82.2**
LOGP (equal weights)	99.3	100.0	18.2	85.7	56.7	52.5	99.3	42.1	44.0	92.0	**69.0**	**78.7**
LOGP (heuristic weights)	100.0	96.4	18.2	91.4	40.8	87.7	99.3	10.5	24.0	100.0	**66.8**	**79.6**
LOGP (optimal linear weights)	100.0	76.8	25.0	75.7	63.7	81.1	99.3	13.2	16.0	92.0	**64.3**	**80.0**
LOGP (optimized with CGBP)	99.8	64.3	58.0	73.9	61.5	71.7	98.6	49.3	80.0	94.0	**75.1**	**82.3**
Bagging (Decision Table)	100.0	66.1	72.7	80.0	73.2	72.1	99.3	15.8	20.0	94.0	**69.3**	**82.5**
Bagging (j4.8)	100.0	60.7	63.6	75.7	69.4	75.4	99.3	28.9	40.0	82.0	**69.5**	**81.7**
Bagging (1R)	100.0	42.9	54.5	61.4	44.6	70.5	98.0	13.2	24.0	80.0	**58.9**	**73.6**
Boosting (Decision Table)	100.0	67.9	70.5	80.0	67.5	73.0	99.3	28.9	76.0	98.0	**76.1**	**83.8**
Boosting (j4.8)	100.0	60.7	70.5	77.1	65.6	67.2	98.6	42.1	60.0	84.0	**72.6**	**81.5**
Boosting (1R)	100.0	73.2	70.5	77.1	68.2	73.0	100.0	60.5	72.0	100.0	**79.4**	**85.3**
Number of Samples	302	56	44	70	157	122	147	38	25	50		1011

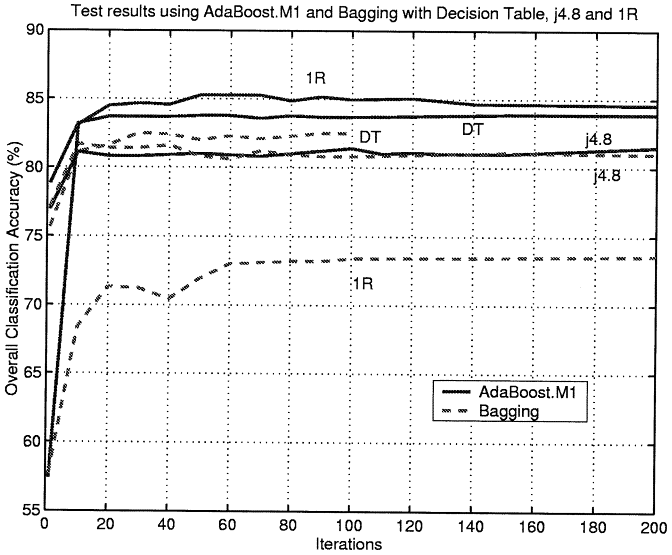

Fig. 3. Colorado data. Test accuracies as a function of the number of iterations.

goal was not to compare the computational demands of bagging and boosting each base classifier, but rather to compare them in terms of the highest achievable test accuracies. Therefore, it was decided to use the "optimal" number of iterations for each base classifier instead of using the same number of iterations for all three base classifiers.

TABLE IV
TRAINING AND TEST SAMPLES FOR INFORMATION CLASSES IN THE EXPERIMENT ON THE ANDERSON RIVER DATASET

Class #	Information Class	Training Size	Test Size
1	Douglas Fir (31-40m)	971	1250
2	Douglas Fir (21-30m)	551	817
3	Douglas Fir + Other Species(31-40m)	548	701
4	Douglas Fir + Lodgepole Pine (21-30m)	542	705
5	Hemlock + Cedar (31-40m)	317	405
6	Forest Clearings	1260	1625
	Total	4189	5503

From Tables II and III, it is apparent, that all three multiple classifier schemes show improvement over the single classifiers (minimum Euclidean distance, decision table, j4.8, 1R, and two-layer conjugate-gradient backpropagation with 40 hidden neurons) in terms of both average and overall accuracies (the ML classifier was not applicable here because of singularity problems). The highest overall and average training accuracies were achieved by boosting the j4.8 decision tree. However, the highest overall and average test accuracies were obtained by boosting the 1R base classifier, which gave far worse training and test accuracies on its own than the other base

TABLE V
TRAINING ACCURACIES IN PERCENTAGE FOR THE DIFFERENT CLASSIFICATION METHODS APPLIED TO THE ANDERSON RIVER DATASET

Method	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	**Average Accuracy**	**Overall Accuracy**
MED	40.4	8.9	47.6	67.7	42.3	72.4	**46.6**	**50.5**
ML	54.6	31.6	87.8	90.9	81.4	73.3	**69.9**	**68.2**
Decision Table	78.7	59.3	76.8	70.8	75.7	83.4	**74.1**	**76.1**
j4.8	93.9	91.5	93.8	93.9	96.2	97.0	**94.4**	**94.7**
1R	61.3	6.5	22.8	74.2	19.2	77.5	**43.6**	**52.4**
CGBP (30 hidden neurons)	72.2	34.4	67.2	74.6	79.2	83.1	**68.4**	**70.7**
LOP (equal weights)	49.6	0.0	0.0	51.5	0.0	94.9	**32.7**	**47.6**
LOP (heuristic weights)	68.2	0.0	0.0	73.1	24.3	89.4	**42.5**	**54.0**
LOP (optimal linear weights)	69.8	42.7	81.20	77.5	70.4	78.9	**70.1**	**71.5**
LOP (optimized with CGBP)	69.0	45.0	81.3	76.9	85.0	78.4	**72.6**	**71.8**
LOGP (equal weights)	68.7	28.1	79.6	78.8	81.7	74.3	**68.5**	**68.8**
LOGP (heuristic weights)	68.9	33.2	78.5	79.5	75.7	75.8	**68.6**	**69.4**
LOGP (optimal linear weights)	71.9	40.3	79.7	75.1	82.0	79.1	**71.4**	**72.1**
LOGP (optimized with CGBP)	81.2	56.0	84.3	88.7	91.7	86.4	**81.4**	**81.6**
Bagging (Decision Table)	97.6	91.5	97.6	96.9	100.0	99.0	**97.1**	**97.3**
Bagging (j4.8)	98.7	96.2	97.4	98.3	99.4	99.3	**98.2**	**98.4**
Bagging (1R)	75.1	14.9	46.7	48.7	71.6	80.6	**56.3**	**61.4**
Boosting (Decision Table)	99.5	97.3	99.1	99.3	99.4	99.7	**99.0**	**99.2**
Boosting (j4.8)	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**	**100.0**
Boosting (1R)	84.7	71.9	85.0	90.2	96.8	93.3	**87.0**	**87.3**
Number of Samples	971	551	548	542	317	1260		4189

classifiers. In contrast, bagging the 1R gave poor accuracies. The best overall and average accuracies for consensus-theoretic classifiers were achieved with the LOGP optimized by conjugate-gradient backpropagation. Those results were comparable in terms of overall accuracies to the best results achieved using bagging.

The classification accuracies for the individual classes are interesting in Tables II and III. When 1R is used, it gives extremely low accuracies for classes number 2, 3, 4, 8, and 9. Boosting gives significant improvements to the training and test classification accuracies for all these classes. For example, 1R gives 0% training and test accuracies for class number 9, but boosting the 1R base classifier gives a training accuracy of 100% and test accuracy of 72% for that class. Boosting using other base classifiers gives accuracy improvements for all other classes. Bagging also gives improvements in terms of class-specific accuracies but is outperformed by boosting.

B. Anderson River Dataset

In the second experiment, the Anderson River dataset, which is a multisource remote sensing and geographic dataset made available by the Canada Centre for Remote Sensing (CCRS) [21], was used. This dataset is very difficult to classify [6] due to a number of mixed forest type classes.

Six data sources were used:

1) Airborne Multispectral Scanner (AMSS) with 11 spectral data channels (ten channels from 380–1100 nm and one channel from 8–14 μm);
2) steep mode synthetic aperture radar (SAR) with four data channels (X-HH, X-HV, L-HH, L-HV);
3) shallow mode SAR with four data channels (X-HH, X-HV, L-HH, L-HV);
4) elevation data (one data channel, where elevation in meters $= 61.996 + 7.2266 *$ pixel value);
5) slope data (one data channel, where slope in degrees $=$ pixel value);
6) aspect data (one data channel, where aspect in degrees $=$ $2 *$ pixel value).

There are 19 information classes in the ground reference map provided by CCRS. In the experiments, only the six largest ones were used, as listed in Table IV. Here, training samples were selected uniformly, giving 10% of the total sample size. All other known samples were then used as test samples.

The results of the different classification methods are shown in Table V (training) and Table VI (test). For the single classifier methods in Tables V and VI, the MED and ML showed different characteristics. The MED was not acceptable in terms of classification accuracies, but the ML accuracies were rela-

TABLE VI
TEST ACCURACIES IN PERCENTAGE FOR THE DIFFERENT CLASSIFICATION METHODS APPLIED TO THE ANDERSON RIVER DATASET

Method	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	**Average Accuracy**	**Overall Accuracy**
MED	39.7	8.9	48.4	70.2	46.0	71.7	**47.5**	**50.8**
ML	50.8	27.7	84.5	81.9	73.8	72.0	**64.3**	**65.1**
Decision Table	73.8	42.4	66.5	61.7	72.8	77.0	**65.7**	**67.5**
j4.8	71.2	47.4	69.2	72.3	74.8	81.2	**69.4**	**70.8**
1R	58.5	4.3	19.3	74.5	20.0	77.4	**42.3**	**50.2**
CGBP (30 hidden neurons)	71.9	29.3	67.5	73.8	79.3	82.4	**67.4**	**68.8**
LOP (equal weights)	49.8	0.0	0.0	50.4	0.0	95.3	**32.6**	**45.8**
LOP (heuristic weights)	68.9	0.0	0.0	73.1	20.8	89.3	**42.0**	**53.9**
LOP (optimal linear weights)	66.4	34.3	78.5	74.8	72.6	79.5	**67.7**	**68.6**
LOP (optimized with CGBP)	67.1	36.7	77.3	75.1	83.4	77.6	**69.5**	**69.2**
LOGP (equal weights)	67.9	23.1	77.8	77.5	81.2	73.7	**66.9**	**66.4**
LOGP (heuristic weights)	69.0	31.8	75.9	78.6	75.6	75.1	**67.6**	**68.6**
LOGP (optimal linear weights)	68.6	32.4	75.2	71.2	81.7	80.1	**68.2**	**68.7**
LOGP (optimized with CGBP)	75.4	43.1	76.9	79.5	87.2	82.1	**74.0**	**74.1**
Bagging (Decision Table)	80.7	48.2	82.9	77.3	89.6	85.5	**77.4**	**77.8**
Bagging (j4.8)	80.0	51.2	81.3	79.6	86.4	87.5	**77.7**	**78.5**
Bagging (1R)	72.7	11.6	39.7	48.5	70.9	80.7	**54.0**	**58.6**
Boosting (Decision Table)	77.3	51.7	73.9	75.0	83.7	85.4	**74.5**	**75.6**
Boosting (j4.8)	83.0	54.2	81.9	81.4	88.9	88.9	**79.7**	**80.6**
Boosting (1R)	61.1	36.2	58.3	69.5	67.4	81.9	**62.4**	**64.7**
Number of Samples	1250	817	701	705	405	1625		5503

tively good, especially considering that the data are clearly not Gaussian [21]. The j4.8 method outperformed the other single classifier methods in terms of both training and test accuracies and achieved an overall accuracy for test data of 70.8%. The test accuracy of the decision table was somewhat lower than that of the CGBP neural network, which achieved a test accuracy of 68.8%.

For the LOP and LOGP, six data classes (corresponding to the information classes in Table IV) were defined in each data source. The AMSS and SAR data sources were modeled to be Gaussian, but the topographic data sources were modeled by Parzen density estimation with Gaussian kernels [6]. From the results for the consensus-theoretic methods in Tables V and VI, it is clear that the LOGP optimized with a neural network outperformed all other consensus-theoretic methods in terms of overall and average training and test accuracies. It is noteworthy that the CGBP optimization increased the overall accuracies of the equally weighted LOGP by approximately 12% (training) and 6% (test), and the LOGP with nonlinearly optimized weights outperformed easily the best single-stage neural network classifiers both in terms of training and test accuracies. In contrast, the CGBP-optimized LOP only gave comparable results to the single-stage CGBP with 30 hidden neurons. However, the best CGBP-optimized LOP results were achieved with zero hidden neurons where the best CGBP-optimized LOGP results were reached with 45 hidden neurons. These results are not surprising. The LOP is a linear combination of posterior probabilities, but the LOGP is nonlinear.

In the case of bagging, 100 iterations were selected for the j4.8, 30 iterations for the decision table, and 700 iterations for 1R. The test accuracies, both for bagging and boosting, as a function of the number of iterations are shown in Fig. 4. The bagging of the decision table and the j4.8 was done directly on the six-class classification problem. However, the 1R classifier had difficulties with the six-class problem, so it was fragmented into multiple two-class problems using *MultiClassClassifier*, as was done for all cases in the classification of the ten-class Colorado dataset. As can be seen in Fig. 4, the test accuracy seemed to have approximately converged in all bagging cases. Evidently, the bagging algorithm improved on the best training or test results given by LOGP when the j4.8 base classifier is used. This result comes as no surprise, since decision tree classifiers are typical unstable classifiers that should perform well in classification by the bagging algorithm. Bagging based on the decision table does almost as well in terms of test accuracies and, in fact, slightly better than bagging based on the j4.8 after 30 iterations.

For boosting, the Adaboost.M1, with 100 iterations for j4.8, was employed. However, after 19 iterations the boosting of the decision table aborted. This demonstrates how strict

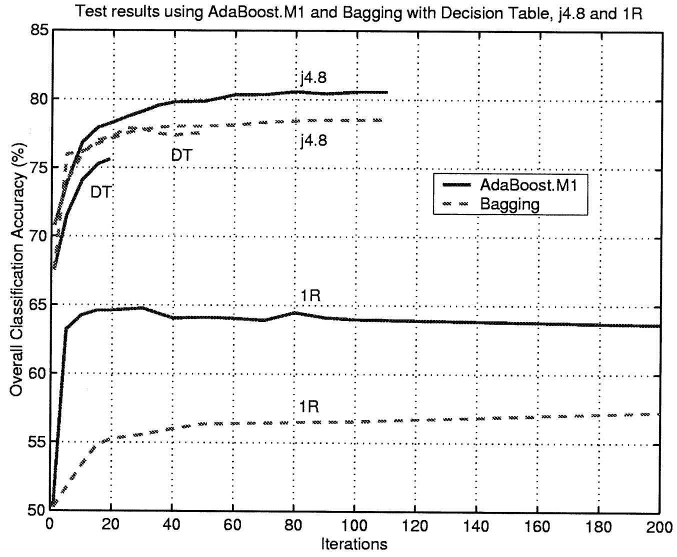

Fig. 4. Anderson River data. Test accuracies as a function of the number of iterations.

the demand for 50% accuracy is for multiclass problems. Nevertheless, the *MultiClassClassifier* was only employed for the 1R base classifier. The fact that *MultiClassClassifier* was used for all bagging and boosting methods on the Colorado dataset, and the 1R method on the Anderson River dataset, but not for the Decision Table and j4.8 methods on the Anderson River dataset, skews the comparison between different methods somewhat. It is expected that, in general, given enough data and computational resources, better accuracies would be achieved by applying the base classifiers directly on the multiclass problem, instead of multiple two-class problems. However, it was deemed necessary for the ten-class Colorado dataset to split it into multiple two-class problems, and the 1R base classifier could not make a single iteration of the AdaBoost.M1 on the Anderson River dataset without applying the *MultiClassClassifier*.

The justification for using different numbers of iterations for each method is that our aim was to compare the methods in terms of the highest achievable test accuracies, rather than the computational requirements.

As can be seen from Tables V and VI, the Adaboost.M1 algorithm improved on the results given by bagging in the case of the j4.8 classifier, but not the decision table. The j4.8 base classifier results are outstanding. These results are the best accuracies achieved for the whole experiment on the Anderson River data. It is of interest to note that the best overall test accuracies were achieved 95 iterations after the training accuracy reached 100%. Similar to the experiment on the Colorado dataset, boosting and bagging, in most cases, gave good improvements on the accuracies of the single classifiers when looked at on a class-by-class basis. Contrary to the Colorado dataset, boosting and bagging the 1R classifier on the Anderson River dataset did have problems. A plausible reason for these problems is that the 1R base classifier, while giving good results on the seven-feature Colorado dataset, is too simple to classify the 22 features of the Anderson River dataset based on just one feature.

IV. Conclusions

In this paper, three multiple classification schemes were investigated. All three schemes performed well and outperformed several single classifiers in terms of accuracies. Therefore, the results presented here demonstrate that multiple classification methods can be considered desirable alternatives to conventional classification methods for classification of multisource remote sensing and geographic data. In particular, the AdaBoost.M1 method yielded the most accurate classifier both in terms of training and test accuracies, when the j4.8 decision tree was used as the base classifier in the case of the Anderson River dataset, and when the 1R method was used as the base classifier in the case of the Colorado dataset. The AdaBoost.M1 did not demonstrate overtraining although it achieved 100% training accuracy. For the Anderson River dataset, the simpler bagging algorithm performed better than AdaBoost.M1 in the case of the decision table base classifier, where the AdaBoost.M1 aborted after only 19 iterations. Bagging does not suffer from the restriction of needing at least 50% accuracy and has the further advantage of needing not as much computational resources as the other methods. The LOGP consensus-theoretic classifier performed well in experiments. Consensus-theoretic classifiers have the potential of being more accurate than conventional multivariate methods in classification of multisource data, since a convenient multivariate model is not generally available for such data. Also, consensus theory overcomes two of the problems with the conventional ML method. First, using a subset of the data for individual data sources lightens the computational burden of a multivariate statistical classifier. Second, a smaller feature set helps in providing better statistics for the individual data sources when a limited number of training samples is available.

Acknowledgment

The Colorado dataset was loaned by R. Hoffer of Colorado State University. The Anderson River SAR/MSS dataset was acquired, preprocessed, and loaned by the Canada Centre for Remote Sensing, Department of Energy Mines, and Resources, of the Government of Canada.

References

[1] T. Lee, J. A. Richards, and P. H. Swain, "Probabilistic and evidential approaches for multisource data analysis," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-25, pp. 283–293, 1987.

[2] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Conjugate-gradient neural networks in classification of multisource and very-high-dimensional data," *Int. J. Remote Sens.*, vol. 14, pp. 2883–2903, 1993.

[3] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[4] K. Tumer and J. Ghosh, "Analysis of decision boundaries in linearly combined neural classifiers," *Pattern Recognit.*, vol. 29, pp. 341–348, 1996.

[5] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 993–1001, Oct. 1990.

[6] J. A. Benediktsson, J. R. Sveinsson, and P. H. Swain, "Hybrid consensus theoretic classification," *IEEE Trans. Geosci. Remote Sensing*, vol. 35, pp. 833–843, July 1997.

[7] J. A. Benediktsson, J. R. Sveinsson, O. K. Ersoy, and P. H. Swain, "Parallel consensual neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 54–65, Jan. 1997.

[8] T. K. Ho, "A theory of multiple classifier systems and its application to visual word recognition," Ph.D. dissertation, State Univ. New York, Buffalo, 1992.

[9] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.

[10] R. E. Schapire, "A brief introduction to boosting," in *Proc. 16th Int. Joint Conf. Artificial Intelligence*, 1999.

[11] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Machine Learning*, 1996.

[12] L. Breiman, "Bagging predictors," Univ. California, Dept. Stat., Berkeley, Tech. Rep. 421, 1994.

[13] K. Tumer and J. Ghosh, "Linear and order statistics combiners for pattern classification," in *Combining Artificial Neural Nets*, A. Sharkey, Ed. New York: Springer-Verlag, 1999, pp. 127–162.

[14] J. A. Benediktsson and P. H. Swain, "Consensus theoretic classification methods," *IEEE Trans. Syst. Man Cybern.*, vol. 22, pp. 688–704, July/Aug. 1992.

[15] R. Kohavi, "The power of decision tables," in *Proc. Euro. Conf. Machine Learning*, 1995.

[16] I. H. Witten and E. Frank, *Data Mining–Practical Machine Learning Tools With Java Implementations*. San Francisco, CA: Morgan Kaufmann, 2000.

[17] J. R. Quinlan, *C4.5, Programs for Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1993.

[18] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Mach. Learn.*, vol. 11, pp. 63–91, 1993.

[19] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, 1962.

[20] I. H. Witten *et al.*. (2001) Weka 3.2—Machine Learning Software in Java. [Online]. Available: http://www.cs.waikato.ac.nz/~ml/weka/.

[21] D. G. Goodenough, M. Goldberg, G. Plunkett, and J. Zelek, "The CCRS SAR/ MSS Anderson River dataset," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-25, pp. 360–367, 1987.

Gunnar Jakob Briem (S'97–A'01) was born in Reykjavik, Iceland in 1972. He received the B.S. degree in electrical engineering in 1999 and the M.S. degree in 2001, both from the University of Iceland, Reykjavik.

In 2001, he joined Dimon Software, an Icelandic IT company. His research interests are signal processing, communications, and pattern recognition.

Mr. Briem was a student member of IEEE throughout his studies at University of Iceland and was Vice-Chairman of the IEEE Student Branch at the University of Iceland from 1998 to 1999.

Jon Atli Benediktsson (S'86–M'90–SM'99) received the Ph.D. degree from the School of Electrical Engineering at Purdue University, West Lafayette, IN, in 1990.

He is currently a Professor of electrical and computer engineering at the University of Iceland, Reykjavik. He is also a Visiting Professor at the School of Computing and Information Systems, Kingston University, Kingston upon Thames, U.K. He has held visiting positions at the Joint Research Centre of the European Commission, Ispra, Italy, Denmark's Technical University (DTU), Lyngby, and the School of Electrical and Computer Engineering, Purdue University. He was a Fellow at the Australian Defence Force Academy (ADFA), Canberra, in August of 1997. His research interests are in pattern recognition, neural networks, remote sensing, image processing, and signal processing, and he has published extensively in those fields

Dr. Benediktsson is currently Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGARS) and Vice President of technical activities in the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS). From 1996 to 1999, he was the Chairman of GRSS's Technical Committee on Data Fusion. He co-edited (with D. A. Landgrebe) a Special Issue on Data Fusion of TGARS (May 1999). He is the current and founding Chairman of the IEEE Iceland Section. In 1991, he received from Purdue University the Stevan J. Kristof Award as outstanding graduate student in remote sensing. In 1997, he was the recipient of the Icelandic Research Council's Outstanding Young Researcher Award, and in 2000, he was granted the IEEE Third Millennium Medal.

Johannes R. Sveinsson (S'86–M'90–SM'02) received the B.S. degree from the University of Iceland, Reykjavik, and both the M.S. (Eng.) and the Ph.D.degrees from Queen's University, Kingston, ON, Canada, all in electrical engineering.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Iceland. From 1981 to 1982, he was with the Laboratory of Information Technology and Signal Processing, University of Iceland. He was a Visiting Research Student at Imperial College of Science and Technology, London, U.K. from 1985 to 1986. At Queen's University, he held teaching and research assistantships and received Queen's Graduate Awards. From November 1991 to 1998, he was with the Engineering Research Institute, University of Iceland, as a Senior Member of the research staff and a Lecturer in the Department of Electrical Engineering, University of Iceland. His current research interests are in systems and signal theory.

Dr. Sveinsson is a member of SIAM.