Multiple Classifier Systems in Remote Sensing: From Basics to Recent Developments

Jon Atli Benediktsson[1], Jocelyn Chanussot[2], and Mathieu Fauvel[1,2]

[1] University of Iceland, Hjardarhagi 2-6, 107 Reykjavik, Iceland
benedikt@hi.is
[2] GIPSA-Lab, INP Grenoble, BP 46 - 38402 St Martin d'Heres, France
jocelyn.chanussot@lis.inpg.fr, mathieu.fauvel@lis.inpg.fr

Abstract. In this paper, we present some recent developments of Multiple Classifiers Systems (MCS) for remote sensing applications. Some standard MCS methods (boosting, bagging, consensus theory and random forests) are briefly described and applied to multisource data (satellite multispectral images, elevation, slope and aspect data) for landcover classification. In a second part, special attention is given to Support Vector Machines (SVM) based algorithms. In particular, the fusion of two classifiers using the spectral and the spatial information, respectively, is discussed in the frame of hyperspectral remote sensing for the classification of urban areas. In all the cases, MCS provide a significant improvement of the classification accuracies. In order to address new challenges for the analysis of remote sensing data, MCS provide invaluable tools to handle situations with an ever growing complexity. Examples include extraction of multiple features from one data set, use of multi-sensor data, and complementary use of several algorithms in a decision fusion scheme.

1 Introduction

Over the past decades, remote sensing has become a central source of information for the observation of the Earth. Numerous satellites have been launched, providing images in different modalities. On one hand, active imagery systems use radar sensors (e.g., synthetic aperture radar, polarimetric or interferometric imagery): an electromagnetic wave is generated and the sensor records the information reflected by the ground surface when illuminated. On the other hand, passive imagery systems use optical sensors (panchromatic, multispectral or hyperspectral images) where the sensor records the information naturally emitted by the ground when illuminated by the sun. Multisource data can also include geographic data such as elevation and slope [7]. All these data have different characteristics, e.g., different spatial and spectral resolutions, different angle of view, and different dates of acquisition. They thus provide complementary information.

Remote sensing data are used in a wide range of applications, including monitoring of the environment, management of major disasters, urban planning,

M. Haindl, J. Kittler, and F. Roli (Eds.): MCS 2007, LNCS 4472, pp. 501–512, 2007.

precision agriculture, and strategic defense issues. In most of these applications, an automatic analysis of the data is required. The first step of the analysis usually consists in a classification at pixel-level, be it (semi-)supervised or not. Numerous algorithms have been proposed in the geoscience and remote sensing community to address these emerging issues. Considering the complexity of the data and the variety of available algorithms, multiple classifier systems (MCS) proved to be of the utmost interest in numerous remote sensing applications, significantly improving the classification performances. The aim of this paper is to present some of the recent issues addressed by multiple classifier systems in remote sensing. Special attention will be paid to classification algorithms based on Support Vector Machines.

Several multiclassifier systems have been used in remote sensing research. Bagging, boosting and consensus theory are among the most commonly used such approaches. Their application to multisource remote sensing data is discussed in Section 2. Based on these approaches, ensemble of classification and regression tree classifiers can be formed, leading to random forest classifier. This strategy used for land cover classification is presented in Section 3.

Support Vector Machines (SVM) have been widely used of late in classification of remote sensing data. Section 4 briefly presents the principle of this machine learning algorithm. The fusion of SVM for classification of hyperspectral data is then addressed, making a joint use of spatial and spectral information. In the conclusion, we also discuss the new trends in the use of MCS for remote sensing applications, such as decision fusion schemes.

2 Boosting, Bagging and Consensus Theory for Multisource Data

The combination of multisource remote sensing and geographic data offers 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 [1], 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, boosting, and consensus-theoretic classifiers. These multiple classifiers have different characteristics.

2.1 Boosting

Boosting is a general and well known method which is used to increase the accuracy of any classifier. In this study, we use the AdaBoost.M1 method which can be used on classification problems with more than two classes [17]. In the beginning of AdaBoost, all patterns have the same weight and the classifier C_1 is the same as the base classifier. If the classification error is greater than 0.5, then the method does not work and the procedure is stopped. A minimum

accuracy is thus required for the base classifier, which can be of considerable disadvantage in multiclass problems. Iteration by iteration, the weight of the samples which are correctly classified goes down. The algorithm consequently concentrates on the difficult samples. At the end of the procedure, T weighted training sets and T base classifiers have been generated. In most cases, the overall accuracy is increased. Many practical classification problems include samples which are not equally difficult to classify. AdaBoost is suitable for such problems. It tends to exhibit virtually no overfitting when the data is noiseless. Other advantages of boosting include that the algorithm has a tendency to reduce both the variance and the bias of the classification. On the other hand, AdaBoost is computationally more demanding than other simpler methods. The lack of robustness to noise is another shortcoming.

2.2 Bagging

Bagging is an abbreviation of ***b****ootstrap* ***agg****regat****ing*** [18]. Bootstrap methods are based on randomly and uniformly collecting m samples with replacement from a sample set of size m. Many different bags are constructed 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, e.g., because bagging always uses resampling instead of re-weighting, it does not change the distribution of the samples and all classes in the bagging algorithm have equal weights during the voting. Furthermore, bagging can be done in parallel, i.e., all the bags can be designed at once. On the other hand, boosting is always done in series, and each sample set is based on the latest weights.

For a particular bag S_i, the probability that a sample from the training set 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, that is, 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, like e.g., a k-NN classifier, then bagging can actually reduce the classification accuracy because each classifier receives less of the training data. 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.

2.3 Consensus Theory

Consensus theory aims at combining single probability distributions to summarize estimates from multiple experts with the assumption that the experts

make decisions based on Bayesian decision theory [8]. The combination formula is called a consensus rule. These 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. The most common 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 is unimodal and less dispersed than the LOP and it processes the data sources independently.

The simplest approach of the weighting scheme consists in giving all the data sources equal weights. Measures of reliability of the different sources can also be used for heuristic weighting. 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. In [9], the statistical consensus models are optimized with neural networks, and achieve improved classification.

2.4 Experimental Results

Experiments [1] were conducted on multisource remote sensing and geographic data from Colorado. These data were originally acquired, preprocessed by Dr. Roger Hoffer from the Colorado State University. Access to the data set is gratefully acknowledged.

The classification was performed on a data set consisting of the following four data sources:

1. Landsat MSS data (4 spectral data channels).
2. Elevation data (in 10 m contour intervals, 1 data channel).
3. Slope data (0-90 degrees in 1 degree increments, 1 data channel).
4. Aspect data (1-180 degrees in 1 degree increments, 1 data channel).

Each channel comprised an image of 135 rows and 131 columns, and all channels were spatially co-registered. The area used for classification is a mountainous area in Colorado. It has 10 ground-cover classes: one class is water; the others are forest types (namely Colorado Blue Spruce, Mountane/Subalpine Meadow, Aspen, Ponderosa Pine, Ponderosa Pine/Douglas Fir, Engelmann Spruce, Douglas Fir/White Fir, Douglas Fir/Ponderosa Pine/Aspen and Douglas Fir/White Fir/Aspen). 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. Reference data were compiled for the area by comparing a cartographic map to a color composite of the Landsat data and also to a line printer output of each Landsat channel. By this method, 2019 reference points (11.4% of the area) were selected comprising two or more homogeneous fields in the imagery for each class. Approximately 50% of the reference samples were used for training, and the rest was used as a test set.

Several single classifiers were applied to the data, namely the minimum Euclidean Distance (MED) classifier and conjugate gradient backpropagation (CGBP) with two and three layers. The base classifiers which were used for bagging and boosting were also trained as single classifiers on the data. These base classifiers were: a decision table, the j4.8 decision tree [19] and the simple classifier 1R [20]. The results are summarized in Table 1.

Table 1. Training and Testing Accuracies in Percentage for the Different Classification Methods

Method	Average Accuracy Training set	Overall Accuracy Training set	Average Accuracy Test set	Overall Accuracy Test set
MED	37.8	40.3	35.5	38.0
Decision Table	73.0	82.8	63.4	77.0
j4.8	81.3	88.0	63.4	77.4
1R	35.9	60.3	34.0	58.8
CGBP (40 hidden neurons)	95.6	96.3	67.0	78.4
LOP (equal weights)	49.3	68.1	46.5	66.4
LOP (heuristic weights)	55.8	74.2	54.9	73.4
LOP (optimal linear weights)	66.2	80.3	66.1	80.2
LOP (optimized with CGBP)	74.6	83.5	72.9	82.2
LOGP (equal weights)	69.2	79.0	69.0	78.7
LOGP (heuristic weights)	69.2	80.5	66.8	79.6
LOGP (optimal linear weights)	65.1	79.7	64.3	80.0
LOGP (optimized with CGBP)	89.1	91.4	75.1	82.3
Bagging with DecisionTable	79.5	88.3	69.3	82.5
Bagging with j4.8	84.3	90.4	69.5	81.7
Bagging with 1R	61.5	74.9	58.9	73.6
Boosting with Decision Table	89.6	91.4	76.1	83.8
Boosting with j4.8	97.6	97.5	72.6	81.5
Boosting with 1R	88.7	90.9	79.4	85.3
Number of Samples		1008		1011

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 but the topographic data sources were modeled by Parzen density estimation with Gaussian kernels. Several different weighting schemes were used for the LOP and LOGP.

In the case of bagging, 100 iterations were selected for the decision table, 10 iterations for j4.8 and 200 iterations for 1R. Adaboost.M1 was employed, with 50 iterations for the decision table, 200 iterations for j4.8 and 60 iterations for 1R. In each case, the 10 class problem was converted into multiple two class problems.

The obtained overall and average accuracies are shown in Table 1 for both the training and the test sets. The multiple classifiers show improvement over all the single classifiers The highest training accuracies were obtained 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 classifiers. In contrast, bagging the 1R gave poor accuracies. The best overall and average accuracies for consensus theoretic classifiers were achieved with the LOGP optimised by

conjugate gradient backpropagation. Those results were comparable in terms of overall accuracies to the best results achieved using bagging.

3 Random Forests

To further improve the classification performances and overcome the shortcomings of the previous approaches (e.g., sensitivity to noise, computational load and the need for parametric statistical modeling of each data source), random forests have been proposed. Random forests are ensembles of tree-type classifiers, that use a similar but improved method of bootstrapping as bagging, and can be considered an improved version of bagging. Random forests have been shown to be comparable to boosting in terms of accuracies, but without the drawbacks of boosting [11]. In addition, the random forests are computationally much less intensive than boosting. Recently, random forests have been applied to classification of hyperspectral remote sensing data [10]. Their approach is implemented within a multiclassifier system arranged as a binary hierarchy and provides good results for a hyperspectral data set with limited training data. Here, we consider random forests for classification of multisource remote sensing and geographic data [2]. It is of great interest since it is not only nonparametric [12], but it also provides a way of estimating the importance of the individual variables (data channels) in the classification.

Random forest is a general term for ensemble methods using tree-type classifiers $h(x, \theta_k), k = 1, \ldots$ where the θ_k are independent identically distributed random vectors and x is an input pattern [11]. In training, the random forest algorithm creates multiple CART-like trees, each trained on a bootstrapped sample of the original training data, and searches only across a randomly selected subset of the input variables to determine a split (for each node). For classification, each tree in the random forest casts a unit vote for the most popular class at input x. The output of the classifier is determined by a majority vote of the trees.

The number of variables is a user-defined parameter that is often blindly selected to the square root of the number of inputs. By limiting the number of variables used for a split, the computational complexity of the algorithm is reduced, and the correlation between trees is also decreased. Finally, the trees in random forests are not pruned, further reducing the computational load. As a result, the random forest algorithm can handle high dimensional data and use a large number of trees in the ensemble. As each tree is only using a portion of the input variables in a random forest, the algorithm is considerably lighter than conventional bagging with a comparable tree-type classifier.

A random forest classifier was applied to the same data set as boosting, bagging and consensus theory, considering the same 10 classes (see Section 2.4). It performed well (overall test set accuracy: 83%), outperforming the single CART classifier (78%), and being comparable to the accuracies obtained by other ensemble methods (Bagging: 83%-decision table, 82%-j4.8, 74%-1R ; Boosting: 84%-decision table, 82%-j4.8, 85%-1R) [2]. However, the random forest classifier

was much faster in training when compared to the ensemble methods, especially boosting. The random forest algorithm does not overfit, and it does not require guidance (although its accuracy can be tweaked slightly by altering the number of variables used for a split). Furthermore, the algorithm can estimate the importance of variables for the classification. Such estimation is of value for feature extraction and/or feature weighting in multisource data classification. The random forest algorithm can also detect outliers, which can be very useful when some of the cases may be mislabeled. With this combination of efficiency and accuracy, along with very useful analytical tools, the random forest classifier is very desirable for multisource classification of remote sensing and geographic data, where no convenient statistical models are usually available.

4 Support Vector Machines (SVM) and Multiple Classifier Systems

4.1 SVM Formulation and the Use of Different Kernel Functions

We first briefly recall the general formulation of SVM classifiers [13]. Let us first consider a two-class problem in a n-dimensional space $\mathbb{R}^n$. We assume that l training samples, $\mathbf{x}_i \in \mathbb{R}^n$ (vector of attributes, or pixel vectors in the case of hyperspectral analysis) are available with their corresponding class labels given by $y_i = \pm 1$, $S = \{(\mathbf{x}_i, y_i) \mid i \in [1, l]\}$. The SVM method consists in finding the hyperplane that maximizes the margin (see Fig. 1), *i.e.*, the distance to the closest training data points in both classes. Noting $\mathbf{w} \in \mathbb{R}^n$ as the vector normal to the hyperplane and $b \in \mathbb{R}$ as the bias, the hyperplane H_p is defined as

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0, \ \forall \mathbf{x} \in H_p \tag{1}$$

where $\langle \mathbf{w}, \mathbf{x} \rangle$ is the inner product between $\mathbf{w}$ and $\mathbf{x}$. If $\mathbf{x} \notin H_p$ then $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ is the distance of $\mathbf{x}$ to H_p. The sign of f corresponds to decision function $y = sgn\,(f(\mathbf{x}))$. Such a hyperplane has to satisfy:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \ \forall i \in [1, l]. \tag{2}$$

For the non-linearly separable case, *slack* variables ξ are introduced to deal with misclassified samples, and (2) becomes:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ \forall i \in [1, l]. \tag{3}$$

Finally, the optimal hyperplane has to jointly maximize the margin $2/\|\mathbf{w}\|$ and minimize the sum of errors $\sum_{i=1}^{l} \xi_i$. This is a convex optimization problem:

$$\min_{\mathbf{w}, \xi_i, b} \left[\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{l} \xi_i \right], \text{ subject to (3)} \tag{4}$$

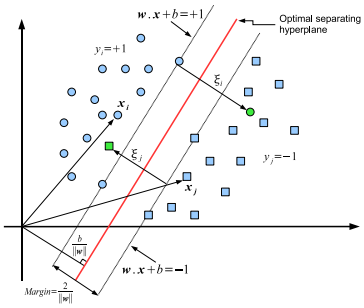

Fig. 1. Classification of non-linearly data by SVMs

where the parameter C balances the minimization of errors and the smoothness (regularization) of the solution, thus directly affecting the generalization capability of the classifier. This primal problem can be solved by considering the dual optimization problem through the use of Lagrange multipliers α_i:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &\text{subject to } 0 \leq \alpha_i \leq C \quad \forall i \in [1, l] \\ &\sum_{i=1}^{l} \alpha_i y_i = 0. \end{aligned} \tag{5}$$

The relation between the primal ($\mathbf{w}$) and the dual parameters (α_i) is given by $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$ [14]. The solution vector is a linear combination of the samples of the training set associated with non-null α_i, which are called *support vectors*. The hyperplane decision function can thus be written as $y_u = sgn\left(\sum_{i=1}^{l} y_i \alpha_i \langle \mathbf{x}_u, \mathbf{x}_i \rangle + b\right)$ where $\mathbf{x}_u$ is an unseen sample. To address non-linear problems while preserving the simplicity of linear models, the input space is projected in higher dimensional feature Hilbert space $\mathcal{H}$ according to a non-linear mapping Φ [15]. The SVM algorithm can be simply considered with the following training samples: $\Phi(S) = \{(\Phi(\mathbf{x}_i), y_i) \mid i \in [1, l]\}$, which leads to a new solution, in which the inner product is: $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Inner products in feature spaces are computed using the *kernel trick* [13], which allows one to work in the mapped kernel space without knowing explicitly the mapping Φ, but only the kernel function k: $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$. This way, the decision function is given by $y_u = sgn\left(\sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}_u, \mathbf{x}_i) + b\right)$.

The most kernels are presented below:

– *Polynomial*. The inner product is computed in the space of all monomials up to degree d: $k_{poly}(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + \theta)^d$. The parameter θ tunes the weight of the higher order polynomial.

– *Gaussian Radial Basis Functions*. This kernel is given by $k_{gauss}(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{z}\|^2\right)$. For this kernel, $k_{gauss}(\mathbf{x}, \mathbf{x}) = 1$. The parameter γ tunes the flexibility of the kernel.

SVMs are designed to solve binary classification problems. Two main approaches have been proposed to address multiclass (N classes) problems [14]:

– *One versus the rest:* N binary classifiers are applied to each class against the others. Each sample is assigned to the class with the *maximum* output.
– *Pairwise classification*: $\frac{N(N-1)}{2}$ binary classifiers are applied on each pair of classes. Each sample is assigned to the class getting the highest number of votes.

The aforementioned multiclass architectures as well as other multicategory strategies that can be applied to classification of hyperspectral images are presented and discussed in [16].

4.2 Joint Spatial and Spectral SVMs

A recent trend in multi- and hyperspectral remote sensing tends to use simultaneously spatial and spectral information, for improved classification performances. One way to address this issue consists in designing a decision fusion scheme. In [6], a landcover multiclass problem is considered on ROSIS data provided by the German Aerospace Agency (DLR) from urban area (N=9 classes, 115 spectral bands ranging from 0.43 to 0.86 μm, 1.3 m per pixel for the spatial resolution). The hyperspectral images are first preprocessed to extract some spatial information and the data are classified using Support Vector Machines (SVM). Another SVM classifier is applied on the initial spectral values, with no spatial information. As a matter of fact, it has been demonstrated that both the spatial and the spectral information are required to actually achieve good classification performances.

Using the *one versus one* classification strategy, 36 binary classifiers are used for each classifier. The standard method consists in combining the results with a majority voting scheme. However, a better multiclassifier system can be designed by storing for each result the actual distance to the hyperplane, following the general idea that it is more useful to have access to the belief of the classifiers rather than the final decision [5]. For a given sample, the larger is the distance to the hyperplane, the more reliable is the label. The most reliable source is thus the one that gives the largest *absolute* distance.

Let us consider that m SVM classifiers are used (in our case 2 classifers: one based on spatial features, one based on spectral information). We have the following results: $\{S_1, S_2, \ldots, S_m\}$, where $S_1 = d_{ij}^1$ is the distance provided by the first SVM classifier which separates class i from class j. The *absolute maximum* decision rule is defined as follows:

$$S_f = AbsMax(S_1, \ldots, S_m) \tag{6}$$

where $AbsMax$ is the logical rules:

$$\begin{aligned} &\textbf{if}(|S_1| > |S_2|, \ldots, |S_m|) \textbf{ then } S_1 \\ &\textbf{else if}(|S_2| > |S_1|, \ldots, |S_m|) \textbf{ then } S_2 \\ &\qquad\qquad\qquad\qquad\vdots \\ &\textbf{else if}(|S_m| > |S_1|, \ldots, |S_{m-1}|) \textbf{ then } S_m. \end{aligned} \tag{7}$$

The agreement of the classifiers can also be taken into account. Each distance is multiplied by the *maximum* probability associated to the two considered classes [21]: $p_i = \frac{2}{N(N-1)} \sum_{j=0, j\neq i}^{N} I(d_{ij})$,where I is the indicator function. The absolute *maximum* rule is applied on these weighted results:

$$S_f = AbsMax\left(\max(p_i^1, p_j^1)S_1, \ldots, \max(p_i^m, p_j^m)S_m\right). \tag{8}$$

A last approach consists in simply applying a majority voting on the $m * N(N-1)/2$ binary classifers used when each of the m classifiers uses the *one versus one* strategy.

The results are summarized on Table 2: the overall and average accuracies are clearly improved by the decision fusion, as well as the Kappa coefficient, with some variations among the different classes.

Table 2. Classification accuracies (%) for the SVMs based on the spectral or the spatial info, or with the 3 fusion operators

	Spectral info. only	Spatial info. only	Abs. Max.	Weighted Abs. max.	Maj. Vot.
Overall Accuracy	81.0	85.2	89.6	**89.7**	86.1
Avergae Accuracy	88.3	90.8	93.6	**93.7**	88.5
Kappa Coefficient	76.2	80.9	86.6	**86.7**	81.8
Asphalt	83.7	**95.4**	93.2	93.0	94.0
Meadows	70.3	80.3	83.9	84.0	**85.3**
Gravel	70.3	**87.6**	82.1	82.2	64.9
Trees	97.8	98.4	**99.7**	**99.7**	99.67
Metalsheets	99.4	**99.5**	**99.5**	99.4	**99.5**
Bare soil	**92.3**	63.7	91.2	91.8	61.6
Bitumen	81.6	**98.9**	97.0	97.2	93.0
Bricks	92.6	95.4	96.4	96.4	**98.8**
Shadows	96.6	97.7	**99.6**	**99.6**	**99.6**

5 Conclusion and Future Trends

Over the past years, multiple classifiers systems have been designed to address numerous applications in remote sensing. Dealing with land cover classification, this paper briefly presented the use of standard algorithms (boosting, bagging and consensus theory) in the case of multi-source data. Random forests is a valuable extension of these algorithms. The focus was then on classifiers based on Support Vector Machines. They provide very promising results in various remote sensing applications and one application was presented in the frame of hyperpsectral data from urban areas.

Future trends in the use of MCS in remote sensing arise from the three following items:

- **Multi-sensor data:** As stated in the introduction, numerous imaging satellites have been launched in the last decades and a lot of new ones are scheduled for the next few years. As a consequence, in many applications, images provided by different sensors are available and MCS can help taking advantage of their complementary characteristics. For instance, in [3], SVM classifiers working on multitemporal radar and optical data, respectively, are aggregated with excellent results.
- **Multiple feature extraction:** To address the difficulty and complexity of the emerging remote sensing applications, such as the accurate classification of very high resolution images from urban areas, multiple features are required. For instance, the spectral information (characterizing the physical nature of the different materials) is complementary to the spatial information (characterizing the shape and geometry of the different objects in the picture). Again, MCS can help taking advantage of their complementary characteristics. An example was described in section 4.2. Another strategy is described in [4], i.e., the spatial features are aggregated with the spectral information prior to classification using feature extraction and dimension reduction techniques. More generally speaking, the joint use of spatial and spectral features for a better understanding of the content of an image, be it multi- or hyperspectral, is one of the key problems in the close future of remote sensing.
- **Fusion of multiple algorithms (*decision fusion*):** Many different algorithms have been proposed in remote sensing research to address various applications. In most of the cases, none of these algorithms strictly outperforms all others. Every algorithm has its own merits, and, again, MCS can help taking advantage of their complementary characteristics. A key issue when designing a decision fusion scheme lies in the reliability of each source (a source being the result of one algorithm). How can one assess this reliability ? In the case of SVM classifiers, as previously described, the distance to the separating hyperplane can be used. In [5], a general framework based on fuzzy logic is presented. A fusion rule incorporating in a flexible way prior knowledge on the different sources and local reliability estimated from the classifiers outputs for each pixel is proposed and tested in the frame of urban areas classification.

The future for novel remote sensing classifiers is closely tied to the design of appropriate MCS, enabling an optimal use of all the available information, with some key issues: 1) How can one handle very high dimensional data?, 2) How can one assess the reliability of one given classifier?, and 3) How can one handle temporal variability in the data?

Acknowledgement. This work was supported in part by the Research Fund of the University of Iceland and in part by the Jules Verne Program of the French and Icelandic governments.

References

[1] Briem G.J., Benediktsson J.A., Sveinsson J.R., Multiple classifiers applied to multisource remote sensing data. IEEE Trans. on Geoscience and Remote Sensing. **vol.40 n.10** (2002) 2291–2299

[2] Gislason P.O., Benediktsson J.A., Sveinsson J.R., Random forests for land cover classification. Pattern Recognition Letters **vol.27** (2006) 294–300

[3] Waske B., Benediktsson J.A., Fusion of support vector machines for classification of multisensor data. to appear in IEEE Trans. on Geoscience and Remote Sensing

[4] Palmason J.A., Benediktsson J.A., Sveinsson J.R., Chanussot J., Fusion of morphological and spectral information for classification of hyperspectral urban remote sensing data. IEEE Geoscience and Remote Sensing Symposium (IEEE IGARSS'06) (2006), Denver, Colorado

[5] Fauvel M., Chanussot J., Benediktsson J.A., Decision fusion for the classification of urban remote sensing images. IEEE Trans. on Geoscience and Remote Sensing. **vol.44 n.10** (2006) 2828–2838

[6] Fauvel M., Chanussot J., Benediktsson J.A., A combined support vector machines classification based on decision fusion. IEEE Geoscience and Remote Sensing Symposium (2006), Denver, Colorado

[7] Benediktsson J.A., Swain P.H., Ersoy, O.K., Neural network approaches versus statistical methods in classification of multisource remote sensing data. IEEE Trans. on Geoscience and Remote Sensing. **vol.28** (1990) 540–542

[8] Benediktsson J.A., Swain P.H., Consensus theoretic classification methods. IEEE Trans. Systems, Man Cybernet.**vol. 22** (1992) 688–704.

[9] Benediktsson J.A., Sveinsson J.R., Swain P.H., Hybrid consensus theoretic classification. IEEE Trans. on Geoscience and Remote Sensing. **vol.35** (1997) 833–843.

[10] Ham J., Chen Y., Crawford M.M., Gosh J., Investigation of the random forest framework for classification of hyperspectral data. IEEE Trans. on Geoscience and Remote Sensing. **vol.43** (2005) 492–501.

[11] Breiman L., Random Forests. Mach. Learn. **vol.40** (2001) 5-32.

[12] Duda R.O., Hart P.E., Stork D., Pattern Classification, second ed. (2001) Wiley, New York.

[13] Boser B. E., Guyon I. M., Vapnik V. N., A training algorithm for optimal margin classifier. *Fifth ACM Annual Workshop on Computational Learning* (1992) 144–152.

[14] Scholkopf B., Smola J., Learning with kernels (2002) MIT Press.

[15] Muller K. R., Mika S., Ratsch G., Tsuda K., Scholkopf B., An introduction to kernel-based learning algorithms. IEEE Trans. on Neural Networks. **vol.12** (2001) 181–202.

[16] Melgani F., Bruzzone L., Classification of hyperspectral remote-sensing images with support vector machines. IEEE Trans. on Geoscience and Remote Sensing. **vol.42** (2004) 1778–1790.

[17] Freund Y., Schapire R. E., Experiments with a new boosting algorithm. Proc. 13th Int. Conf. Machine Learning (1996).

[18] Breiman L., Bagging predictors. Univ. California, Dept. Stat., Berkeley, Tech. Rep. 421 (1994).

[19] Witten I. H., Frank E., Data MiningPractical Machine Learning Tools With Java Implementations. San Francisco, CA: Morgan Kaufmann (2000).

[20] Holte R. C., Very simple classification rules perform well on most commonly used datasets. Mach. Learn. **vol.11** (1993) 63-91.

[21] Wu T., Lin C., Weng R., Probability estimates for multiclass classification by pairwise coupling. Journal of Machine Learning. **vol.5** (2004) 975-1005.