UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

COMBINING PARAMETRIC AND NON-PARAMETRIC ALGORITHMS FOR A PARTIALLY UNSUPERVISED CLASSIFICATION OF MULTITEMPORAL REMOTE-SENSING IMAGES

Lorenzo Bruzzone, Roberto Cossu and Gianni Vernazza

2002

Technical Report # DIT-02-0024

Also: to be published on Information Fusion

Combining Parametric and Non-parametric Algorithms for a Partially Unsupervised Classification of Multitemporal Remote-Sensing Images

Lorenzo Bruzzone*, Roberto Cossu, Gianni Vernazza

Dept. of Information and Communication Technologies

University of Trento, Via Sommarive ,14

I-38050, Povo, Trento, Italy

lorenzo.bruzzone@ing.unitn.it, phone +39-0461-882056, fax +39-0461-881696

Abstract. In this paper, we propose a classification system based on a multiple-classifier architecture, which is aimed at updating land-cover maps by using multisensor and/or multisource remote-sensing images. The proposed system is composed of an ensemble of classifiers that, once trained in a supervised way on a specific image of a given area, can be retrained in an unsupervised way to classify a new image of the considered site. In this context, two techniques are presented for the unsupervised updating of the parameters of a maximum-likelihood (ML) classifier and a radial basis function (RBF) neural-network classifier, on the basis of the distribution of the new image to be classified. Experimental results carried out on a multitemporal and multisource remote-sensing data set confirm the effectiveness of the proposed system.

Keywords: multiple-classifier systems, unsupervised retraining algorithms, maximum-likelihood classifier, radial basis function neural networks, expectation-maximization algorithm.

1 Introduction

The increasing availability of remote-sensing images, acquired periodically by satellite sensors on the same geographical area, makes it extremely interesting to develop monitoring systems

capable of automatically producing and regularly updating land-cover maps of the considered site. The monitoring task can be accomplished by supervised classification techniques, which have proven to be effective categorisation tools [1]-[5]. Unfortunately, these techniques require the availability of a suitable training set (and hence of ground-truth information) for each new image of the considered area to be classified. However, in real applications, it is not possible to rely on suitable ground truth information for each of the available images of the analysed site. Consequently, not all the remote-sensing images acquired on the investigated area at different times can be used for updating the related land-cover maps. In this context, it would be important to develop classification methods capable of analysing the images of the considered site for which no training data are available, thus increasing the effectiveness of monitoring systems based on the use of remote-sensing images.

Recently, the authors faced this problem by proposing an unsupervised retraining technique for maximum-likelihood (ML) classifiers capable of producing accurate land-cover maps even for images for which ground-truth information is not available [6]. This technique allows the unsupervised updating of the parameters of an already trained classifier on the basis of the distribution of the new image to be classified. However, given the complexity inherent with the task of unsupervised retraining, the resulting classifier may be intrinsically

less reliable and less accurate than the corresponding supervised one, especially for complex data sets.

In this paper, in order to define a robust classification system for an unsupervised updating of land-cover maps, we propose: i) to extend the unsupervised retraining technique proposed in [6] to radial basis function (RBF) neural network classifiers; ii) to integrate the resulting unsupervised retraining classifiers in the framework of multiple-classifier systems. In greater detail, the proposed system is based on two different unsupervised retraining classification algorithms: a parametric maximum-likelihood (ML) classifier and a nonparametric radial basis function (RBF) neural-network classifier. Both techniques allow the existing "knowledge" of the classifiers (i.e., the parameters of the classifiers obtained by supervised learning on a first image, for which a training set is assumed available) to be updated in an unsupervised way, on the basis of the distribution of the new image to be categorised. The combination of the above-mentioned classification algorithms is used as a tool for increasing the accuracy and the reliability of the classification maps obtained by each single classifier. Classical approaches to classifier combination are adopted. As compared to previous works [6], the main novelty of this paper consists in the original retraining technique proposed for the RBF classifier and in the multiple classifier architecture used in the context of partially unsupervised classification.

The paper is organized into seven sections. In Section 2 the considered problem is formulated. The architecture of the proposed system is described in Section 3. The unsupervised retraining classifiers are described in Section 4. Section 5 presents the strategies adopted for the combination of the ensemble of unsupervised retraining classifiers considered. Experimental results are given in Section 6. Finally, in Section 7, discussion is provided and conclusions are drawn.

2 Formulation of the Problem

Let $\mathbf{X}_1 = \{x_1^1, x_2^1, .., x_B^1\}$ and $\mathbf{X}_2 = \{x_1^2, x_2^2, .., x_B^2\}$ denote two multispectral images composed of B pixels and acquired in the area under analysis at the time t_1 and t_2, respectively. Let x_j^i be the $1 \times d$ feature vector associated with the j-th pixel of the image $\mathbf{X}_i$ (where d is the dimensionality of the input space). Let X_i be a multivariate random variable that represents the pixel values (i.e., the feature vector values) in $\mathbf{X}_i$. Let us assume that the same set $\Omega = \{\omega_1, \omega_2, ..., \omega_C\}$ of C land-cover classes characterizes the considered geographical area at both t_1 and t_2. This means that in our system only the spatial and spectral distributions of such land-covers classes are supposed to vary (i.e., the set of land-cover classes that characterize the considered site is fixed over time). This assumption is quite realistic in several real applications of classification of remote-sensing data [7]-[9]. Finally,

let us assume that a reliable training set $\mathbf{Y}_1$ is available at t_1, whereas a training set is not available at t_2. This prevents the generation of the t_2 land-cover map, as the training of the classifier on the image $\mathbf{X}_2$ cannot be performed. At the same time, it is not possible to apply the classifier trained on the image $\mathbf{X}_1$ to the image $\mathbf{X}_2$ because, in general, the estimates of the statistical parameters of the classes at t_1 do not provide accurate approximations for the same terms at t_2. This depends on several factors (e.g., differences in the atmospheric and light conditions at the image-acquisition dates, sensor non-linearities, different levels of soil moisture, etc.) that alter the spectral signatures of land-cover classes in different images and consequently the distributions of such classes in the feature space.

It is worth noting that the proposed approach is based on a separate analysis of the two images $\mathbf{X}_1$ and $\mathbf{X}_2$. Consequently, it does not require that the images are accurately co-registrated.

3 Description of the Architecture of the Proposed Classification System

The proposed classification system is based on a multiple-classifier architecture. The choice of this architecture mainly depends on the intrinsic complexity of the unsupervised retraining procedures, which may result in less reliable and less accurate classifiers than the corresponding supervised ones, especially for complex data sets. In this context, the use of a

multiple-classifier approach allows one to integrate the complementary information provided by an ensemble of different classifiers, thus involving a more robust and reliable classification system.

The classifiers composing the ensemble are developed within the framework of the Bayes decision theory. Consequently, the decision rule adopted to classify a generic pixel x_j^1 of the image $\mathbf{X}_1$ can be expressed as [10]:

$$x_j^1 \in \omega_k \quad \text{if} \quad \omega_k = \underset{\omega_i \in W}{arg\,max}\left\{P_1\left(\omega_i / x_j^1\right)\right\} , \tag{1}$$

where $P_1\left(\omega_i / x_j^1\right)$ is the estimate of the posterior probability of the class ω_i at t_1, given the pixel x_j^1. According to (1), the classification of the image $\mathbf{X}_1$ requires the estimation of the posterior probabilities $P_1\left(\omega_i / X_1\right)$ for all classes $w_i \in W$. These estimates involve the computation of a parameter vector θ_1, which represents the "knowledge" of the classifier concerning the distributions of the classes in the feature space (i.e., the status of the classifier at t_1). The number and nature of the vector components will be different depending on the specific classifier used. In our system, we propose to consider two different unsupervised retraining approaches: the former is a parametric approach, which is based on the ML

classifier; the latter consists of a non-parametric technique, which is based on RBF neural networks. Both techniques allow the parameter vectors $\boldsymbol{J}_1^p$ (corresponding to the parametric approach) and $\boldsymbol{J}_1^n$ (corresponding to the nonparametric approach), which are obtained by supervised learning on the first image $\mathbf{X}_1$, to be updated in an unsupervised way.

In the proposed multiple-classifier approach, *N* different classifiers are trained at the time t_1 by using the information contained in the available training set $\mathbf{Y}_1$. In particular, a classical parametric ML classifier [10] and *N-1* different architectures of non-parametric RBF neural networks [5] are used. As a result, a parameter vector $\boldsymbol{J}_1^p$ corresponding to the parametric approach, and the *N-1* parameter vectors $\boldsymbol{J}_1^{n,r}$ ($r=1,\ldots,N-1$) corresponding to the nonparametric RBF neural approach, are derived. Then, at time t_2, the classifiers are retrained in an unsupervised way by using the information contained in the distribution $p(X_2)$ of the new image $\mathbf{X}_2$. At the end of the retraining phase, a new parameter vector $\boldsymbol{J}_2^p$ is obtained for the ML classifier and *N-1* new parameter vectors $\boldsymbol{J}_2^{n,r}$ ($r=1,\ldots,N-1$) are obtained for the *N-1* RBF neural-network architectures considered. Finally, the results provided by different unsupervised retraining classifiers are combined by using a classical multiple-classifier approach.

4 The Proposed Unsupervised Retraining Classifiers

The main idea of the proposed unsupervised retraining approach is that rough estimates of the parameter values that characterize the classes considered at the time t_2 can be obtained by exploiting the parameters of the classifiers estimated at the time t_1 by supervised learning. Such estimates are then updated in an unsupervised way by using the information contained in the distribution $p(X_2)$ of the new image $\mathbf{X}_2$. In the following, a detailed description of the proposed unsupervised retraining technique for the RBF neural-network classifiers is given. Concerning the retraining technique for the ML classifier, we provide only a brief description since it was already proposed in [6].

4.1 The Proposed Retraining Technique for the ML Classifier

In the case of a parametric ML classifier, the vector of parameters that should be estimated for classifying the new image $\mathbf{X}_2$ is given by:

$$\boldsymbol{J}_2^p = \left[\boldsymbol{q}_{2,1}^p, P_2(\omega_1), \boldsymbol{q}_{2,2}^p, P_2(\omega_2), \ldots\ldots, \boldsymbol{q}_{2,C}^p, P_2(\omega_C)\right] \tag{2}$$

where $\boldsymbol{q}_{2,i}^p$ is the vector of the parameters that characterize the conditional density function $p_2(X_2/\omega_i)$ of the class ω_i (e.g., the mean vector $\mu_{2,i}$ and the covariance matrix $\Sigma_{2,i}$ in the Gaussian case). For each class $\omega_i \in \Omega$, the initial values of both the prior probability $P_2^0(\omega_i)$ and the conditional density function $p_2^0(X_2/\omega_i)$ can be approximated by the values computed in the supervised training phase at t_1. Then, such estimates can be improved

by exploiting the information associated with the distribution $p_2(X_2)$ of the new image $\mathbf{X}_2$. In particular, the proposed method is based on the observation that the statistical distribution of the pixel values in $\mathbf{X}_2$ can be described by the following mixed-density distribution:

$$p_2(X_2) = \sum_{i=1}^{C} P_2(\omega_i) p_2(X_2 / \omega_i) \ , \tag{3}$$

where the mixing parameters and the component densities are the *a priori* probabilities and the conditional density functions of the classes, respectively. In this context, the retraining of the ML classifier at the time t_2 becomes a mixture density estimation problem, which can be solved by exploiting the iterative expectation-maximization (EM) algorithm [11]-[14]. The iterative equations to be used are the following:

$$P_2^{t+1}(\omega_k) = \frac{1}{B} \sum_{x_j^2 \in \mathbf{X}_2} P_2^t\left(\omega_k / x_j^2\right) \tag{4}$$

$$m_{2,k}^{t+1} = \frac{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t\left(\omega_k / x_j^2\right) \cdot x_j^2}{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t\left(\omega_k / x_j^2\right)} \tag{5}$$

$$\Sigma_{2,k}^{t+1} = \frac{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t\left(\omega_k / x_j^2\right) \cdot \left(x_j^2 - \mu_{2,k}^{t+1}\right)^T \left(x_j^2 - \mu_{2,k}^{t+1}\right)}{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t\left(\omega_k / x_j^2\right)}. \quad (6)$$

where the superscripts t and $t+1$ refer to the values of the parameters at the current and next iterations, respectively, the superscript T refers to the vector transpose operation, and the estimated posterior probability $P_2^t\left(\omega_k / x_j^2\right)$ is equal to:

$$P_2^t\left(\omega_k / x_j^2\right) = \frac{p_2^t\left(x_j^2 / \omega_k\right) \cdot P_2^t\left(\omega_k\right)}{\sum_{i=1}^{C} p_2^t\left(x_j^2 / \omega_i\right) \cdot P_2^t\left(\omega_i\right)} \quad (7)$$

where the density function $p_2^t\left(x_j^2 / \omega_i\right)$ is computed by using the estimates of the terms $\mu_{2,i}^t$ and $\Sigma_{2,i}^t$ obtained at current iteration.

For each class $w_i \in W$, the estimates obtained at convergence of the EM algorithm are the new parameters of the ML classifier at the time t_2. Since the unsupervised retraining approach for the ML classifier is not the novel aspect of this paper, we refer the reader to [6] for greater details on this method.

4.2 The Proposed Unsupervised Retraining Technique for RBF Neural-Network Classifiers

The proposed nonparametric classifier is based on Gaussian RBF neural networks, which consist of three layers: an input layer, a hidden layer, and an output layer (see Fig. 1). The input layer relies on as many neurons as the input features. The input neurons just propagate the input features to the next layer. Each one of the Q neurons in the hidden layer is associated with a Gaussian kernel function. The output layer is made up of as many neurons as the classes to be recognized. Each output neuron computes a simple weighted summation over the responses of the hidden units for a given input pattern (we refer the reader to [5] for more details on RBF neural-network classifiers).

In the context of RBF neural classifiers, the conditional densities of equation (3) can be written as a sum of contributes due to the Q kernel functions $\boldsymbol{j}_q$ of the neural architecture [14]:

$$p_2(X_2) = \sum_{q=1}^{Q} P_2(\boldsymbol{j}_q) p_2(X_2 / \boldsymbol{j}_q) , \tag{8}$$

where the mixing parameters and the component densities are the *a priori* probabilities and the conditional density functions of the kernels. Equation (8) can be rewritten as:

$$p_2(X_2) = \sum_{i=1}^{C} \sum_{q=1}^{Q} P_2(\omega_i / \mathbf{j}_q) \cdot P_2(\mathbf{j}_q) \cdot p_2(X_2 / \mathbf{j}_q) \,, \tag{9}$$

where the mixing parameter $P_2(\omega_i / \mathbf{j}_q)$ is the conditional probability that the kernel $\mathbf{j}_q$ belongs to class ω_i. In this formulation, kernels are not deterministically owned by classes; so the formulation can be considered as a generalization of a standard mixture model [14]. The value of the weight w_q^i that connects the q-th hidden unit to the i-th output node, can be computed as [14]:

$$w_q^i = P(\omega_i / \mathbf{j}_q) \cdot P(\mathbf{j}_q) \,. \tag{10}$$

By analysing equation (9), it can be noticed that, as for the ML classifier, the retraining of the RBF classifier at time t_2 becomes a parameter estimation problem. In particular, the parameter vector to be estimated is given by:

$$J_2^n = \left[\mathbf{f}_{2,1}, P(\mathbf{j}_1), P_2(\omega_1 / \mathbf{j}_1), \ldots, P_2(\omega_C / \mathbf{j}_1), \ldots, \mathbf{f}_{2,Q}, P(\mathbf{j}_Q), P_2(\omega_1 / \mathbf{j}_Q), \ldots, P_2(\omega_C / \mathbf{j}_Q)\right] \tag{11}$$

where $\mathbf{f}_{2,q}$ is the vector of parameters that characterises the density function $p_2(X_2 / \mathbf{j}_q)$ (e.g., if Gaussian kernel functions are considered, $\mathbf{f}_{2,q}$ is composed of the mean $\boldsymbol{\mu}_{2,q}$ and the width $\sigma_{2,q}$ characterizing the q-th kernel). However, the parameter vector J_2^n is more complex to be estimated than the parameter vector J_2^p related to the ML classifier. In

particular, the presence of the mixing terms $P_2(\omega_i / \varphi_q)$ do not allow the new estimates to be accomplished in a fully unsupervised way. Hence, additional information should be available in order to compute such statistical terms. In the following, we will assume to know the values of the mixing parameters $P(\omega_i / \varphi_q)$; we refer the reader to the Appendix for the description of a technique that exploits the architecture of the proposed system (and, in particular, some of the results provided by the ML classifier) for estimating such parameters. For simplicity, let us assume that all the Q kernel functions $\phi_{2,q}$ are characterized by the same width σ_2. Under the above-mentioned assumptions, it is possible to prove that the following equations (derived by exploiting the EM algorithm) can be applied iteratively to update the RBF neural-network classifier parameters:

$$P_2^{t+1}(\varphi_q) = \frac{1}{B} \sum_{x_j^2 \in \mathbf{X}_2} P_2^t(\varphi_q / x_j^2) \tag{12}$$

$$\mu_{2,q}^{t+1} = \frac{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t(\varphi_q / x_j^2) \cdot x_j^2}{\sum_{x_j^2 \in \mathbf{X}_2} P_2^t(\varphi_q / x_j^2)} \tag{13}$$

$$\boldsymbol{s}_2^{t+1} = \frac{1}{d \cdot B} \sum_{x_j^2 \in \mathbf{X_2}} \left[\sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) \cdot \left\| x_j^2 - \boldsymbol{p}_{2,q}^{t+1} \right\|^2 \right] \tag{14}$$

where the superscripts *t* and *t+1* refer to the values of the parameters at the current and next iterations, respectively, and the estimated posterior probability $P_2^t\left(\boldsymbol{j}_q / x_j^2\right)$ is given by:

$$P_2^t\left(\boldsymbol{j}_q / x_j^2\right) = \frac{p_2^t\left(x_j^2 / \boldsymbol{j}_q\right) \cdot P_2^t\left(\boldsymbol{j}_q\right)}{\sum_{i=1}^{Q} p_2^t\left(x_j^2 / \boldsymbol{j}_i\right) \cdot P_2^t\left(\boldsymbol{j}_i\right)} \tag{15}$$

where the density function $p_2^t\left(x_j^2 / \boldsymbol{j}_i\right)$ is computed by using the estimates of the terms $\boldsymbol{p}_{2,i}^t$ and $\boldsymbol{s}_2^t$ obtained at current iteration.

All the components of $\boldsymbol{J}_2^n$ are initialized according to the values obtained in a supervised way on the t_1 image. It is possible to prove that at each iteration, the log-likelihood function of the estimates increases until a maximum is reached. Although the EM algorithm may converge to a local maximum, its convergence is guaranteed [11]-[14]. The values of the parameters obtained at convergence for each RBF neural classifier are used to analyse the new image to be classified.

5 Multiple Classifier Strategies

We propose the use of different combination strategies to integrate the complementary information provided by the ensemble of unsupervised retraining parametric and non-

parametric classifiers described in the previous section. The use of these strategies for combining the decisions provided by each single classifier results in a more robust behavior in terms of accuracy and reliability of the final classification system.

As stated in Section 3, let us assume that a set of *N* classifiers (an unsupervised retraining ML classifier and *N-1* unsupervised retraining RBF neural classifiers with different architectures) are retrained on the $\mathbf{X}_2$ image in order to update the corresponding parameters by using the procedures described in Section 4. In this context, several strategies for combining the decisions of the different classifiers can be adopted [15], [16]. We will focus on three widely used combination strategies: the *Majority Voting* [15], the *Combination by Bayesian Average* [16], and the *Maximum Posterior Probability* strategies. It is worth noting that, in our case, the use of these unsupervised combination strategies is mandatory because a training set is not available at t_2, and therefore more complex supervised approaches cannot be adopted.

The *Majority Voting* principle faces the combination problem by considering the results of each single classifier in terms of the class labels assigned to the patterns. Hence, a given input pattern receives *N* classification labels from the multiple-classifier system, each label corresponding to one of the C classes considered. The combination method is based on the interpretation of the classification label resulting from each classifier as a "vote" for one of the

C land-cover classes. The data class that receives the largest number of votes is taken as the class of the input pattern.

The second method considered, the *Combination by Bayesian Average* strategy, is based on the observation that for a given pixel x_j^2 in the image $\mathbf{X}_2$ the N classifiers considered provide an approximation of the posterior probability $P_2(w_i / x_j^2)$ for each class $w_i \in W$. Therefore, a possible strategy for combining these classifiers consists in the computation of the average posterior probabilities, i.e.,

$$P_2^{ave}(w_i / x_j^2) = \frac{1}{N} \sum_{n=1}^{N} \hat{P}_2^n(w_i / x_j^2) \tag{16}$$

where $\hat{P}_2^n(w_i / x_j^2)$ is the approximation of the posterior probability $P_2(w_i / x_j^2)$ provided by the n-th classifier. The classification is then carried out according to the Bayes rule by selecting the land-cover class associated with the maximum average posterior probability.

The third method considered (i.e., the *Maximum Posterior Probability* strategy) is based on the same observation of the previous one. However, in this case, the strategy for combining classifiers consists in a winner-takes-all approach: the land-cover class that has the larger posterior probability among all classifiers is taken as the class of the input pattern.

6 Experimental Results

In order to assess the effectiveness of the proposed approach, different experiments were carried out on a data set made up of two multispectral images acquired by the Thematic Mapper (TM) multispectral sensor of the Landsat 5 satellite. The selected test site was a section (412×382 pixels) of a scene including Lake Mulargias on the Island of Sardinia, Italy. The two images used in the experiments were acquired in September 1995 (t_1) and July 1996 (t_2). Figure 2 shows channels 2 of both images.

The available ground truth was used to derive a training set and a test set for each image. Five land-cover classes (i.e., urban area, forest, pasture, water body, and vineyard), which characterize the test site at the above-mentioned dates, were considered. A detailed description of the training and test sets of both images is given in Table 1. To carry out the experiments, we assumed that only the training set associated with the image acquired in September 1995 was available. It is worth noting that the images considered were acquired in different periods of the year. Therefore, in this case, the unsupervised retraining problem turned out to be rather complex.

An ML and two RBF classifiers (one with 60 hidden neurons, i.e., RBF-1, the other with 80 hidden neurons, i.e., RBF-2) were trained in a supervised way on the September 1995

image to estimate the parameters that characterize the density functions of the classes at the time t_1. For the ML classifier, the assumption of Gaussian distributions was made for the density functions of the classes (this was a reasonable assumption, as we considered TM images). In order to exploit the non-parametric characteristic of the two RBF neural classifiers, they were trained using not only the 6 available spectral channels, but also 5 texture features based on the gray-level co-occurrence matrix (i.e., sum variance, sum average, correlation, entropy and difference variance) [17]. These features were computed by using a window size equal to 7x7 and an interpixel distance equal to 1. After the supervised training on the $\mathbf{X}_1$ image, the effectiveness of the classifiers was evaluated on the test sets related to both images (see Table 2). On the one hand, as expected, the classifiers provided high overall classification accuracies for the test set related to the September 1995 image (i.e., 90.97%, 81.79% and 81.74% for the ML, the RBF-1, and the RBF-2 classifiers, respectively). On the other hand, they exhibited very poor performances on the July 1996 test set. In particular, the overall classification accuracy provided by the ML classifier for the July test set was equal to 50.43%, which is not an acceptable result. Also the accuracies exhibited by the two RBF neural classifiers considered are not sufficiently high (i.e., 69.78% and 71.27%).

At this point, the considered classifiers were retrained on the t_2 image (July 1996) by using the proposed unsupervised retraining techniques. The ML and RBF retraining processes converged in 11 and 15 iterations, respectively, taking few minutes of processing on a Sun Ultra80 workstation. The overall and class-by-class accuracies exhibited by the different classifiers after the retraining phase are given in Table 3. By a comparisons of Table 2 and Table 3, one can see that the classification accuracies provided by the considered unsupervised retraining classifiers for the July 1996 test set are sharply higher than the ones exhibited by the single classifiers trained on the September 1995 image (i.e., 92.76% vs 50.43%, 95.34% vs 71.27%, 95.44% vs 69.78% for the ML, the RBF-1, the RBF-2 classifiers, respectively). In greater detail, the retrained classifiers exhibited high accuracies on all land-cover classes, with exception of the vineyard class, which is a minority one.

At this point, the three classifiers were combined according to the strategies described in Section 5. In order to evaluate the accuracy of the resulting classification system, it was applied to the July 1996 test set. The overall and class-by-class accuracies yielded are given in Table 4. As one can see, the overall accuracies provided by all the considered combination strategies (i.e., 95.58%, 95.39%, and 95.75% for the Majority Voting, the Bayesian Average, and the Maximum Posterior Probability strategies, respectively) are

similar to the one yielded by the best-performing classifier composing the ensemble (i.e., 95.44% obtained by the RBF-2 classifier).

It is worth stressing that the objective of the multiple-classifier architecture is not only to increase the accuracy of the classification system but also to increase its robustness. In particular, the combination strategy should allow one to recover the possible failure of a single unsupervised retraining classifier of the ensemble by exploiting the results provided by the other considered classifiers. In order to assess this last issue, an experiment was carried out in which the failure of the retraining process of one of the RBF classifiers (i.e., RBF-1) was simulated. To this end, the RBF classifier with 60 hidden neurons, after being trained on the $\mathbf{X}_1$ image, was not retrained on the $\mathbf{X}_2$ image (let us indicate this classifier as RBF-3). In this condition, the classification accuracy exhibited by the RBF-3 classifier on the July 1996 test set results equal to the one yielded by the RBF-1 classifier on the same test set before the unsupervised retraining phase (see Table 2). As already observed, this overall accuracy (i.e., 71.27%) is not acceptable. At this point, the ML classifier and the RBF-2 and RBF-3 neural classifiers were combined according to the strategies described in Section 5. The accuracies exhibited by the resulting multiple-classifier system are reported in Table 5. As one can see, even though RBF-3 provided low accuracy on the July 1996 test set, all the combination strategies resulted in high classification accuracies, so recovering the simulated

failure of the unsupervised retraining process. In greater detail, the obtained accuracies are comparable to the ones achieved by combining the three "well-retrained" classifiers (i.e., ML, RBF-1, and RBF-2).

7 Discussion and Conclusions

In this paper, the problem of unsupervised retraining of classifiers for the updating of land-cover maps has been addressed in the framework of a multiple-classifier system. The proposed system produces accurate land-cover maps of a specific study area also from images for which a reliable ground truth (and hence a suitable training set) is not available. This is made possible by an unsupervised updating of the parameters of an ensemble of parametric and non-parametric classifiers on the basis of the new image to be classified. In particular, an ML parametric classifier and RBF neural network non-parametric classifiers have been considered. However, given the complexity inherent with the task of unsupervised retraining, the resulting classifiers are intrinsically less reliable and less accurate than the corresponding supervised approaches, especially for complex data sets. Therefore, the use of methodologies for the combination of classifiers has been proposed in order to increase the reliability and the accuracy of single unsupervised retraining classifiers.

Although extensive experiments on other data sets are necessary for a final validation of the method, the results we obtained on the considered data set are very interesting. In particular, they pointed out that the proposed system is a promising tool for attaining high classification accuracies also for images of a given area for which an updated training set is not available.

The presented method is based on the assumption that the estimates of the classifier parameters derived from a supervised training on a previous image of the considered area can represent rough estimates of the class distributions in the new image to be categorised. Then the EM algorithm is applied in order to iteratively improve such estimates on the basis of the global density function of the new image.

It is worth noting that the initial estimates usually cannot be directly used to classify the new image to be analyzed. In fact in practical situation, depending on differences in the atmospheric or light conditions existing between the two acquisition dates, such initial estimates may be significantly different from the true ones. The proposed method copes with this situation, i.e., the EM algorithm is able to improve the initial estimates so that the classification of the new image can be accurately performed. However, in order to minimize the possibility that the retraining does not converge to accurate estimates, if possible, we recommend the application of a pre-processing phase aimed at reducing the differences

between images due to the above-mentioned factors (simple correction algorithms can be adopted).

At the present, the authors are addressing the problem of defining criteria suitable to identify the cases in which the initial estimates of the class distributions are so different from the true ones that may involve a failure of the retraining process.

Appendix I. Estimation of the Mixing Parameters $P_2(\omega_i / \jmath_q)$ for the Retraining of RBF Neural-Network Classifiers

In this appendix, we propose a method for estimating the values of the mixing parameters $P_2(\omega_i / \jmath_q)$ of the RBF neural classifiers (see section 4.2). These parameters can be estimated by exploiting the multiple-classifier architecture of the proposed system. In particular, they can be derived by using the updated parameter vector of the ML classifier. The strategy adopted is the following. Let $\mathbf{L}_2$ be the set of pixels x_j^2 that are most likely correctly classified by the ML classifier. This set can be identified by analysing the estimates of the posterior probability $P_2(\omega_i / x_j^2)$ provided by the ML classification algorithm. Let us consider the j-th pixel x_j^2 of the image $\mathbf{X}_2$ and let us assume that x_j^2 is classified by the ML classifier as belonging to the class ω_k (i.e., $w_k = \arg\max_{\omega_i \in \Omega}\{P_2(w_i / x_j^2)\}$). The pixel x_j^2 is

likely to be correctly classified by the ML classifier (and thus is assigned to the set $\mathbf{L}_2$ and labelled as belonging to the class ω_k) if its estimated posterior probability is above a given threshold (i.e., $P_2\left(\omega_k / x_j^2\right) \geq \alpha$, where $0.5<\alpha<1$ is a real number usually close to 1). The set $\mathbf{L}_2$ is then used to estimate the mixing parameters $P_2\left(\omega_i / \mathbf{j}_q\right)$ according to the following iterative equation:

$$P_2^{t+1}\left(\omega_i / \mathbf{j}_q\right) = \frac{\sum_{x_j^2 \in \mathbf{L}_2^i} P_2^t\left(\mathbf{j}_q / x_j^2\right)}{\sum_{x_j^2 \in \mathbf{L}_2} P_2^t\left(\mathbf{j}_q / x_j^2\right)} \tag{17}$$

where $\mathbf{L}_2^i$ is the subset of $\mathbf{L}_2$ containing the pixels x_j^2 labelled as belonging to the class ω_i. At each step of the EM algorithm used for the unsupervised estimation of the other RBF neural-network parameters [see equations (12), (13), and (14)], also the equation (17) is iterated in order to increase the accuracy in the estimation of the mixing parameters.

Appendix II. Derivation of the Equations for Estimating the Parameters of RBF Neural-Network Classifiers

Equation (12)-(14) and (17) can be derived by maximizing the following log-likelihood function:

$$\mathbf{Y}\left(\mathbf{X}_2 / \mathbf{J}_2^n\right) = \sum_{x_j^2 \notin \mathbf{L}_2} log \sum_{q=1}^{Q}\left[p_2\left(x_j^2 / \mathbf{j}_q\right) P_2\left(\mathbf{j}_q\right)\right] +$$

$$+\sum_{i=1}^{C}\left\{\sum_{x_j^2\in\mathbf{L}_2^i} log\left[\sum_{q=1}^{Q} p_2\left(x_j^2/\boldsymbol{j}_q\right)P_2\left(\boldsymbol{j}_q\right)P_2\left(\boldsymbol{w}_i/\boldsymbol{j}_q\right)\right]\right\} \tag{18}$$

which is equivalent to minimizing the error function $E\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)$:

$$E\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)=-\boldsymbol{Y}\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right) \tag{19}$$

This task can be achieved by means of the technique described in [18]. In particular, let us consider the change ΔE in the function (19) when replacing the parameter values of the current iteration with the one of the next iteration:

$$\boldsymbol{D}E = E^{t+1}\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)-E^{t}\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)=$$

$$=-\sum_{x_j^2\notin\mathbf{L}_2} log\frac{\sum_{q=1}^{Q}\left[p_2^{t+1}\left(x_j^2/\boldsymbol{j}_q\right)P_2^{t+1}\left(\boldsymbol{j}_q\right)\cdot\frac{P_2^t\left(\boldsymbol{j}_q/x_j^2\right)}{P_2^t\left(\boldsymbol{j}_q/x_j^2\right)}\right]}{\sum_{r=1}^{Q}\left[p_2^t\left(x_j^2/\boldsymbol{j}_r\right)P_2^t\left(\boldsymbol{j}_r\right)\right]}+$$

$$-\sum_{i=1}^{C}\left\{\sum_{x_j^2\in\mathbf{L}_2^i}^{B} log\frac{\sum_{q=1}^{Q}\left[p_2^{t+1}\left(x_j^2/\boldsymbol{j}_q\right)P_2^{t+1}\left(\boldsymbol{j}_q\right)P_2^{t+1}\left(\boldsymbol{w}_i/\boldsymbol{j}_q\right)\cdot\frac{P_2^t\left(\boldsymbol{j}_q/x_j^2\right)}{P_2^t\left(\boldsymbol{j}_q/x_j^2\right)}\right]}{\sum_{r=1}^{Q}\left[p_2^t\left(x_j^2/\boldsymbol{j}_r\right)P_2^t\left(\boldsymbol{j}_r\right)P_2^t\left(\boldsymbol{w}_i/\boldsymbol{j}_r\right)\right]}\right\} \tag{20}$$

where $E^t\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)$ and $E^{t+1}\left(\mathbf{X}_2/\boldsymbol{J}_2^n\right)$ are the error functions computed with the parameters estimated at the current and next iterations, respectively. The terms $P_2^t\left(\boldsymbol{j}_q/x_j^2\right)$ are introduced in order to apply the Jensen's inequality. Thanks to such inequality, the following upper-bound can be obtained:

$$\Delta E \le -\sum_{x_j^2 \notin \mathbf{L}_2} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) log \frac{p_2^{t+1}\left(x_j^2 / \boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{j}_q\right)}{\sum_{r=1}^{Q} p_2^t\left(x_j^2 / \boldsymbol{j}_r\right) P_2^t\left(\boldsymbol{j}_r\right) P_2^t\left(\boldsymbol{j}_q / x_j^2\right)} +$$

$$-\sum_{i=1}^{C} \sum_{x_j^2 \in \mathbf{L}_2^i}^{B} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) log \frac{p_2^{t+1}\left(x_j^2 / \boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{w}_i / \boldsymbol{j}_q\right)}{\sum_{r=1}^{Q} p_2^t\left(x_j^2 / \boldsymbol{j}_r\right) P_2^t\left(\boldsymbol{j}_r\right) P_2^t\left(\boldsymbol{w}_i / \boldsymbol{j}_r\right) P_2^t\left(\boldsymbol{j}_q / x_j^2\right)} \qquad (21)$$

We aim at minimizing this bound with respect to the values of the parameters computed at the next iteration. Dropping the terms which depends only on the "old" parameters, the right-hand side of (21) can be rewritten as:

$$\boldsymbol{Q} = -\sum_{x_j^2 \notin \mathbf{L}_2} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) log\left[p_2^{t+1}\left(x_j^2 / \boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{j}_q\right)\right] +$$

$$-\sum_{i=1}^{C} \sum_{x_j^2 \in \mathbf{L}_2^i}^{B} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) log\left[p_2^{t+1}\left(x_j^2 / \boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{j}_q\right) P_2^{t+1}\left(\boldsymbol{w}_i / \boldsymbol{j}_q\right)\right] \qquad (22)$$

and for the Gaussian case:

$$\boldsymbol{Q} = -\sum_{x_j^2 \notin \mathbf{L}_2} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) \cdot \left[log\, P_2^{t+1}\left(\boldsymbol{j}_q\right) - d\, log\, \boldsymbol{s}_2^{t+1} - \frac{\left\| x_j^2 - \boldsymbol{p}_{2,q}^{t+1} \right\|^2}{2\left(\boldsymbol{s}_2^{t+1}\right)^2} \right] +$$

$$-\sum_{i=1}^{C} \sum_{x_j^2 \in \mathbf{L}_2^i}^{B} \sum_{q=1}^{Q} P_2^t\left(\boldsymbol{j}_q / x_j^2\right) \cdot \left[log\, P_2^{t+1}\left(\boldsymbol{j}_q\right) + log\, P_2^{t+1}\left(\boldsymbol{w}_i / \boldsymbol{j}_q\right) - d\, log\, \boldsymbol{s}_2^{t+1} - \frac{\left\| x_j^2 - \boldsymbol{p}_{2,q}^{t+1} \right\|^2}{2\left(\boldsymbol{s}_2^{t+1}\right)^2} \right] . \qquad (23)$$

At this point it is possible to minimize $\boldsymbol{Q}$ (and hence the error function $E^{t+1}\left(\mathbf{X}_2 / \boldsymbol{J}_2^n\right)$) with respect to the "new" parameters. Concerning the parameters $\boldsymbol{s}_2$ and $\boldsymbol{p}_{2,q}$ the minimization

is straightforward and leads to equations (13)-(14). Concerning the parameters $P_2(\boldsymbol{j}_q)$ and $P_2(\boldsymbol{w}_i/\boldsymbol{j}_q)$ the following constraints should be considered:

$$\sum_{q=1}^{Q} P_2(\boldsymbol{j}_q) = 1 \tag{24}$$

$$\sum_{i=1}^{C} P_2(\boldsymbol{w}_i/\boldsymbol{j}_q) = 1 \tag{25}$$

This can be easily done by introducing two Lagrange multipliers. Accordingly equations (12) and (17) can be obtained.

Acknowledgements

This research was supported by the Italian Space Agency (ASI).

References

1. J.A. Richards, Remote sensing digital image analysis, 2nd ed., Springer-Verlag, New York, 1993.
2. J.A. Benediktsson, P.H. Swain, O.K. Ersoy, Neural networks approaches versus statistical methods in classification of multisource remote sensing data, IEEE Transactions on Geoscience and Remote Sensing, 28, (1990), 540-552.

3. J.A. Benediktsson, P.H. Swain, Consensus theoretic classification methods, IEEE Transactions on Systems, Man and Cybernetics, 22, (1992), 688-704.
4. L. Bruzzone, D. Fernàndez Prieto, S.B. Serpico, A neural statistical approach to multitemporal and multisource remote-sensing image classification, IEEE Transactions on Geoscienec and remote Sensing, 37, (1999), 1350-1359.
5. L. Bruzzone, D. Fernàndez Prieto, A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images, IEEE Transactions on Geoscience and Remote-Sensing, 37, (1999), 1179-1184.
6. L. Bruzzone, D. Fernández Prieto, Unsupervised retraining of a maximum-likelihood classifier for the analysis of multitemporal remote-sensing images, IEEE Transactions on Geoscience and Remote Sensing, 39, (2001), 456-460.
7. F. Maselli, M.A. Gilabert, C. Conese, Integration of high and low resolution NDVI data for monitoring vegetation in mediterranean environments, Remote Sensing of Environment, 63, (1998), 208-218.
8. A. Grignetti, R. Salvatori, R. Casacchia, F. Manes, Mediterranean vegetation analysis by multi-temporal satellite sensor data, International Journal of Remote Sensing, 18, (1997), 1307-1318.

9. M.A. Friedl, C.E. Brodley, A.H. Strahler, Maximizing land cover accuracies produced by decision trees at continental to global scales, IEEE Transactions on Geoscience and Remote-Sensing, 37, (1999), 969-977.
10. J.T. Tou, R.C. Gonzalez, Pattern recognition principles, Addison, Reading, MA, 1974.
11. A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of Royal Statistic. Soc., 39**,** (1977), 1-38.
12. B.M. Shahshahani, D. Landgrebe, The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon, IEEE Transactions on Geoscience and Remote-Sensing, 32, (1994), 1087-1095.
13. T.K. Moon, The Expectation-Maximization algorithm, Signal Processing Magazine, 13, (1996), 47-60.
14. D.J. Miller, S.U. Hasan, Combined Learning and Use for a Mixture Model Equivalent to the RBF Classifier, Neural Computation, 10, (1998), 281-293.
15. L. Lam, C.Y. Suen, Application of majority voting to pattern recognition: An analysis of its behavior and performance, IEEE Transactions on System, man and Cybernetics, 27, (1997), 553-568.
16. J. Kittler, M. Hatef, R.P.W. Duin, J. Mates, On combining classifiers, IEEE Transactions on pattern Analysis and machine Intelligence, 20, (1998), 126-239.

17. R.M. Haralick, K. Shanmugan, I. Dinstein I. Textural features for image classification. IEEE Transactions on System, man and Cybernetics, 3, (1973), 610-621.

18. C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press: Oxford, 1995.

FIGURE CAPTIONS

Fig. 1. Standard architecture of a supervised RBF neural-network classifier.

Fig. 2. Channel 5 of the Landsat-5 TM images utilized for the experiments: (a) image acquired in September 1995; (b) image acquired in July 1996.

TABLE CAPTIONS

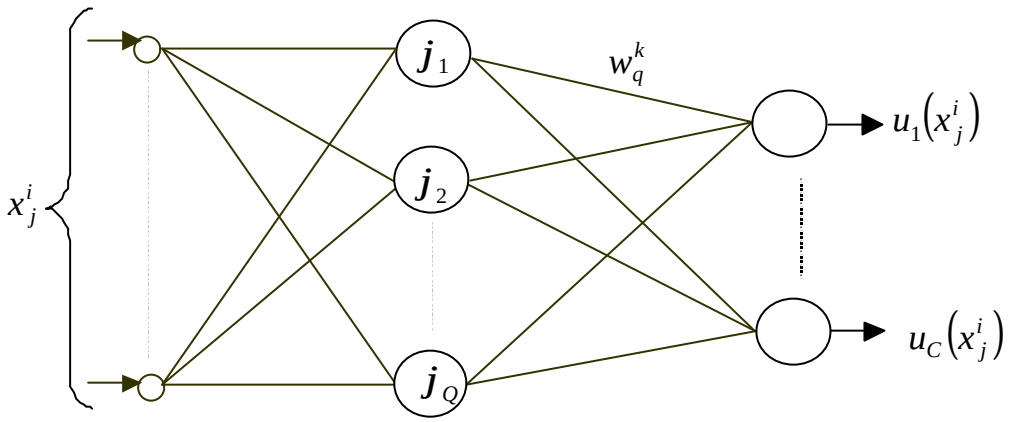
Table 1. Number of patterns in the training and test sets of both the September 1995 and July 1996 images.

Table 2. Overall classification accuracies exhibited by the considered classifiers (trained in a supervised way on the September 1995 image) before the unsupervised retraining.

Table 3. Classification accuracies exhibited by the considered classifiers on the July 1996 test set after the unsupervised retraining.

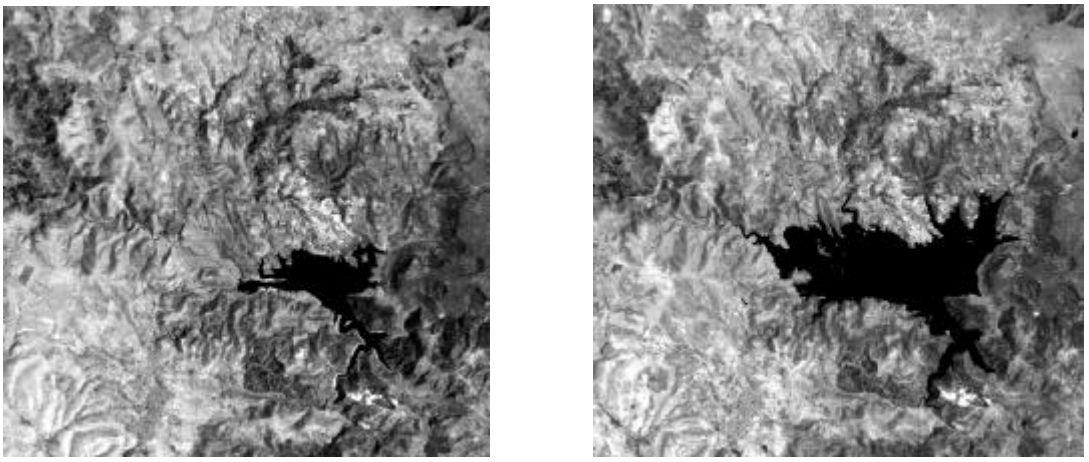
Table 4. Classification accuracies exhibited by the proposed multiple-classifier system on the July 1996 test set.

Table 5. Classification accuracies exhibited by the proposed multiple-classifier system on the July 1996 test set when the failure of the unsupervised retraining of RBF-3 was simulated.



$$u_k\left(x_j^i\right) = \sum_{q=1}^{Q} w_q^k \varphi_q\left(x_j^i\right)$$

Fig. 1



(a) (b)

Fig. 2

Table 1

Land-cover class	Number of patterns	
	Training set	Test set
Pasture	554	589
Forest	304	274
Urban area	408	418
Water body	804	551
Vineyard	179	117
Overall	2249	1949

Table 2

Classification technique	Overall classification accuracy (%)	
	Test set (September 1995)	Test set (July 1996)
ML	90.97	50.43
RBF-1	81.79	71.27
RBF-2	81.74	69.78

Table 3

Land-cover class	Classification accuracy (%) (July 1996 test set)		
	ML	RBF-1	RBF-2
Pasture	94.06	99.83	100.00
Forest	87.22	98.54	98.90
Urban area	93.06	98.56	98.56
Water body	100.00	100.00	100.00
Vineyard	64.10	31.62	31.62
Overall	92.76	95.34	95.44

Table 4

Land-cover class	Classification accuracy (%) (July 1996 test set)		
	Majority Voting	Bayesian Average	Maximum Posterior Probability
Pasture	100.00	99.83	99.32
Forest	98.90	98.90	98.54
Urban area	98.56	98.56	98.08
Water body	100.00	100.00	100.00
Vineyard	34.18	31.62	42.73
Overall	95.58	95.39	95.75

Table 5

Land-cover class	Classification accuracy (%) (July 1996 test set)		
	Majority Voting	Bayesian Average	Maximum Posterior Probability
Pasture	98.47	96.43	90.83
Forest	98.90	98.90	99.27
Urban area	98.56	97.84	98.08
Water body	100	100	100
Vineyard	58.11	52.13	58.11
Overall	96.56	95.43	94.20