Automatic Land Cover Analysis for Tenerife by Supervised Classification using Remotely Sensed Data

Jens Keuchel [a,*], Simone Naumann [a], Matthias Heiler [a],

Alexander Siegmund [b]

[a] *Computer Vision, Graphics, and Pattern Recognition Group*

Dept. of Mathematics and Computer Science

University of Mannheim

D-68131 Mannheim, Germany

[b] *Institute for Geography and Geoecology*

University of Karlsruhe (TH)

D-76128 Karlsruhe, Germany

Abstract

Automatic land cover classification from satellite images is an important topic in many remote sensing applications. In this paper, we consider three different statistical approaches to tackle this problem: Two of them, namely the well-known maximum likelihood classification (ML) and the support vector machine (SVM), are noncontextual methods. The third one, ICM (iterated conditional modes), exploits spatial context by using a Markov random field. We apply these methods to Landsat 5 Thematic Mapper (TM) data from Tenerife, the largest of the Canary Islands. Due to the size and the strong relief of the island, ground truth data could be collected only sparsely by examination of test areas for previously defined land

cover classes.

We show that after application of an unsupervised clustering method to identify subclasses, all classification algorithms give satisfactory results (with statistical overall accuracy of about 90%) if the model parameters are selected appropriately. Although being superior to ML theoretically, both SVM and ICM have to be used carefully: ICM is able to improve ML, but when applied for too many iterations, spatially small sample areas are smoothed away, leading to statistically slightly worse classification results. SVM yields better statistical results than ML, but when investigated visually, the classification result is not completely satisfying. This is due to the fact that no a priori information on the frequency of occurrence of a class was used in this context, which helps ML to limit the unlikely classes.

Key words: Tenerife, land cover analysis, supervised classification, ICM, support vector machines

1 Introduction

The automatic analysis of remotely sensed data has become an increasingly important topic over the last decades. Especially the segmentation of satellite images into regions of different land cover is of major interest: Given data from several spectral bands, one wants to determine for each pixel of the image which type of land cover is present at the corresponding area on the

* Corresponding author. Tel.: +49-621-181-3492; fax: +49-621-181-2744
Email addresses: jkeuchel@ti.uni-mannheim.de (Jens Keuchel), snaumann@uni-mannheim.de (Simone Naumann), heiler@ti.uni-mannheim.de (Matthias Heiler), Alexander.Siegmund@ifgg.uni-karlsruhe.de (Alexander Siegmund).

surface.

The island of Tenerife is a particularly interesting study area for this purpose as, due to its great vertical extent and its position in the Atlantic, it offers a whole range of different vegetational classes (Siegmund & Naumann, 2001). However, the large area and strong relief of the island also raise the problem that training data can only sparsely be collected. Moreover, the spectral reflectances recorded by the satellite sensor may vary within a land cover class depending on slope and aspect. Therefore, appropriate preprocessing of the given seven bands of Landsat 5 Thematic Mapper (TM) data with the help of a digital elevation model is essential for the success of the subsequent classification procedure.

Classification can be performed in several ways, e.g. supervised or unsupervised, parametric or nonparametric, contextual or noncontextual. In this paper, we focus on the application of *supervised* classification algorithms. Therefore, different classes of land cover are defined in advance, and their properties are learned from collected training samples. Then, all data points are classified according to the models defined this way (Richards & Jia, 1999).

Spectral classifiers are usually distinguished into parametric and nonparametric methods (Hubert-Moy et al., 2001). We study supervised classifiers from both categories: On the one hand, the well-known Maximum Likelihood algorithm (ML) is a *parametric* method, which assumes a special probability distribution (usually a Gaussian distribution) of the given data a priori and determines the appropriate parameters (mean vector and covariance matrix) from the training data. Each data point is then assigned to the class for which its values are most likely, i.e., the class with the highest a posteriori proba-

bility. There exists abundant literature on ML and its application to remote sensing data; a comprehensive overview can be found e.g. in Swain & Davis (1978) or Richards & Jia (1999). For the Canary Islands, first results of the application of ML were given by Schweichel (1999) and Siegmund & Naumann (2001).

On the other hand, support vector machines (SVMs) belong to the category of *nonparametric* methods, which do not attempt to model the distribution of the data, but try to separate the different classes by directly searching for adequate boundaries between them. The advantage of this approach is that it generalizes well even if trained with a small number of samples. Support vector machines were developed only recently (Boser et al., 1992) and are not yet routinely applied to remote sensing data. However, some results are reported by Hermes et al. (1999) and Huang et al. (2002).

Both approaches suffer from the drawback that they typically yield noisy segmentations, while in nature larger areas of the same land cover are more likely. This effect is mainly caused by considerable noise in the input data due to reflectances from neighboring pixels. Additionally, mixed pixels composed of more than one land cover class also contribute to this effect, as they often cannot be classified uniquely.

An appealing strategy to overcome these problems is to exploit *spatial context*, where besides spectral values for each pixel information from its neighboring pixels is also evaluated (e.g. Mohn et al., 1987; Gong & Howarth, 1989; Sharma & Sarkar, 1998). In this regard, we tested the ICM (iterated conditional modes) algorithm (Besag, 1986) on the given image data. ICM has already been applied successfully in the context of remote sensing (Solberg et al., 1996; Cortijo

& Pérez de la Blanca, 1998; Hermes et al., 1999; Hubert-Moy et al., 2001). Basically, this parametric method models the prior distribution of the image as a locally dependent Markov random field (Li, 1995; Winkler, 1995), for which the maximum a posteriori estimate is approximated iteratively. After obtaining a first estimate using some non-contextual method like ML or SVM, at every iteration step each pixel is assigned to the class which is most probable given its spectral values and the current labels of its neighbors. This leads to a final segmentation which is smoother and less sensitive to noise than the results of non-contextual methods.

In this work we will compare these three supervised classification algorithms and discuss their advantages and their limits in the context of automatic analysis of Landsat TM imagery of a landscape with strongly rugged terrain for which training data is only given sparsely. The results illustrate some of the inherent problems of labeling remote sensing data, and thus should help researchers to find an appropriate classification procedure in similar situations.

2 Study Area

The research area for the automatic land cover analysis is Tenerife, with about 2050 km^2 the largest of the Canary Islands. Due to its vertical and horizontal extent and its position in the middle Atlantic, Tenerife's climate and vegetation varies strongly. Because of the heterogeneous nature and cultivation area a lot of different classes of land cover have to be considered. On the one hand the classes result from unsupervised classifications (cluster analysis), on the other hand from several ground checks. Due to the coarse resolution of the satellite images and the high number of different land cover classes, it was

unavoidable to reduce this number by fitting similar classes together (Naumann, 2001). At last, after all technical and geographical considerations for the selection of the classes were done, we decided to use $m = 10$ classes in the automatic land cover analysis for Tenerife (cf. Table 1).

Due to the size of the island, the difficult terrain characteristics and the strong presence of mixed vegetation, ground truth data could only sparsely be collected. Altogether, only approximately 4.8% of the area of Tenerife is represented by the training samples. The exact numbers of ground areas and their sizes are given in Table 1.

The satellite images used in this study were taken on August 7, 1988 by the Landsat 5 TM. They contain seven spectral bands with a resolution of 30×30 meters per pixel for each band apart from the thermal band 6, which has a resolution of 120×120 meters. Fig. 1 shows the image of band 5 of the study area as an example.

The spread of the vegetation zones for Tenerife not only depends on the climate situation in the Atlantic Ocean, but also critically on the altitude of the location. In this context, it is helpful to use a Digital Elevation Model (DEM) to encode the altitude data, so that it can be incorporated into the classification process afterwards. To this end, we created a DEM by digitizing the contour-lines of topographical maps in a scale of 1:50,000, and registered this model to the given image data.

3 Methodology

3.1 Preprocessing

In the given Landsat TM scene of Tenerife, some part of the island unfortunately is covered by clouds. In order not to have a negative impact on the classification, these clouds were removed from the input bands in advance. This was accomplished by selecting a threshold temperature level in the thermal band 6, which distinguishes between cool clouds and the warmer ground surface. All spectral values of data points which represent clouds according to this criterion were then replaced with zeros. In this way, the clouds can easily be identified in the subsequent classification without having to deal with them separately. Additionally, all training pixels covered by clouds in the Landsat TM images were removed from the training data.

Due to the strong relief of Tenerife the spectral reflectance values for a class may vary considerably depending on the angle of incoming light. Therefore, we also used a radiometric correction procedure (Civco, 1989) to reduce the influence of topography (given by aspect and slope) on the classification result: With the help of the DEM, a Minnaert correction as presented by Ekstrand (1996) was applied to all spectral bands (apart from band 6). In doing so, five different Minnaert constants were calculated for each band (instead of only one), changing with the cosine of the incidence angle.

The data correcting methods described above may not be sufficient to yield homogeneous distributions of the spectral data given for each class. For example, the reflectance characteristics of the vegetation free class differ on the top of

the mountain and at the coast. For this reason, before starting the supervised classification, we first performed an unsupervised clustering for each training data class to detect subclasses which should be better handled separately. The unsupervised classification method we used for this purpose is the mean shift technique developed by Comaniciu & Meer (2002). The clustering result defines a new number (larger than $m = 10$) of classes on the given training data points, which are then used as input to the supervised classification methods. Finally, we obtain a segmentation of the image into the 10 previously defined classes by relabeling each pixel to that class from which its subclass label was derived.

In the following description of the different classification algorithms, we generally denote by m the number of classes present in the training data set, by n the total number of training vectors, and by d the dimension of these input data vectors.

3.2 Maximum Likelihood Classification (ML)

The maximum likelihood algorithm (e.g. Richards & Jia, 1999) belongs to the class of parametric classification methods. This means that the data is assumed to be distributed according to a previously defined probability model, for which the parameters are determined from a given training set. Each data point $x_j \in \mathbb{R}^d$ is then classified independently by labeling it as belonging to the class $y_j \in \{\omega_1, \ldots, \omega_m\}$ which is most likely given the data x_j, i.e. which has the highest a posteriori probability $P(y_j = \omega_k | x_j)$. To calculate these

probabilities, the Bayes' rule is used:

$$P(y_j = \omega_k | x_j) = \frac{P(x_j | y_j = \omega_k) p(\omega_k)}{\sum_i P(x_j | y_j = \omega_i) p(\omega_i)}. \quad (1)$$

Here, $P(x_j | y_j = \omega_k)$ is the probability that the data point x_j is observed for the given class ω_k, whereas $p(\omega_k)$ denotes the a priori probability of the class ω_k, which is known approximately from ground checks.

To calculate the probability $P(x_j | y_j = \omega_k)$, a multivariate Gaussian distribution with mean vector μ_k and covariance matrix Σ_k is assumed for each class ω_k:

$$P(x_j | y_j = \omega_k) \propto ((2\pi)^d |\Sigma_k|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_j - \mu_k)^T \Sigma_k^{-1} (x_j - \mu_k)\right). \quad (2)$$

Consequently, the a posteriori probability (1) is maximized for a given data point x_j when it is assigned the label ω_{k^*} according to the following classification rule

$$k^* = \arg \max_{k=1,..,m} -\frac{1}{2}(x_j - \mu_k)^T \Sigma_k^{-1} (x_j - \mu_k) - \frac{1}{2} \ln(|\Sigma_k|) + \ln\left(p(\omega_k)\right). \quad (3)$$

This classification rule leads to quadratic decision boundaries between the classes. The corresponding class dependent parameters μ_k and Σ_k are estimated from the training data vectors x_i available for class k:

$$\mu_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i \quad \text{and} \quad \Sigma_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (x_i - \mu_k)(x_i - \mu_k)^T, \quad (4)$$

where n_k denotes the number of training samples for class k.

After the preprocessing of the training data with an unsupervised classification method (see Section 3.1), the number n_k of training samples may become quite small for some classes in comparison to the sample sizes proposed in the

literature (less than 20 vs. between $10d$ and $100d$, cf., e.g., Lillesand & Kiefer (2000)). This may lead to unreliable estimates of the corresponding covariance matrices Σ_k. Nevertheless, these small classes should not be discarded, as they are possibly due to small training areas which give valuable information for the classification process. Moreover, as the training points were usually collected from quite coherent areas (especially for the water class), the variance obtained from the training data may be an underestimate of the true variance of the corresponding class distribution, even for larger numbers n_k. To mitigate these problems, we employed regularization methods (Cortijo & Pérez de la Blanca, 1997; Friedman, 1989) which improve the estimates of the covariance matrices Σ_k by adding appropriate multiples of the identity matrix or of the common covariance matrix $\Sigma = \frac{1}{m}\sum_{i=1}^{m}\Sigma_i$.

A way to handle outliers in the data that do not fit in any of the predefined classes (which may, e.g., be due to incorrect assumptions on the number of classes or missing information in the training data set) is to define an additional *out-class* (Hjort & Mohn, 1984). If a data point does not give a probability higher than a given threshold for any of the classes, it is labeled as belonging to this class. Moreover, a *doubt-class* may also be added to which all data points are assigned that give very similar likelihoods (1) for two or more different classes ω_k.

In our application, three different drawbacks of the ML classifier are obvious:

- A unimodal distribution of the data points is assumed for each class, although multimodal distributions are more likely to be encountered in practice.
- The classification is solely based on the spectral information of each pixel

without using any information given by neighboring pixels, which usually leads to a noisy classification result.

- The fixed form of the decision boundaries prohibits a finer adjustment to the given training data.

The first point is handled by preprocessing the data with an unsupervised classification method as described in Section 3.1. The other two points are tackled by the algorithms described in the next two sections.

3.3 Iterated Conditional Modes (ICM)

Basically, the ICM algorithm (Besag, 1986) uses the same classification method as ML: Based on the Bayes' rule (1), each data point is assigned the label which is most likely. The difference is the underlying probability model: Whereas the ML classifier only uses the spectral information for each data point to calculate the a posteriori probabilities $P(y_j = \omega_k|x_j)$, the ICM algorithm also incorporates spatial context. To this end, it is assumed that the true image is a realization of a locally dependent Markov random field (Li, 1995), so that the probability of a label for a specific data point x_j also depends on the labeling of the neighbors of x_j:

$$P(y_j = \omega_k|x_j) \propto P(x_j|y_j = \omega_k)p(\omega_k|y_{\delta_j}), \tag{5}$$

with δ_j denoting the neighboring data points of x_j. Comparing this to (1), the a priori probability $p(\omega_k)$ of a label is thus replaced by the conditional probability $p(\omega_k|y_{\delta_j})$ which depends on the labels y_{δ_j} surrounding the data point x_j. As proposed by Besag (1986), we use a second-order neighborhood, i.e. δ_j contains the eight data points adjacent to x_j in the image plane.

Since naturally, the probability of the data point x_j to belong to class ω_k should increase with the number of its neighbors that are also labeled as belonging to this class ω_k, the corresponding conditional probability can be stated as

$$p(\omega_k|y_{\delta_j}) \propto \exp\Big(\beta u_j(\omega_k)\Big)p(\omega_k), \tag{6}$$

where $u_j(\omega_k)$ denotes the number of data points in the neighborhood δ_j labeled ω_k, and $p(\omega_k)$ is the original a priori probability of ω_k. The parameter $\beta > 0$ is used to control the smoothness of the resulting classification: The larger its value, the more important is that the data point x_k is labeled according to the majority of its neighbors (see Besag, 1986, for details).

Assuming as for the ML classifier a multivariate Gaussian distribution (2) for each class, substituting (6) and (2) into (5) yields the following classification rule to assign the label ω_{k^*} to pixel x_j:

$$k^* = \arg\max_{k=1,\ldots,m} -\frac{1}{2}(x_j - \mu_k)^T\Sigma_k^{-1}(x_j - \mu_k) - \frac{1}{2}\ln(|\Sigma_k|) + \beta u_j(\omega_k) + \ln\Big(p(\omega_k)\Big). \tag{7}$$

Note that except for the contextual term $\beta u_j(\omega_k)$, the classification rules (7) and (3) coincide.

To compute the numbers $u_j(\omega_k)$, the labeling of the neighbors of x_j must already be known. As it is computationally demanding to calculate (7) for all data points simultaneously, the ICM algorithm estimates the solution iteratively. An initial estimate of the classification $\hat{y}_j$ for each data point x_j is obtained by applying the conventional ML classifier as described in the last section. Afterwards, in a single iteration of the ICM algorithm, each data point is labeled with a new $\hat{y}_j$-value in turn according to (7), using the current $\hat{y}_i$-values of the neighboring data points x_i to calculate $\beta u_j(\omega_k)$. This proce-

dure is applied until convergence, or, in practice, for a predefined number of iterations to arrive at the final classification y_j for each data point.

As the parameters of the Gaussian distributions are estimated from the training data for each class, they usually are not known to be correct. Due to this fact, we re-estimated the class parameters after each iteration by using the current classification result as input to (4). Thus, the class parameters are not static, but are adjusted to the data during the algorithm. For this reason, a good initial estimate is very important, as data points classified incorrectly may influence the new class parameters negatively. To deal with this kind of error propagation, the additional out- and doubt-classes mentioned in the previous section can be used, as they prevent data points to be assigned to a class they do not really fit in.

Other class-properties which are known in advance may also be modeled in the ICM algorithm. For example, the knowledge that some classes do not occur next to one another can be incorporated by means of an additional term $-\beta' u'_j(\omega_k)$ in (7) similar to the contextual term $\beta u_j(\omega_k)$, but with $u'_j(\omega_k)$ in this case denoting the number of neighbors which are labeled with a class that should *not* be neighbored to ω_k. By increasing the parameter $\beta' > 0$ towards infinity with each iteration, the probability for an unwanted neighboring classification becomes very small (cf. Besag, 1986).

3.4 Support Vector Machines (SVM)

Support vector machines (Boser et al., 1992) are discriminative binary classifiers motivated by results from statistical learning theory (Vapnik, 1995).

Discriminative means that, in contrast to the classification algorithms outlined above, SVMs do not attempt to model the probability distribution $P(x_i|y_i = \omega_k)$ of the training vectors. Instead, the decision function $f(x)$ is obtained from the training data points x_i as the solution to the regularization problem (Evgeniou et al., 2000)

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i f(x_i)) + \lambda \|f\|_K^2 \tag{8}$$

in a suitable Hilbert space $\mathcal{H}$ specified by a symmetric, positive definite kernel function K. The class labels y_i of the training points are encoded as numbers $\{+1, -1\}$, such that ideally $y_i = \text{sign}(f(x_i))$. Thus, the first term in (8) calculates the average number of misclassifications on the training data while the second term, weighted by regularization parameter λ, favors smooth decision functions with small norm $\|\cdot\|_K^2$. After the binary decision function f has been determined, each data point x can be labeled according to whether $f(x)$ is positive or negative.

Under mild conditions the solution to (8) can be written in the form

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) + b, \quad \alpha_i \in \mathbb{R}, \tag{9}$$

where b is an offset introduced to allow for arbitrary shifts of the decision boundary. Note that α_i and b can be found efficiently by solving a quadratic program (see, e.g., Cristianini & Shawe-Taylor (2000) for suitable quadratic programming algorithms and for a more detailed introduction to the support vector method).

The choice of the kernel function K is crucial for good classification perfor-

mance. In our experiments we used the Gaussian radial basis function kernel

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \tag{10}$$

which is one of the most commonly used kernel functions.

Various extensions of the binary SVM classification method to problems with more than two classes exist: A popular approach is to split a multiclass problem into multiple binary problems which can be handled by standard SVMs. We decided for the directed acyclic graph method (Platt et al., 2000), where $m(m-1)/2$ binary SVMs are trained to distinguish each pair of classes $(\omega_i, \omega_j), i \neq j$. The resulting ensemble of binary classifiers is then arranged in a directed acyclic graph. A recent study (Hsu & Lin, 2002) has shown this method to be efficient in terms of training and classification speed as well as in terms of accuracy.

4 Results and Discussion

In this section we present the results obtained with the different classification methods. Extensive pre-testing revealed that the best results were achieved when all seven (radiometrically corrected) spectral bands from Landsat 5 TM were simultaneously used as input data (Naumann, 2001). This also includes the thermal band 6, which indirectly provides helpful information concerning the land cover for the area under consideration, as for regions with dense vegetation higher temperatures are recorded than for urban and desert regions. To yield the same resolution as the other spectral bands, the data from band 6 has been resampled accordingly beforehand.

Additionally, we incorporated the altitude data from the DEM as an eighth input band by normalizing the height values to the greyvalue range of the other bands (0–255) and rounding them to nearest integers. Using the elevation data in this way (instead of combining it a posteriori with the classification result) has several advantages: First, the different land use classses are efficiently restricted to the altitude level according to their training data during the whole classification process. Second, misclassification of data points to land use classes which do not exist in the corresponding altitude is prevented. Third, subclasses and outliers in differing heights can be found easier. And finally, no additional postprocessing technique influences the classification results obtained with the methods presented in the last section in either way. At the same time, due to the high correlation of the land cover and the elevation on Tenerife, it is still reasonable to assume normal distributions for the different land cover classes during ML and ICM classification.

Fig. 2 shows a projection of the training data points onto the two-dimensional space spanned by bands 4 and 7. Although this is the two-band combination where the data points visually are spread most, the classes are highly overlapping — with the only exception being the classes representing water and clouds, respectively, in the lower left corner — and thus cannot be separated easily.

As described in Section 3.1, the training data points were preprocessed with an unsupervised clustering method by applying the mean shift technique for each class separately. The clusters containing less than five data points were considered to be outliers and eliminated from the training data. This resulted in $m = 39$ subclasses with a total of $n = 155,928$ training samples as shown in Table 2, with 11 subclasses containing less than 20 samples.

To get an estimate of the accuracy for each of the different classification methods we used a 10-fold cross-validation technique (Kohavi, 1995). In this context, the ground truth data was split randomly into 10 pairs of test and training data. In order to reduce the positive bias introduced by spatial correlations between neighboring pixels we ensured that the distance of each training vector to any test vector was at least three pixels.

We want to point out that we used the same cross-validation data sets for parameter estimation and for reporting test accuracies. This might lead to slightly overoptimistic results (Scheffer & Herbrich, 1997). For this reason, and because accuracy assessment on remote sensing data is a problematic task as such — since the measurements only give the degree of agreement to the collected ground data which does not necessarily represent the whole scene (Foody, 2002) — we also examined each classification visually.

4.1 ML classification

Different choices for the parameters were tested for the ML classification. Concerning the covariance matrices, the best results with overall cross-validation accuracy of $90.21\% \pm 1.11\%$ (we report cross-validation results always with the standard deviation added) were obtained by using the covariance matrices Σ_k calculated from the training data and regularizing them by adding a small constant value $c = 10^{-10}$ on the diagonal to avoid matrices that are not invertible (see Section 3.2). Using larger values for c always gave better results for classes which are represented by less training samples, but also downgraded the results for the other classes. In comparison to that, the use of linear decision boundaries by means of a common $\Sigma = \frac{1}{m}\sum_{i=1}^{m}\Sigma_i$ for all classes only

yielded an overall accuracy of $85.90\% \pm 1.35\%$. Regularization between Σ_k and Σ as proposed by Friedman (1989) also did not reveal better choices for the covariance matrix.

The choice of the a priori probabilities $p(\omega_k)$ for the different subclasses had an important impact on the result. As the a priori probabilities were only given for the 10 predefined classes (see Table 2) and not for their subclasses obtained by mean shift, these probabilities had to be distributed appropriately to the subclasses. In this context, the best statistical results were obtained by weighting the a priori probability of each class by the relative number of training samples from each corresponding subclass. In this way, the subclasses containing many training samples become more important than those which are represented only by a small portion. In contrast to this policy, it is also possible to give each subclass the same weight by dividing the a priori probability of a class by the number of its subclasses. While this results in a more suitable way to weight subclasses which are underrepresented in the training data, it also increases the importance of those classes which have only few subclasses. For the training data given for Tenerife, the latter approach only achieved an overall cross-validation accuracy of $87.92\% \pm 1.23\%$, compared to $90.21\% \pm 1.11\%$ of the first approach. On the other hand, with uniform a priori probabilities for all subclasses an overall accuracy of $88.86\% \pm 1.74\%$ was obtained.

The confusion matrix of the cross-validation result for the ML classifier with the best parameter choice is given in Table 3. Fig. 3 shows the corresponding classification of the whole Landsat TM scene; in this case, all available training samples were used to estimate the parameters. Moreover, black pixels in this figure indicate data points for which the probability of the best labeling was

too small, meaning that they were put in an out-class.

4.2 ICM classification

The ICM algorithm should improve the classification result of the ML classifier by using spatial context. We started ICM with the classification given in Fig. 3, and ran it for 10 iterations, with the value of the smoothness parameter set to $\beta = 1.0$ (cf. eqn. (7)). The interesting result is that cross-validation accuracy only improved after the first iteration, where the mean vectors and covariance matrices estimated from the training data were used unchanged. After the mean vectors and the covariance matrices were re-estimated from the previous classification, the number of correctly classified pixels reduced with every subsequent iteration, with each land cover class being affected equally.

There are three possible explanations for this fact: First, each re-estimation of the parameters is no longer based on the training data only, but on the complete current classification of the scene. Consequently, the parameters are mainly influenced by data points for which the true class membership is unknown. This may lead to a shift of the Gaussian distribution of a land cover class into a direction which does not conform to the training data any longer, especially if many mixed pixels are involved. We observed that in this context, the use of an out-class for data points which do not fit in any class or of a doubt-class for data points with uncertain labeling did not help to increase the number of correctly classified pixels.

A second explanation has been brought up in the context of semi-supervised learning (see Cohen et al., 2002, and references therein). These studies mainly

attribute the increase in classification error to unrealistic modeling assumptions — an explanation which cannot be dismissed in our case either: For some classes, the assumption of a Gaussian distribution may be incorrect.

Finally, another problem is that due to the spatial context used by ICM, pixels initially assigned to the correct class may be relabeled when they are isolated from other pixels of their class and their neighborhood is dominated by another class. Thus, narrow sample areas are smoothed away. For Tenerife, this effect can especially be observed for the settlement class, which contains many small training areas.

The problem of decreasing accuracy may of course be circumvented by avoiding the re-estimation of the parameters and using the original mean vectors and covariance matrices calculated from the training data throughout the algorithm. Indeed, in this case with every iteration a higher accuracy was obtained, and the algorithm converged after a few steps with a total accuracy of about 91%. Although ICM is used in this form in most applications, we refrain from this proceeding here, as it usually needs accurate initial parameter estimations, which we cannot guarantee in the case considered in this paper. Moreover, the results obtained in this way were not convincing visually, as they still contained a lot of noise.

Therefore, as a compromise we analyze the results of the ICM algorithm after 5 iterations, in this way preventing too much deterioration but still allowing a significant amount of smoothing. The corresponding classification is presented in Fig. 4, again with black pixels denoting those data points that were assigned to the out-class. In this case, a total cross-validation accuracy of $88.55\% \pm 1.35\%$ was achieved. For comparison, the corresponding accuracy of the initial

solution obtained with ML was $89.18\% \pm 1.11\%$. Although this indicates that the result of ICM should be slightly worse, the classification is visually more satisfying than the result of ML as it is less noisy. The comparison of the complete confusion matrix for ICM given in Table 4 with the confusion matrix of ML indicates which classes are most problematic: Whereas less pixels of the settlement class were classified correctly (which punctuates the statements made above), too many pixels are assigned to the fayal-brezal class, which is also visible in Fig. 4 at the borders of the masked out clouds. Note that incorporating knowledge about which classes do not occur next to one another (as presented at the end of Section 3.3) had no impact on the result of ICM, as this information is already encoded in the altitude values.

4.3 SVM classification

The free parameters for SVM classification are the width σ of the Gaussian kernel function (eqn. (10)) and the regularization parameter λ of the SVM optimization problem (eqn. (8)). In a first step, we had to determine suitable values for these parameters.

We tested the SVM with different choices for σ on our cross-validation data sets. The training data sets, containing about 50,000 vectors each, led to relatively large instances of SVM multiclass classification problems and thus to long training times. To speed up parameter selection we randomly sampled 1,000 training vectors for each class and each data set. Note that this sampling strategy introduced a slight bias towards classes that were underrepresented in our training data sets.

We found that the choice of σ was not critical for the SVM's classification accuracy: For σ chosen from $\{0.1, 1, 5, 10, 20, 50, 100, 250, 500\}$ only the two smallest and the two largest values led to a degradation in classification accuracy. We chose $\sigma = 50$ for our further experiments. In a similar way we determined the value of the regularization parameter $\lambda = 0.05$ of the SVM optimization problem.

In our experiments we found that preprocessing the data with mean shift did not help the SVM to find better classifications. This is not surprising as, in contrast to ML and ICM, the SVM does not assume the classes being distributed unimodally. As mean shift preprocessing increases the number of classes to be handled we did not employ this technique for SVM classification.

Table 5 gives the confusion matrix for the SVM trained on a subset of the cross-validation training data and tested on the same test data as the ML and ICM classifiers. Note that for the classes where only little training data is available, i.e. the laurisilva class, classification accuracy is severely reduced. The overall accuracy of $93.32\% \pm 0.61\%$ is surprisingly good and probably optimistically biased: We have very little ground truth data at hand and the class variation caught by this sample might be lower than the true variation within the classes. Also, we cannot rule out that remaining spatial correlations between training and test vectors make the confusion matrix look better than it should.

Fig. 5 depicts a classification of the whole island based on 10,000 randomly selected training vectors. Comparing this result to ML classification (Fig. 3) we immediately realize that the SVM classifier spreads a large amount of the settlement class along the south-eastern coast: In this area ground truth

data is rare so the SVM decides for settlement, which has a relatively large inner-class variation. ML also erroneously detects settlement along the south-eastern coast, but as ML utilizes a priori information about class probabilities (eqn. (1)) it assigns most of the unclassified area to the cardonal-tabaibal class which is probably the correct classification.

5 Conclusion

In this paper, we applied three different supervised classification algorithms to find a labeling for a relatively large area into previously defined land cover classes. As has been indicated in Section 3, all methods depend on various parameters (e.g. number of subclasses, number of iterations, kernel width) which directly influence the computational complexity and the classification performance. Although they could also be used with standard parameter settings, some configuration effort is needed to obtain the best classification results. Comparing the computational complexity of the different algorithms, typical runs (on a 2GHz Linux PC, without special algorithmic tuning effort) for a complete classification of the whole scene took about 15 minutes for ML, about 110 minutes for five iterations of ICM and about 90 minutes for SVM based on 10,000 training vectors, with each method requiring more than 400 MB of memory.

Even though only little training data was available, cross-validation tests revealed a high classification accuracy for all classifiers. But the statistical results may be misleading: Even though SVM obtains the highest accuracy, the visual impression is not as satisfying as the ML result because too many pixel were labeled as settlement. In this context, it is a key advantage of ML that a

priori information about class probabilities can be incorporated very easily, so that unlikely classes are suppressed in the final classification. In this direction, combining SVM with a contextual parametric classifier like ICM, which is able to use a priori probabilities and to smooth the noisy classification, would be a useful refinement (Hermes et al., 1999).

Exploiting spatial context by applying ICM to the result of a ML classification was beneficial: A much smoother classification was obtained. But the sparseness of the training data leads to segmentations which statistically decrease in accuracy with each iteration, as the new estimates of the parameters are mainly based on the previous classification of data points for which the true class membership is unknown. In this context, handling mixed pixels more appropriately may play an important role to achieve better results. Instead of hard classification of each pixel, one can use fuzzy labelings (see, e.g., Wang, 1990; Jensen, 1996; Duda & Canty, 2002) during ICM, which allow a data point to have partial class membership. In that way, the estimation of the parameters depends less critically on pixels for which the correct labeling is unknown. Closely related to this technique are probabilistic label relaxation methods (see, e.g., Gong & Howarth, 1989; Li, 1995; Richards & Jia, 1999), which however, usually only consider the original input data during the initialization and not within the subsequent iterations, in contrast to ICM. We did not investigate these methods in this study, because they become quite expensive in terms of memory requirements for large images, as instead of only one value, now as many values as classes are given have to be stored for each data point.

For Tenerife, including the altitude values as an additional input channel improved classification results, as some land cover classes were efficiently re-

stricted to the correct altitude range. However, this additional input channel may also prevent to correctly label data points for which no training data from the corresponding altitude is available. Thus the additional information also enforces the need to collect more appropriate training data. Nevertheless, it is promising to further include other information, like, e.g., texture measures, into the classification process.

For ML classification, the assumption of a unimodal Gaussian distribution is not always correct. In this context, preprocessing the training data with the mean shift algorithm was very helpful. The detected subclasses were much better represented by Gaussian distributions than were the original classes. However, in cases where the basic assumption on the form of the distribution is violated too strongly, the use of a nonparametric classifier like SVM or of another model structure as proposed by Cohen et al. (2002) may be more appropriate.

Acknowledgements

The authors are grateful to Professor Christoph Schnörr for his support and helpful comments. The authors would also like to thank the anonymous reviewers for their constructive criticism and valuable suggestions. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG; grant Schn457/3).

References

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, *48* (3), 259–302.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In: D. Haussler (Ed.), *5th Annual ACM Workshop on COLT* (pp. 144–152). Pittsburgh, PA: ACM Press.

Civco, D. L. (1989). Topographic normalization of Landsat Thematic Mapper digital imagery. *Photogrammetric Engineering and Remote Sensing*, *55* (9), 1303–1309.

Cohen, I., Cozman, F. G., & Bronstein, A. (2002). The effect of unlabeled data on generative classifiers, with application to model selection. Tech. Rep. HPL-2002-140, HP Laboratories Palo Alto.

Comaniciu, D. & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, *24* (5), 603–619.

Cortijo, F. J. & Pérez de la Blanca, N. (1997). A comparative study of some non-parametric spectral classifiers. Applications to problems with high-overlapping training sets. *International Journal of Remote Sensing*, *18* (6), 1259–1275.

Cortijo, F. J. & Pérez de la Blanca, N. (1998). Improving classical contextual classifications. *International Journal of Remote Sensing*, *19* (8), 1591–1613.

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

Duda, T. & Canty, M. J. (2002). Unsupervised classification of satellite imagery: choosing a good algorithm. *International Journal of Remote Sensing*, *23* (11), 2193–2212.

Ekstrand, S. (1996). Landsat TM-based forest damage assessment: Correction for topographic effects. *Photogrammetric Engineering and Remote Sensing*, *62* (2), 151–161.

Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. *Advances in Computational Mathematics*, *13*, 1–50.

Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, *80* (1), 185–201.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, *84* (405), 165–175.

Gong, P. & Howarth, P. J. (1989). Performance analyses of probabilistic relaxation methods for land-cover classification. *Remote Sensing of Environment*, *30*, 33–42.

Hermes, L., Frieauff, D., Puzicha, J., & Buhmann, J. M. (1999). Support vector machines for land usage classification in Landsat TM imagery. In: *Proc. of the IEEE International Geoscience and Remote Sensing Symposium*, Vol. 1 (pp. 348–350). Hamburg.

Hjort, N. L. & Mohn, E. (1984). A comparison of some contextual methods in remote sensing classification. In: *Proc. 18th International Symposium on Remote Sensing of Environment* (pp. 1693–1702). CNES, Paris.

Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, *13*, 415–425.

Huang, C., Davis, L. S., & Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, *23* (4), 725–749.

Hubert-Moy, L., Cotonnec, A., Le Du, L., Chardin, A., & Pérez, P. (2001).

A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote Sensing of Environment*, *75* (2), 174–187.

Jensen, J. R. (1996). *Introductory Digital Image Processing: A Remote Sensing Perspective*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proc. of the 14th International Joint Conference on Artificial Intelligence* (pp. 1137–1145). San Francisco, CA: Morgan Kaufmann.

Li, S. Z. (1995). *Markov Random Field Modeling in Computer Vision.* Springer-Verlag.

Lillesand, T. M. & Kiefer, R. W. (2000). *Remote Sensing and Image Interpretation*, 4th ed. New York: Wiley.

Mohn, E., Hjort, N. L., & Storvik, G. O. (1987). A simulation study of some contextual classification methods for remotely sensed data. *IEEE Trans. Geoscience and Remote Sensing*, *25* (6), 796–804.

Naumann, S. (2001). Satellitengestützte Vegetations- und Landnutzungsanaly-se von Tenerife (Kanarische Inseln) unter Einsatz eines Geographischen Informationssystems. Materialien zur Geographie 33, Universität Mannheim.

Platt, J., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In: S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems*, Vol. 12 (pp. 547–553). Cambridge, MA: MIT Press.

Richards, J. A. & Jia, X. (1999). *Remote Sensing Digital Image Analysis: An Introduction*, 3rd ed. Berlin: Springer.

Scheffer, T. & Herbrich, R. (1997). Unbiased assessment of learning algo-

rithms. In: *Proceedings of the Fifteenth Joint International Conference on Artificial Intelligence* (pp. 798–803). Nagoya, Japan.

Schweichel, R. (1999). Die Vegetation der Kanareninsel El Hierro. Anwendung von Fernerkundungsdaten (SPOT 2). Carl von Ossietzky-Univ. Oldenburg: BIS-Verlag.

Sharma, K. M. S. & Sarkar, A. (1998). A modified contextual classification technique for remote sensing data. *Photogrammetric Engineering & Remote Sensing*, *64* (4), 273–280.

Siegmund, A. & Naumann, S. (2001). Der Einsatz satellitenbildgestützter Klassifikationsverfahren zur Analyse von Landnutzungsstrukturen auf Teneriffa — Ein aktueller Beitrag zur naturräumlichen Landschaftsgliederung. *Geoöko*, *22*, 103–116.

Solberg, A. H. S., Taxt, T., & Jain, A. K. (1996). A Markov random field model for classification of multisource satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *34* (1), 100–113.

Swain, P. H. & Davis, S. M. (Eds.) (1978). *Remote Sensing: The Quantitative Approach.* New York: McGraw-Hill.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* New York: Springer-Verlag.

Wang, F. (1990). Fuzzy supervised classification of remote sensing images. *IEEE Trans. on Geoscience and Remote Sensing*, *28* (2), 194–201.

Winkler, G. (1995). *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Vol. 27 of *Appl. of Mathematics.* Heidelberg: Springer-Verlag.

Table 1

Number of training areas and corresponding number of sample data points for the ten different land cover classes used in this study.

Class name (description)	Training areas	Data points
Cardonal-Tabaibal (shrub, euphorbia, sparse vegetation)	63	9780
Fayal-Brezal (heather, copses, degradation of Laurisilva)	20	3282
Laurisilva (forest, laurels)	11	1656
Pinar (forest, pines)	67	23206
Retamar-Codesar (desert landscape, thickets)	16	2433
Rocks without vegetation	25	25336
Plantation	14	758
Settlement	42	15573
Water	5	74307
Clouds	2	9003
Total	265	165334

Table 2

Number of subclasses obtained with mean shift, corresponding total number of training points, and a priori probabilities for each predefined class.

Class	Number of subclasses	Number of training pts.	A priori probability
Cardonal-Tabaibal	2	8077	10.0%
Fayal-Brezal	5	1695	2.0%
Laurisilva	2	124	0.5%
Pinar	5	21063	7.0%
Retamar-Codesar	1	2431	2.0%
Rocks	10	24941	5.0%
Plantation	4	713	0.5%
Settlement	8	13574	3.0%
Water	1	74307	38.0%
Clouds	1	9003	32.0%
Total	39	155928	

Table 3

Cross-validation results for the ML classifier. Each row of the confusion matrix shows how the test data from one class is labeled, whereas each column shows which data points are labeled as belonging to the corresponding class.

Class	C.-T.	F.-B.	Laur.	Pinar	R.-C.	Rocks	Plant.	Settl.	Water	Clouds	Acc. (in %)
Card.-Tab.	813	18	1	59	0	1	0	69	0	0	84.6
Fay.-Brez.	28	138	3	24	0	1	0	3	0	0	70.1
Laurisilva	0	3	2	4	0	0	0	0	0	0	22.2
Pinar	59	56	5	2448	2	43	0	1	0	0	93.7
Ret.-Cod.	0	0	0	1	264	25	0	0	0	0	91.0
Rocks	28	0	0	65	105	2991	0	7	0	0	93.6
Plantation	9	0	0	0	0	0	71	9	0	0	79.8
Settlement	308	37	0	0	0	3	2	1267	0	0	78.4
Water	0	0	0	0	0	0	0	0	524	0	100.0
Clouds	0	0	0	0	0	0	0	0	0	503	100.0
Acc. (in %)	65.3	54.8	18.2	94.1	71.2	97.6	97.3	93.4	100.0	100.0	90.21

Table 4

Cross-validation results for the ICM classifier. In contrast to ML, the confusion matrix contains an additional column to indicate pixels that were assigned to the out-class.

Class	C.-T.	F.-B.	Laur.	Pinar	R.-C.	Rocks	Plant.	Settl.	Water	Clouds	Out	Acc. (in %)
Card.-Tab.	811	31	8	66	0	0	1	43	0	0	1	84.4
Fay.-Brez.	27	143	1	26	0	0	0	0	0	0	0	72.6
Laurisilva	0	3	6	0	0	0	0	0	0	0	0	66.7
Pinar	47	59	19	2461	5	22	0	0	0	0	1	94.2
Ret.-Cod.	0	0	0	0	276	14	0	0	0	0	0	95.2
Rocks	28	0	0	78	113	2939	0	0	0	0	38	92.0
Plantation	6	0	0	0	0	0	76	7	0	0	0	85.4
Settlement	390	54	1	0	0	1	5	1116	1	0	49	69.0
Water	0	0	0	0	0	0	0	0	524	0	0	100.0
Clouds	0	0	0	0	0	0	0	0	0	503	0	100.0
Acc. (in %)	62.0	49.3	17.1	93.5	70.1	98.8	92.7	95.7	99.8	100.0		88.55

Table 5

Cross-validation results for the SVM classifier. The SVM was trained on a randomly selected subset of the cross-validation training data. The test data is the same as for the ML and ICM classifiers.

Class	C.-T.	F.-B.	Laur.	Pinar	R.-C.	Rocks	Plant.	Settl.	Water	Clouds	Acc. (in %)
Card.-Tab.	860	27	1	5	0	0	5	63	0	0	89.5
Fay.-Brez.	5	166	5	12	0	0	0	9	0	0	84.3
Laurisilva	1	4	4	0	0	0	0	0	0	0	44.4
Pinar	24	86	5	2459	10	30	0	0	0	0	94.1
Ret.-Cod.	0	0	0	1	279	10	0	0	0	0	96.2
Rocks	12	0	0	85	130	2963	0	6	0	0	92.7
Plantation	3	0	0	0	0	0	81	5	0	0	91.0
Settlement	96	24	0	0	0	0	4	1493	0	0	92.3
Water	0	0	0	0	0	0	0	0	524	0	100.0
Clouds	0	0	0	0	0	0	0	0	0	503	100.0
Acc. (in %)	85.9	54.1	26.7	96.0	66.6	98.7	90.0	94.7	100.0	100.0	93.32

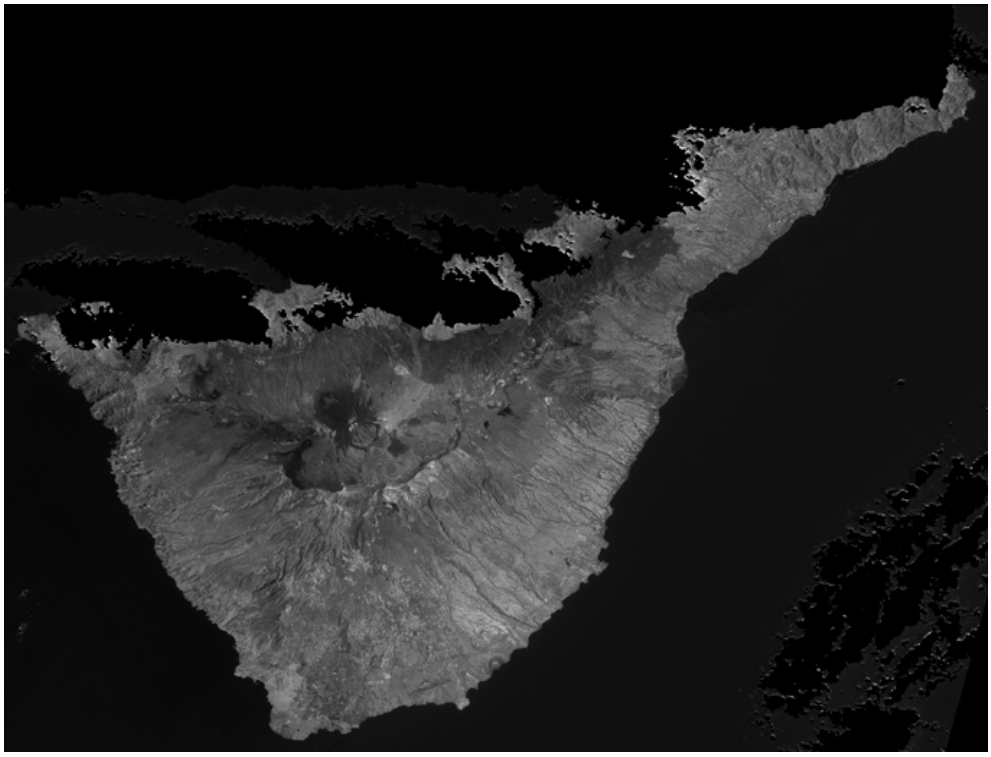

Fig. 1. Landsat 5 TM image of Tenerife, from August 7, 1988, band 5 (clouds removed). Size of the scene: 2728×2073 pixels.

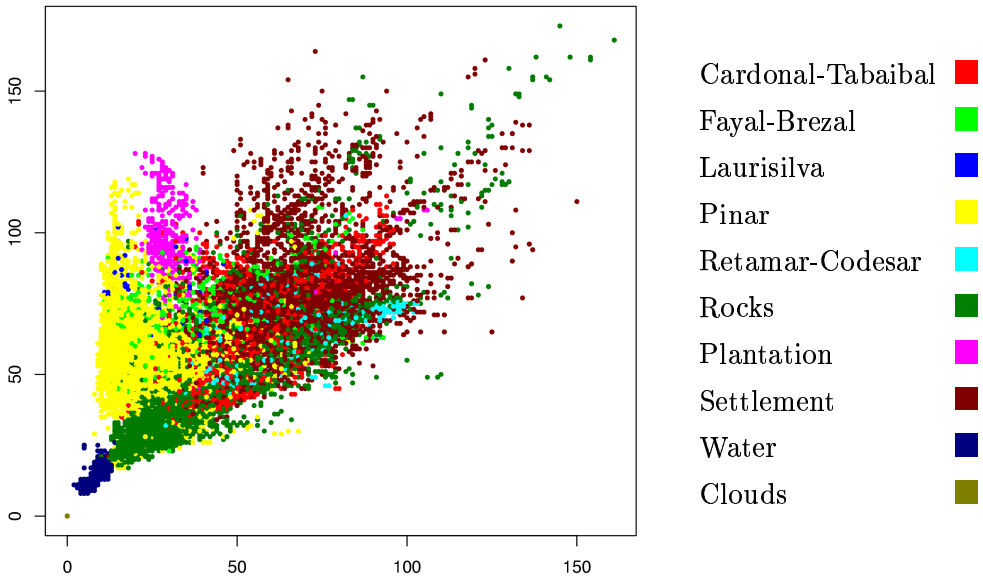

Fig. 2. Projection of the training data on bands 4 (bottom) and 7 (left): All classes are highly overlapping.

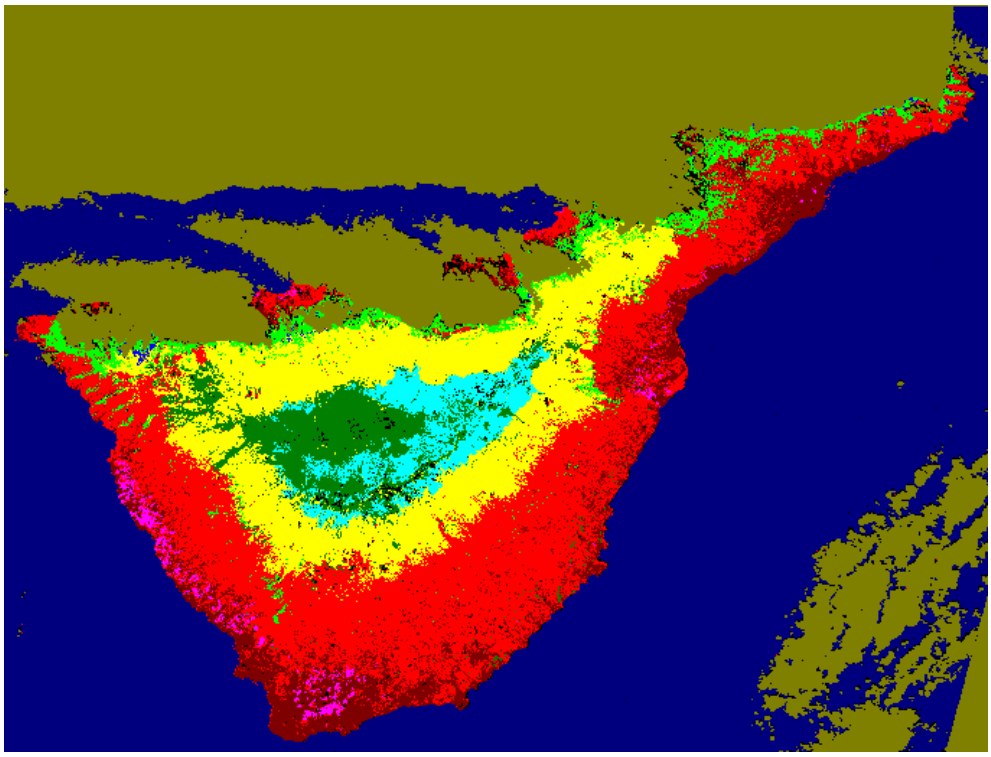

Fig. 3. Result of the Maximum Likelihood classifier, using all available training samples. Black pixels indicate data points assigned to the out-class.

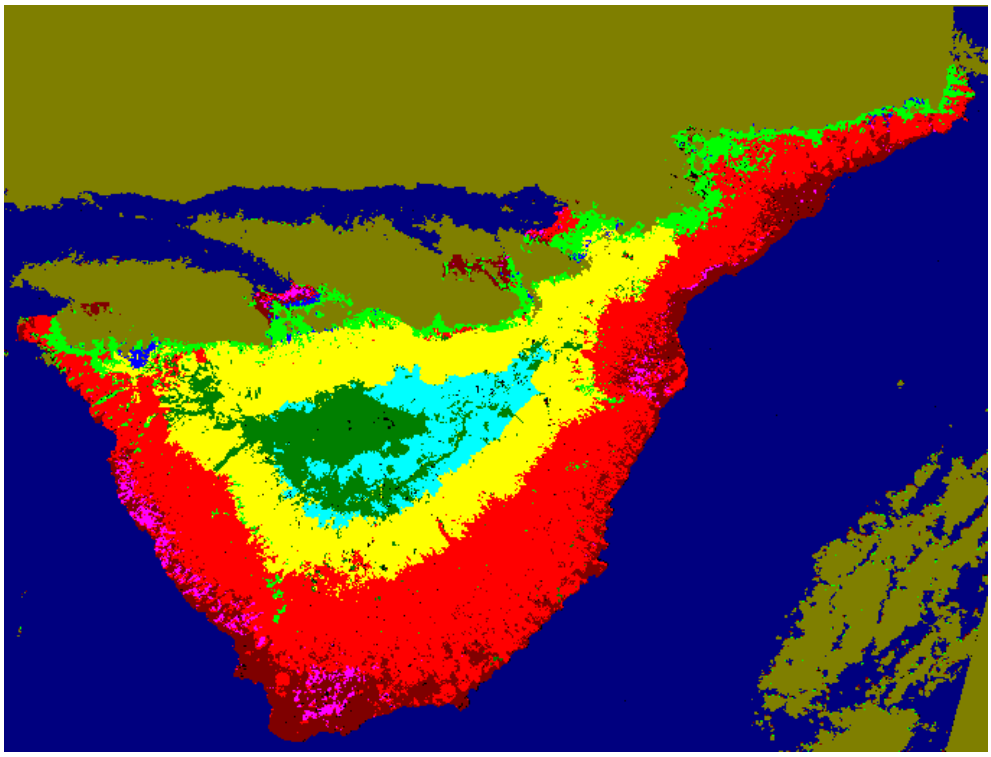

Fig. 4. Result of the ICM classifier after 5 iterations, using the result of the ML classifier as initial estimate. Black pixels indicate data points assigned to the out-class.

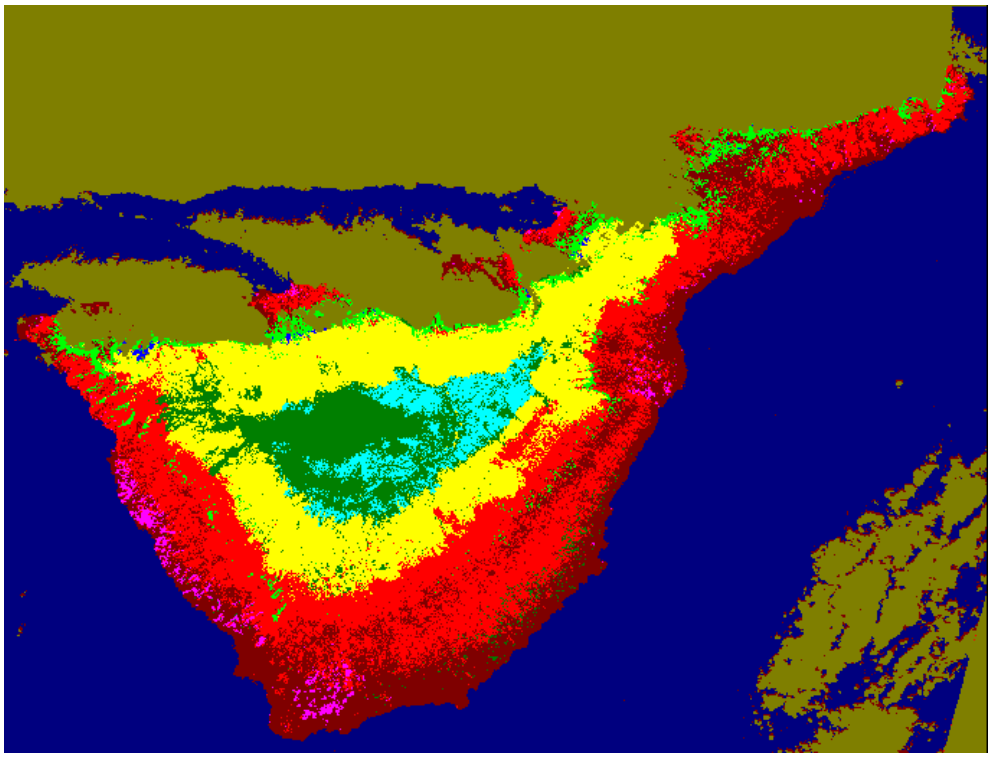

Fig. 5. Result of the SVM classifier trained on a subset of 10,000 randomly selected training vectors.