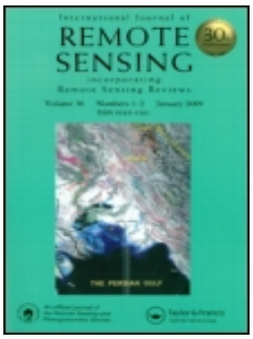

International Journal of Remote Sensing

ISSN: 0143-1161 (Print) 1366-5901 (Online) Journal homepage: https://www.tandfonline.com/loi/tres20

Unsupervised classification of satellite imagery: Choosing a good algorithm

T. Duda & M. Canty

To cite this article: T. Duda & M. Canty (2002) Unsupervised classification of satellite imagery: Choosing a good algorithm, International Journal of Remote Sensing, 23:11, 2193-2212, DOI: 10.1080/01431160110078467

To link to this article: https://doi.org/10.1080/01431160110078467

INT. J. REMOTE SENSING, 2002, VOL. 23, NO. 11, 2193–2212

Unsupervised classification of satellite imagery: choosing a good algorithm

T. DUDA and M. CANTY*

Forschungszentrum Jülich GmbH, D-52425 Jülich, Germany

(*Received 11 January 2000; in final form 26 June 2001*)

Abstract. In the context of land-cover classification with multispectral satellite data several unsupervised classification (clustering) algorithms are investigated and compared with regard to their ability to reproduce ground data in a complex landscape. Ground data is extended to the entire scene using a supervised neural network classification algorithm. The clustering algorithms examined are K-means, extended K-means, agglomerative hierarchical, fuzzy K-means and fuzzy maximum likelihood. Fuzzy clustering is found to perform best relative to a reference scene obtained with the Landsat Thematic Mapper 5 (TM5) platform.

1. Introduction

Supervised classification of multispectral remote sensing imagery is commonly used for land-cover determination. Within this context it is very important to define training areas which adequately represent the spectral characteristics of each class in the image to be classified, as the quality of the training set has a significant effect on the classification process and its accuracy (Chuvieco and Congalton 1988). The process of finding and verifying training areas can be rather labour-intensive, since the analyst must select representative pixels for each of the classes. This must be done by visual examination of the image data and by information extraction from additional sources such as field data or existing maps (Schowengerdt 1997).

Unlike supervised classification, clustering methods (or unsupervised methods) require no training sets at all. Instead they attempt to find the underlying structure automatically by organizing the data into classes sharing similar, i.e. spectrally homogeneous, characteristics. The analyst 'simply' needs to specify the number of clusters present. Such procedures play an especially important role when very little *a priori* information about the data is available. Cluster analysis provides a useful method for organizing a large set of data so that the retrieval of information may be made more efficiently. A primary objective of using clustering algorithms for pre-classification of multispectral remote sensing data in particular is to obtain optimum information for the selection of training regions for subsequent supervised land-use segmentation of the imagery.

*Corresponding author, e-mail: m.canty@fz-juelich.de

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online © 2002 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/01431160110078467

Against this background, five representative clustering algorithms—both conventional and fuzzy logic based—are investigated and compared for land-cover classification of multispectral Landsat Thematic Mapper 5 (TM5) images. These algorithms have been implemented as part of the satellite image classification environment NNSat, which was written in the programming language Delphi in our group at the Research Centre Jülich (Canty 1999).

It is notoriously difficult to assess the results of clustering algorithms in remote sensing. Usually qualitative, subjective criteria are applied, such as homogeneity of the segments, degree of fragmentation and general plausibility. As a yardstick for quantitative comparison we have chosen to use a supervised classification of a reference image, performed without the benefit of clustering but with an extensive ground data database obtained at the time of image acquisition. For this purpose we have selected imagery acquired by Landsat TM5 in the region of Bonn, Germany in July 1997. This scene represents a typically complex central European landscape, consisting of many small mixed agricultural areas, commercial forests, a high density of villages, towns and cities and a dense network of roads and railways. A false colour composite image derived from the first three principal components of the satellite image is shown in figure 1.

In the next section, a simple cost function is derived which then serves, in §3, as a basis for the explanation of the five clustering algorithms investigated. Section 4 discusses briefly the problem of choice of the overall number of clusters, and finally, in 5 the algorithms are evaluated by comparing them pixel-wise with the supervised classification of the same scene.

2. A cost function

We begin with the assumption that the measured features (pixel intensities)

$$\mathbf{x}=[\mathbf{x}_i|i=1\ldots n\}$$

are realizations of random vectors $\mathbf{X}^k$, chosen independently from K multivariate normally distributed populations representing the K principal land cover categories present in the image:[1]

$$\mathbf{X}^k \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \qquad k=1\ldots K \tag{1}$$

Here $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the expected value and covariance matrix of $\mathbf{X}^k$, respectively. We wish to maximize the *posteriori* probability $p(C|\mathbf{x})$ for observing the clusters given the data, where $C=[C_k]$ and C_k denotes the index set for the kth cluster. From Bayes' rule,

$$p(C|\mathbf{x})=\frac{p(\mathbf{x}|C)p(C)}{p(\mathbf{x})} \tag{2}$$

The quantity $p(\mathbf{x}|C)$ is the *likelihood* of observing the data x given the clustering C, $P(C)$ is the *prior probability* for C and $p(\mathbf{x})$ is a normalization independent of C.

[1] In our case the random vectors $\mathbf{X}^k$ are six-dimensional, corresponding to the six optical and infrared channels of the Landsat TM5 platform.

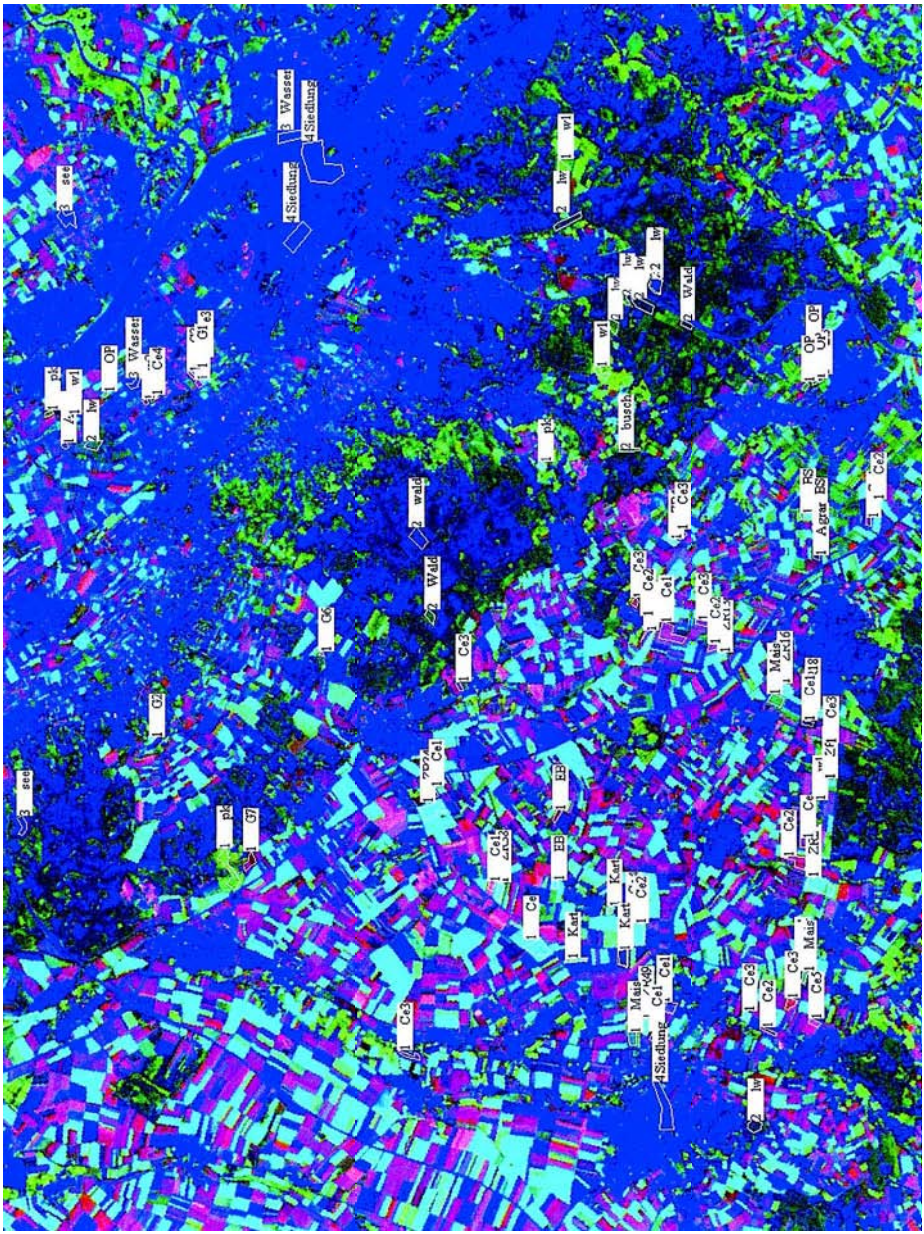

Figure 1. False colour composite of the Landsat image showing the training areas used for supervised classification (see §5). The image was constructed with the first three principal components, which account for >99% of the overall variance.

The likelihood of observing the data is product of the individual probability densities evaluated at the observed values, i.e.

$$p(\mathbf{x}|C)=\prod_{k=1}^{K}\prod_{i\in C_k} p(\mathbf{x}_i|C_k)$$
$$=\prod_{k=1}^{K}\prod_{i\in C_k}(2\pi)^{-N/2}|\mathbf{\Sigma}_k|^{-1/2}\exp(-\tfrac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_k))$$

Forming the product in this way is justified by the independence of the samples. The *log-likelihood* is given by (Fraley 1996)

$$L=\log p(\mathbf{x}|C)=\sum_{k=1}^{K}\sum_{i\in C_k}\left(-\frac{N}{2}\log(2\pi)-\frac{1}{2}\log|\mathbf{\Sigma}_k|-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\mathsf{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_k)\right)$$

With (2) we can therefore write

$$\log p(C|\mathbf{x})\propto L+\log p(C) \tag{3}$$

If all K classes exhibit identical covariance matrices according to

$$\mathbf{\Sigma}_k=\sigma^2\mathbf{I}, \qquad k=1\ldots K \tag{4}$$

where $\mathbf{I}$ is the identity matrix, then L is maximized when the expression

$$\sum_{k=1}^{K}\sum_{i\in C_k}(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\mathsf{T}}\left(\frac{1}{2\sigma^2}\mathbf{I}\right)(\mathbf{x}_i-\boldsymbol{\mu}_k)=\sum_{k=1}^{K}\sum_{i\in C_k}\frac{(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_i-\boldsymbol{\mu}_k)}{2\sigma^2}$$

is minimized. We are thus led to the *cost function*

$$E(C)=\sum_{k=1}^{K}\sum_{i\in C_k}\frac{(\mathbf{x}_i-\boldsymbol{\mu}_k)^{\mathsf{T}}(\mathbf{x}_i-\boldsymbol{\mu}_k)}{2\sigma^2}-\log p(C) \tag{5}$$

An optimal clustering C under these assumptions is achieved for $E(C)\rightarrow\min$.

Now we introduce a 'hard' class dependency in the form of a matrix $\mathbf{u}$ with elements given by

$$u_{ki}=\begin{cases}1 & \text{if } i\in C_k\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

The matrix $\mathbf{u}$ satisfies the conditions

$$\sum_{k=1}^{K}u_{ki}=1, \qquad i=1\ldots n \tag{7}$$

meaning that each sampled pixel $\mathbf{x}_i$, $i=1\ldots p$, belongs to precisely one class, and

$$\sum_{i=1}^{n}u_{ki}>0, \qquad k=1\ldots K \tag{8}$$

meaning that no class C_k, $k=1\ldots K$, is empty. The sum in (8) is the number n_k of pixels in the kth class. An unbiased estimate $\mathbf{m}_k$ of the expected value $\boldsymbol{\mu}_k$ for the kth cluster is therefore given by

$$\boldsymbol{\mu}_k\approx\mathbf{m}_k=\frac{1}{n_k}\sum_{i\in C_k}\mathbf{x}_i=\frac{\sum_{i=1}^{n}u_{ki}\mathbf{x}_i}{\sum_{i=1}^{n}u_{ki}}, \qquad k=1\ldots K \tag{9}$$

and an estimate $\mathbf{F}_k$ of the covariance matrix $\mathbf{\Sigma}_k$ by

$$\mathbf{\Sigma}_k \approx \mathbf{F}_k = \frac{\sum_{i=1}^{n} u_{ki}(\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^{\mathsf{T}}}{\sum_{i=1}^{n} u_{ki}}, \qquad k = 1 \ldots K \tag{10}$$

We can now write (5) in the form

$$E(C) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ki} \frac{(\mathbf{x}_i - \mathbf{m}_k)^{\mathsf{T}}(\mathbf{x}_i - \mathbf{m}_k)}{2\sigma^2} - \log p(C) \tag{11}$$

Finally, if we do not wish to include prior probabilities, we can simply say that all clustering configurations C are *a priori* equally likely. Then the last term in (11) is independent of C and we have, dropping the multiplicative constant $1/2\sigma^2$, the well-known *sum-of-squares* cost function

$$E(C) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ki}(\mathbf{x}_i - \mathbf{m}_k)^{\mathsf{T}}(\mathbf{x}_i - \mathbf{m}_k) \tag{12}$$

3. Five clustering algorithms

We begin with the popular K-means method, which is based on the simple cost function (12), and then consider an algorithm due to Palubinskas (1998) which uses cost function (11) and for which the number of clusters is determined automatically. Then we discuss a common version of hierarchic clustering and conclude with two variants of fuzzy clustering.

3.1. *K-means clustering*

The K-means clustering algorithm (KM) (sometimes referred to as *basic Isodata* (Duda and Hart 1973) or *migrating means* (Richards 1995)) is based on the cost function (12). After initialization of the cluster centres, the distance measure corresponding to a minimization of (12), namely

$$d(i, k) = (\mathbf{x}_i - \mathbf{m}_k)^{\mathsf{T}}(\mathbf{x}_i - \mathbf{m}_k)$$

is used to re-cluster the pixel vectors. Then (9) is used to recalculate the cluster centres. This procedure is iterated until the centres cease to change significantly.

3.2. *Extended K-means clustering*

Denote by $p_k = p(C_k)$ the prior probability for cluster k. The *entropy* S associated with this prior distribution is

$$S = -\sum_{k=1}^{K} p_k \log p_k \tag{13}$$

Distributions with high entropy are those for which the p_i are all similar, that is, the pixels are distributed evenly over all available clusters (Bishop 1995). Low entropy means that most of the data are concentrated in very few clusters. We choose a prior distribution $p(C)$ in (11) for which few clusters are more probable than many clusters, namely

$$p(C) \propto \exp(-\alpha_E S) = \exp\left(\alpha_E \sum_{k=1}^{K} p_k \log p_k\right)$$

where α_E is a parameter. The cost function (11) can then be written as

$$E(C)=\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ki}\frac{(\mathbf{x}_i-\mathbf{m}_k)^{\mathrm{T}}(\mathbf{x}_i-\mathbf{m}_k)}{2\sigma^2}-\alpha_E\sum_{k=1}^{K}p_k\log p_k \tag{14}$$

With

$$p_k\approx\frac{n_k}{n}=\frac{1}{n}\sum_{i=1}^{n}u_{ki} \tag{15}$$

this becomes

$$E(C)=\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ki}\left[\frac{(\mathbf{x}_i-\mathbf{m}_k)^{\mathrm{T}}(\mathbf{x}_i-\mathbf{m}_k)}{2\sigma^2}-\frac{\alpha_E}{n}\log p_k\right] \tag{16}$$

An estimate for the parameter α_E may be obtained as follows (Palubinskas 1998). From (14) and (15)

$$E(C)\approx\sum_{k=1}^{K}\left[\frac{n\sigma_k^2p_k}{2\sigma^2}-\alpha_Ep_k\log p_k\right]$$

Equating the likelihood and prior terms in this expression and taking $\sigma_k^2\approx\sigma^2$ and $p_k\approx 1/\tilde{K}$, where $\tilde{K}$ is some *a priori* expected number of clusters, gives

$$\alpha_E\approx-\frac{n}{2\log(1/\tilde{K})} \tag{17}$$

The parameter σ^2 in (16) can be estimated from the data.

The *extended K-means* (EKM) algorithm is as follows: first an initial configuration with a very large number of clusters K is chosen (for one-dimensional data this might conveniently be the 256 grey values that a pixel with 8-bit resolution can have) and initial values

$$\mathbf{m}_k=\frac{1}{n_k}\sum_{i=1}^{n}u_{ki}\mathbf{x}_k,\qquad p_k=\frac{n_k}{n} \tag{18}$$

are determined. Then the data are re-clustered according to the distance measure corresponding to a minimization of (16):

$$d(i,k)=\frac{(\mathbf{x}_i-\mathbf{m}_k)^{\mathrm{T}}(\mathbf{x}_i-\mathbf{m}_k)}{2\sigma^2}-\frac{\alpha_E}{n}\log p_k \tag{19}$$

The prior term tends to reduce the number of clusters and any class which has in the course of the algorithm $n_k=0$ is simply dropped from the calculation. (Condition (8) is thus relaxed.) Iteration of (18) and (19) continues until no significant changes in $\mathbf{m}_k$ occur.

The explicit choice of the number of clusters K is replaced by the necessity of choosing a value for the 'meta-parameter' α_E. This has the advantage of being able to use one parameter for a wide variety of images and letting the algorithm itself decide on the actual value of K in any given instance.

3.3. *Agglomerative hierarchical clustering*

The *agglomerative hierarchical clustering* (AHC) algorithm that we consider here is, as for K-means, based on the cost function (12), see for example Duda and Hart (1973). It begins by assigning each pixel in the dataset to its own class or cluster.

At this stage of course, the cost function $E(C)$, equation (12), is zero. We write $E(C)$ in the form

$$E(C)=\sum_{k=1}^{K} E_k \tag{20}$$

where E_k is given by

$$E_k=\sum_{i\in C_k}(\mathbf{x}_i-\mathbf{m}_k)^{\top}(\mathbf{x}_i-\mathbf{m}_k) \tag{21}$$

Every agglomeration of clusters to form a smaller number of clusters will increase $E(C)$. We therefore seek a prescription for choosing two clusters for combination that will increase $E(C)$ by the smallest amount possible.

Suppose clusters k with n_k members and ℓ with n_ℓ members are merged, $k<\ell$, and the new cluster is labelled k. Then

$$\mathbf{m}_k\rightarrow\frac{n_k\mathbf{m}_k+n_\ell\mathbf{m}_\ell}{n_k+n_\ell}=:\bar{\mathbf{m}}$$

Thus after the agglomeration, E_k changes to

$$E_k=\sum_{i\in C_k\cup C_l}(\mathbf{x}_i-\bar{\mathbf{m}})^{\top}(\mathbf{x}_i-\bar{\mathbf{m}})$$

and E_ℓ disappears. The net change in $E(C)$ is therefore

$$\sum_{i\in C_k\cup C_l}(\mathbf{x}_i-\bar{\mathbf{m}})^{\top}(\mathbf{x}_i-\bar{\mathbf{m}})-\sum_{i\in C_k}(\mathbf{x}_i-\mathbf{m}_k)^{\top}(\mathbf{x}_i-\mathbf{m}_k)-\sum_{i\in C_l}(\mathbf{x}_i-\mathbf{m}_\ell)^{\top}(\mathbf{x}_i-\mathbf{m}_\ell)$$

$$=\frac{n_kn_\ell}{n_k+n_\ell}(\mathbf{m}_k-\mathbf{m}_\ell)^{\top}(\mathbf{m}_k-\mathbf{m}_\ell) \tag{22}$$

The minimum increase in $E(C)$ is achieved by combining those two clusters k and ℓ which minimize the above expression. Given two alternative candidate cluster pairs with similar combined memberships n_k+n_ℓ and whose means have similar euclidean separations $\|\mathbf{m}_k-\mathbf{m}_\ell\|$, this prescription obviously favours combining that pair with the larger discrepancy between n_k and n_ℓ. Thus similar-sized clusters are preserved and smaller clusters are absorbed by larger ones. The algorithm terminates when the desired number of clusters has been reached or continues until a single cluster has been formed. Assuming that the data consist of $\tilde{K}$ compact and well separated clusters, the slope of $E(C)$ vs the number of clusters K should decrease (become more negative) for $K\leqslant\tilde{K}$.

3.4. *Fuzzy K-means clustering*

We write $E(C)$ in (12) in the equivalent form (Dunn 1973)

$$E(C)=\sum_{k=1}^{K}\sum_{i=1}^{n}u_{ki}^{q}(\mathbf{x}_i-\mathbf{m}_k)^{\top}(\mathbf{x}_i-\mathbf{m}_k),\qquad q>1 \tag{23}$$

and make the transition from 'hard' to 'fuzzy' clustering by replacing (6) by continuous variables

$$0<u_{ki}<1,\qquad k=1\ldots K,\qquad i=1\ldots n \tag{24}$$

but retaining requirements (7) and (8). The matrix $\mathbf{u}$ is now a *fuzzy class membership* matrix.

With i fixed, we seek values for the u_{ki} that solve the minimization problem

$$E_i = \sum_{k=1}^{K} u_{ki}^q (\mathbf{x}_i - \mathbf{m}_k)^\top (\mathbf{x}_i - \mathbf{m}_k) \to \min, \qquad i = 1 \ldots n$$

under conditions (7). By introducing the Lagrange function

$$L_i = E_i - \lambda \left(\sum_{k=1}^{K} u_{ki} - 1 \right)$$

we can equivalently solve the unconstrained problem $L_i \to \min$. Differentiating with respect to u_{ki},

$$\frac{\partial L_i}{\partial u_{ki}} = q(u_{ki})^{q-1} (\mathbf{x}_i - \mathbf{m}_k)^\top (\mathbf{x}_i - \mathbf{m}_k) - \lambda = 0, \qquad k = 1 \ldots K$$

from which we have

$$u_{ki} = \sqrt[q-1]{\frac{\lambda}{q}} \sqrt[q-1]{\frac{1}{(\mathbf{x}_i - \mathbf{m}_k)^\top (\mathbf{x}_i - \mathbf{m}_k)}} \tag{25}$$

The Lagrange multiplier λ is determined by

$$1 = \sum_{k=1}^{K} u_{ki} = \sqrt[q-1]{\frac{\lambda}{q}} \sum_{k=1}^{K} \sqrt[q-1]{\frac{1}{(\mathbf{x}_i - \mathbf{m}_k)^\top (\mathbf{x}_i - \mathbf{m}_k)}}$$

Substituting this into (25), we obtain finally

$$u_{ki} = \frac{\sqrt[q-1]{\dfrac{1}{(\mathbf{x}_i - \mathbf{m}_k)^\top (\mathbf{x}_i - \mathbf{m}_k)}}}{\displaystyle\sum_{k'=1}^{K} \sqrt[q-1]{\frac{1}{(\mathbf{x}_i - \mathbf{m}_{k'})^\top (\mathbf{x}_i - \mathbf{m}_{k'})}}}, \quad k = 1 \ldots K, \qquad i = 1 \ldots n \tag{26}$$

The parameter q determines the 'degree of fuzziness' and is usually chosen as $q = 2$. We have followed this practice here.

The *fuzzy K-means* (FKM) algorithm consists of a simple iteration of equations (9) and (26). The algorithm terminates when the cluster centres $\mathbf{m}_k$, or alternatively when the matrix elements u_{ki}, cease to change significantly. This algorithm should give similar results to the K-means algorithm, but one expects it to be less likely to become trapped in local minima of the cost function.

3.5. *Fuzzy maximum likelihood clustering*

The *fuzzy maximum likelihood* (FML) algorithm (Gath and Geva 1989) ignores the cost function (12) and simply replaces u_{ki} in (26) by the posterior probability $p(C_k|\mathbf{x}_i)$ of class C_k given the observation $\mathbf{x}_i$. That is, using Bayes' theorem,

$$u_{ki} \to p(C_k|\mathbf{x}_i) \sim p(\mathbf{x}_i|C_k) p(C_k)$$

Here $p(\mathbf{x}_i|C_k)$ is taken to be a multivariate normal distribution function with estimated mean $\mathbf{m}_k$ and estimated covariance matrix $\mathbf{F}_k$ given by (9) and (10), respectively.

Thus

$$u_{ki} \sim p(C_k) \frac{1}{\sqrt{|\mathbf{F}_k|}} \exp\{-\tfrac{1}{2}(\mathbf{x}_i - \mathbf{m}_k)^{\mathrm{T}} \mathbf{F}_k^{-1} (\mathbf{x}_i - \mathbf{m}_k)\} \tag{27}$$

One can use the current class membership to estimate $P(C_k)$ as p_k according to (15). The Gath–Geva algorithm is then an iteration of equations (9), (10), (15) and (27) with the same termination condition as for the fuzzy K-means algorithm. After each iteration the columns of $\mathbf{u}$ are normalized according to (7).

Because of the exponential distance dependence of the fuzzy membership in (27), the algorithm is very sensitive to initialization conditions, and can even become unstable. To avoid this problem in our implementation, we first obtain initial values for the $\mathbf{m}_k$ and for $\mathbf{u}$ by preceding the calculation with the fuzzy K-means algorithm.

Since the cost function $E(C)$ is no longer relevant, we choose with (Gath and Geva 1989) the *fuzzy hypervolume* (FHV), defined as

$$\mathrm{FHV} = \sum_{k=1}^{K} \sqrt{|\mathbf{F}_k|} \tag{28}$$

as 'goodness' criterion. This quantity is proportional to the volume in feature space occupied by the ellipsoidal clusters generated by the algorithm. Assuming that the data consist of $\tilde{K}$ well-separated clusters of approximately multivariate normally distributed pixels, FHV should exhibit a minimum at $K = \tilde{K}$.

4. Number of clusters

The *a priori* choice of an appropriate number of clusters presents a major problem in the analysis of multispectral imagery. Even in the case of the extended K-means algorithm of §3.1 the entropy parameter α_E must be chosen somehow, for example via (17). We illustrate the difficulty by considering the case of the FML algorithm together with the fuzzy hypervolume (28).

4.1. *Simulated data*

Figure 2 shows clustering results for simulated two-dimensional data. Ten bivariate normally distributed clusters with random means and covariance matrices were generated with 200 members in each cluster. The FML algorithm with initially 20 or 25 class centres was then used to cluster the data. After convergence, the fuzzy hypervolume was determined. Then the cluster centre with the fewest members was discarded and the algorithm restarted, maintaining the previously determined centres and fuzzy membership matrix. This was repeated until two clusters remained. The fuzzy hypervolumes are shown in the upper two inserts of the figure. The clustered data with eight cluster centres are shown in the lower left insert, and compared with a similar clustering using the FKM algorithm, lower right.

The fuzzy hypervolume shows a reproducible global minimum at around six to eight clusters, a rather good result considering the degree of overlap of the simulated data. The FML algorithm also does well in detecting elongated clusters as opposed to FKM clustering, as can be seen in the lower two inserts.

4.2. *Real data*

A similar experiment was carried out using 2000 three-dimensional pixel vectors (the first three principal components) sampled from the Landsat scene of figure 1. The results are shown in figure 3.

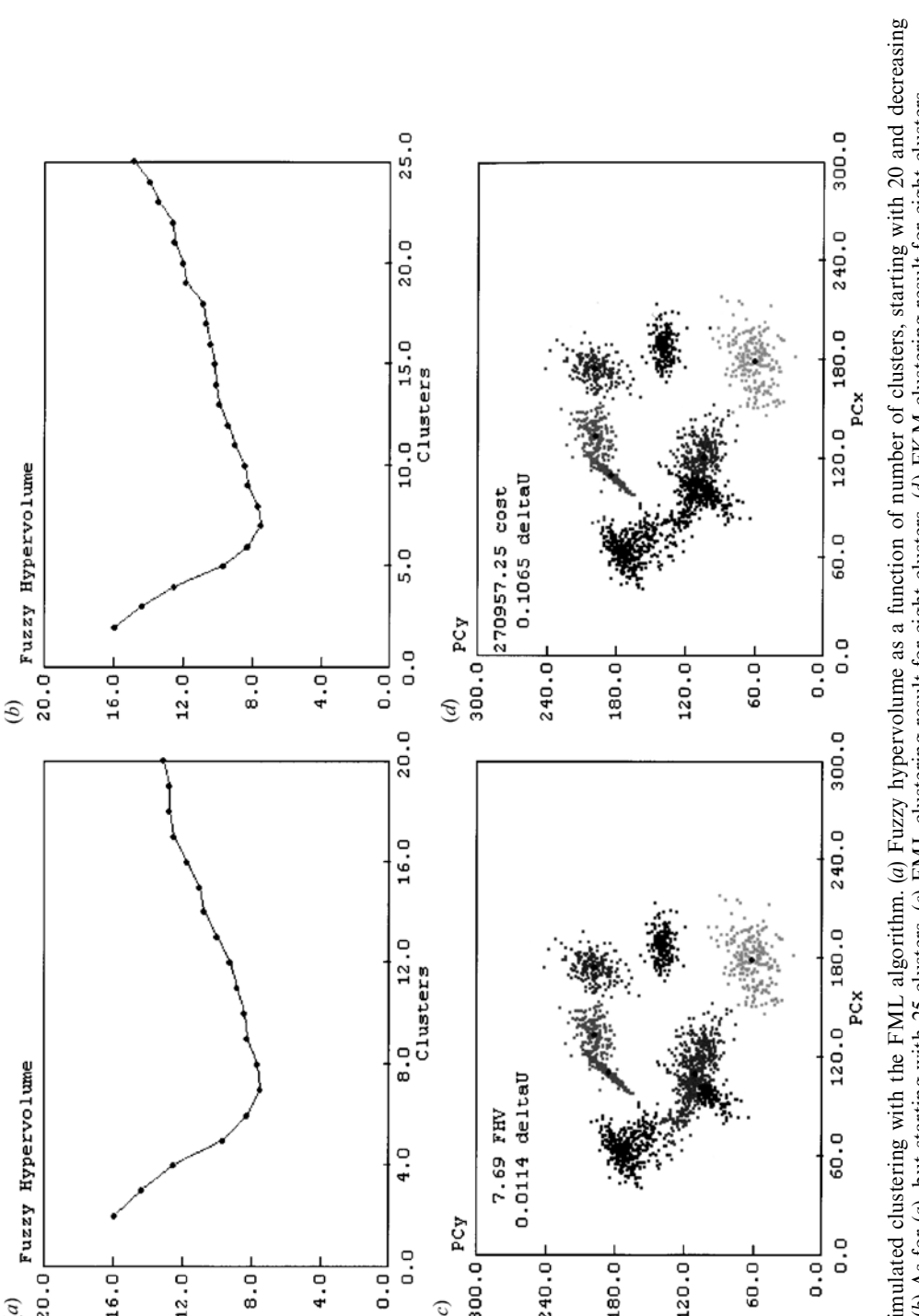

Figure 2. Simulated clustering with the FML algorithm. (*a*) Fuzzy hypervolume as a function of number of clusters, starting with 20 and decreasing to 2. (*b*) As for (*a*), but starting with 25 clusters. (*c*) FML clustering result for eight clusters. (*d*) FKM clustering result for eight clusters.

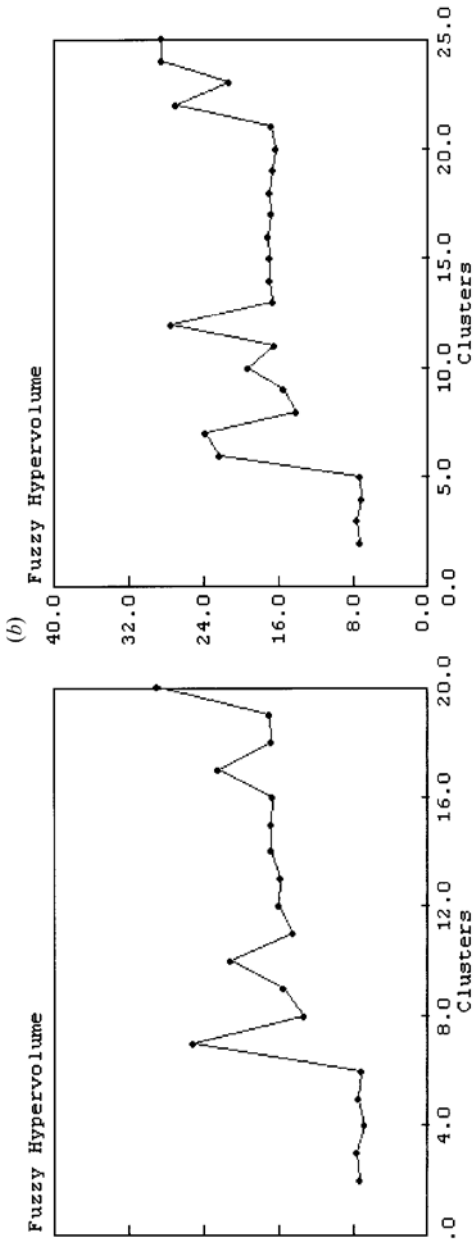

Figure 3. Clustering of Landsat data with the FML algorithm. (*a*) Fuzzy hypervolume as a function of number of clusters, starting with 20 and decreasing to 2. (*b*) As for (*a*), but starting with 25 clusters. (*c*) FML clustering result for eight clusters. (*d*) FKM clustering result for eight clusters.

There is obviously no global minimum, at least not at a useful number of clusters, nor do the results appear to be reproducible. Although scaling effects might be expected in the data, i.e. clusters of clusters reflecting a hierarchical organization of the landscape, the spikes in the fuzzy hypervolume plots may merely be artefacts of the implementation of the algorithm. A stopping criterion

$$\max|\Delta u_{ki}| \leqslant 0.01$$

where the u_{ki} are the current fuzzy class memberships, was used throughout, although it was observed that this quantity oscillates during the course of the iterations. In the case of strongly overlapping clusters (Gath and Geva 1989) suggest alternatively using a *partition density* which they define as the ratio of central members of a cluster to its fuzzy hypervolume. Trials with this criterion produced plots similar to those of figure 3, however. On the basis of these results and in view of a similar lack of criteria for the other algorithms, it was concluded that an *ad hoc* choice of an appropriate number of clusters would have to be made in order to carry out a performance comparison of the five unsupervised methods. This choice is discussed in the next section.

5. Results

The original multispectral data were compressed via a principal components analysis in order to reduce computation time. The eigenvalues (variances of the principal components) are shown in table 1. The supervised and unsupervised classifications described in the sequel were all performed on the first three principal components, which account for more than 99% of the spectral variance in the original six Landsat TM5 channels.

5.1. *Supervised classification*

A supervised classification of the reference scene was carried out on the basis of intensive on-site observation of 91 training areas. These are shown in figure 1. They were characteristic of all land-use categories and were distributed across the imaged area. In all, 5000 pixels were selected randomly from the training areas for supervised classification, along with 2500 additional pixels for subsequent, unbiased evaluation of classification accuracy.

In order to make feasible a comparison with unsupervised classification, the training data were grouped into four 'super' classes: farmland/fields, forest, water and densely built-up areas. The extreme variability of agricultural land use in the reference scene precluded a more detailed comparison. Nevertheless it was felt that the ability to discriminate these four main classes would serve as a good yardstick for the overall usefulness of the clustering methods. Table 2 shows the Jeffries–Matusita separation (Richards 1995) of the training pixels chosen from the four classes. A value of two implies perfect separation, zero implies identical overlap. The classes are seen to be extremely well separated and we expect to be able to do well in classifying the entire scene.

Table 1. Principal components analysis.

Principal component	1	2	3	4	5	6
Normalized variance	0.630	0.326	0.035	0.006	0.003	0.001

Table 2. Separability of training pixels.

Class	Forest	Water	Built-up
Farmland	1.971	2.000	1.970
Forest		1.993	1.990
Water			1.967

A two-layer feed-forward neural network classifier, consisting of three inputs (the three dimensions of the input data), four outputs (the four main land-use categories) and 12 neurons in the first (hidden) layer was trained on the data. The output neurons used the softmax activation function, and the corresponding cross-entropy cost function was minimized with the scaled conjugate gradiant algorithm as described for example in Bishop (1995) and Canty (1999).

Table 3 shows the classification accuracy obtained in the form of a map user's confusion matrix, calculated with the 2500 hold-out pixel vectors. Errors are one standard deviation, assuming that the binomially distributed misclassifications can be approximated by a normal distribution. The corresponding Kappa coefficient of agreement (Cohen 1960) is 0.96. By comparison, a classification with the standard maximum likelihood method achieved a Kappa value of 0.85. This difference is attributable to the complicated structure of the training data. In particular the class *farmland* cannot be well-approximated by a multivariate normal distribution. The classified image obtained with the neural network classifier is shown in figure 4.

5.2. *Clustering*

The number of clusters used for comparison of the unsupervised classification methods was chosen to be 12. This was arrived at by experimentation and represents essentially the minimum number of clusters required to discriminate built-up areas from the other land cover categories. It was a rather subjective decision, since the category *farmland*, due to its high spectral complexity, was represented in all final classified images by at least eight clusters. However, with this choice, a single cluster could be unambiguously associated with built-up areas in all five cases. Furthermore the interpretation of the classified image becomes quite difficult if the number of clusters substantially exceeds this number. The EKM method was treated in a special way, in that the entropy parameter α_E was chosen from (17) with $\tilde{K} = 12$. The actual number of clusters found by the algorithm varied between 11 and 14. This was considered to be a fair way to proceed, since the main advantage of the EKM method is that it can determine the final cluster number from the data (Palubinskas 1998).

Table 3. Confusion matrix (map user's accuracy in %).

Class	Farmland	Forest	Water	Built-up
Farmland	99.2 ± 0.2	0.7	0.0	0.1
Forest	2.7	97.3 ± 0.7	0.0	0.0
Water	0.6	0.0	99.4 ± 0.6	0.0
Built-up	9.4	0.0	1.3	89.3 ± 2.5

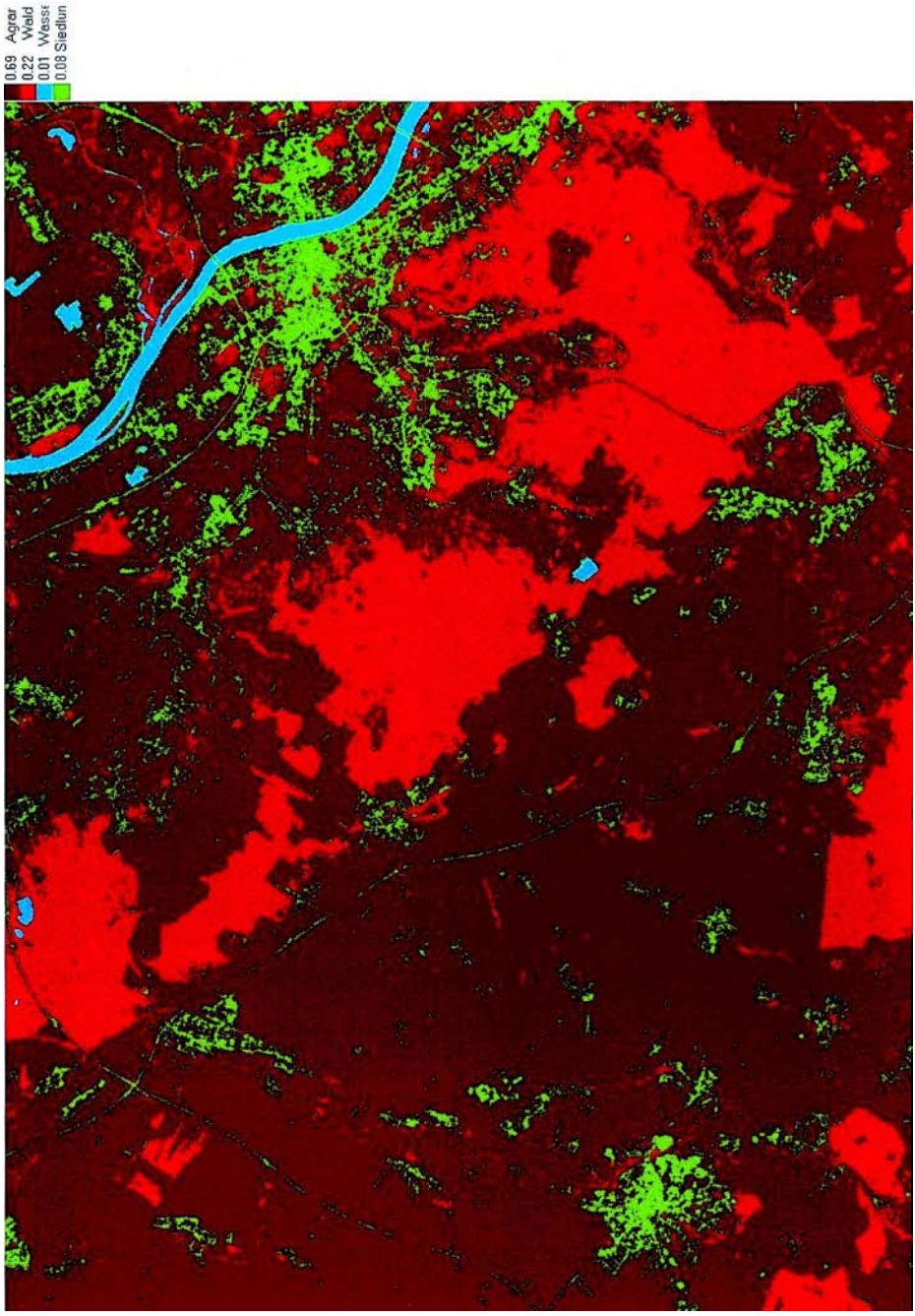

Figure 4. Supervised classification of the reference scene with a two-layer, feed-forward neural network: farmland (dark brown), forest (brown), water (blue), built-up (green).

The clustering algorithms were run on the first three principal components of 5000 representative pixel vectors sampled randomly from the image. After convergence the entire scene was classified. With the exception of AHC, each algorithm was run five times from different starting cluster centres in order to ascertain how sensitive the results are to initial conditions. (The initial conditions for AHC are of course unique.) Agreement is within one standard deviation, except for the EKM algorithm, as mentioned below. Figure 5 shows as an example a classification of the reference scene obtained for the FML algorithm.

By examination, each of the clusters determined was then mapped to one of the four main land-cover classes and colour coded in the same way as for the supervised classification of figure 4. Typical many-to-one correspondences used are given in the second and third columns of table 4. Figure 6 shows a typical result obtained, again for the FML algorithm. This figure should be compared directly with figure 4.

Finally, each clustered image was compared pixel-by-pixel to the supervised classification and the number of differences determined. The percent disagreement, averaged over the five trials where appropriate, is shown in the last column of table 4. These values are reproducible to within 1% (1 standard deviation), with the exception of the EKM algorithm. Here the result is strongly correlated with the number of clusters actually found, with the best result (11.6%) obtained for 14 clusters. Figure 7 shows the spatial distribution of disagreements in the form of a collage of five difference images. Here not only the magnitude of the discrepancies, but also their degree of inhomogeneity is apparent.

6. Conclusions

Fuzzy K-means (FKM) clustering reproduces the supervised classification of the reference scene most accurately, as can be seen from table 4, whereas the agglomerative hiearchic clustering (AHC) method reproduces the ground data most uniformly, as can be seen from figure 7. The K-means (KM) and FKM methods tend to form clusters of similar size and shape, and thus they both fail to recognize the small, compact cluster corresponding to water surfaces. This is clearly not a serious disadvantage when using the algorithms as an aid to determining training areas. (Since water surface accounts for about 1% of the scene area, the difference values for KM and FKM in column 4 of table 4 could in fact be reduced by that amount.) In the case of the FKM algorithm this seems in fact to be a necessary price to pay, as the rest of the image corresponds quite well to ground data. The extended K-means (EKM) algorithm drastically overestimates the built-up areas when the number of clusters found is 12. This is also true of the fuzzy maximum likelihood (FML) method, although to a lesser extent. The FML algorithm, although it can find clusters of widely varying size and elongation and successfully classifies water surfaces, does significantly worse than FKM on the whole.

These results can be understood better by examining the forms of the clusters generated by the respective algorithms. The three relatively compact clusters, namely forest, water and built-up areas, are shown in figure 8, projected onto the 1-2 principal axis plane. The KM algorithm confuses all three categories badly. The FML algorithm attributes forested land cover to a single, oblong cluster but underestimates its size relative to the EKM, FKM and AHC. These yield two clusters that can be clearly associated with forested areas and seem to do better in classifying them. The EKM method finds a quite large cluster for built-up areas, but obviously mixes in too large a proportion of farmland.

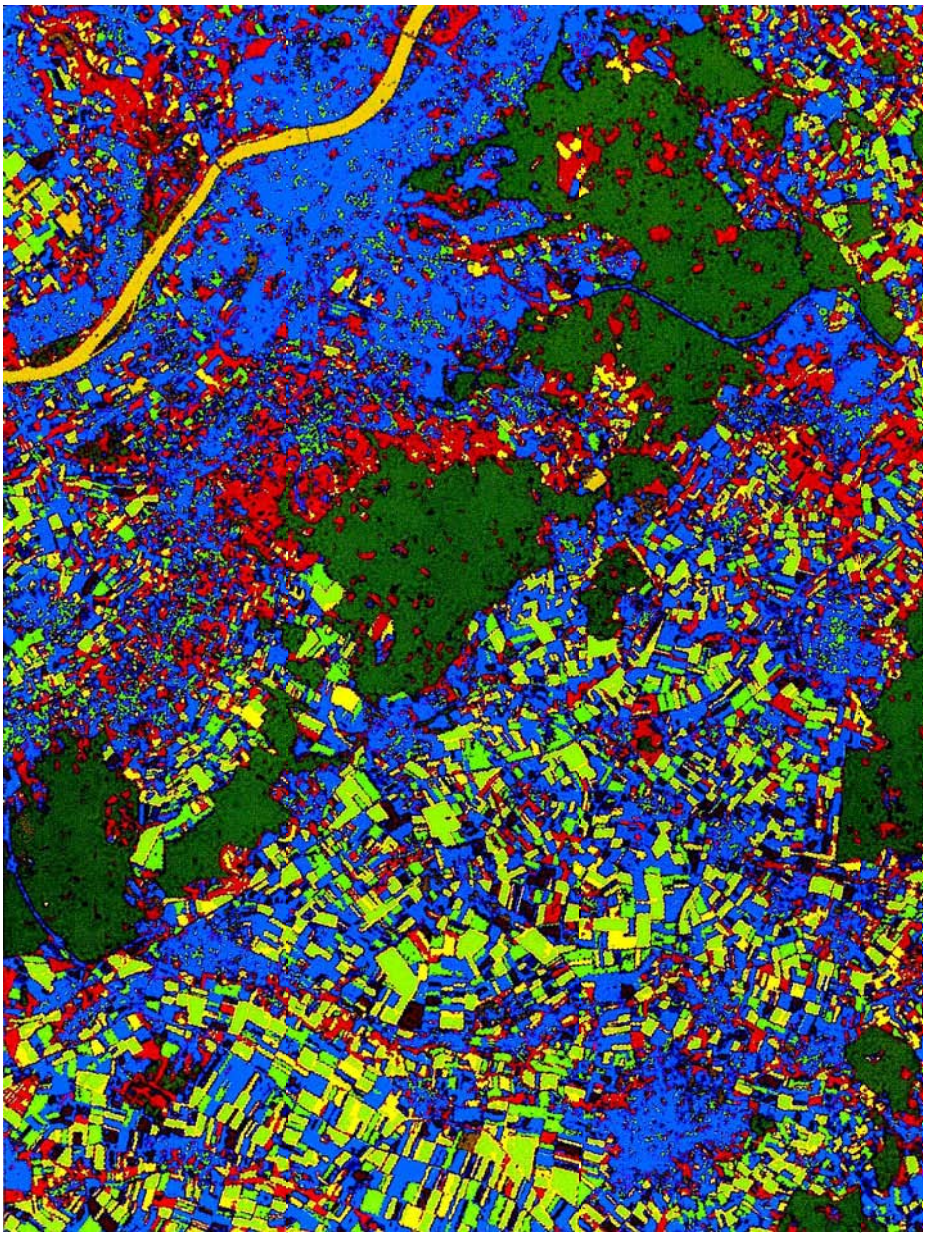

Figure 5. Unsupervised classification of the reference scene with the fuzzy maximum likelihood (FML) algorithm and 12 cluster centres.

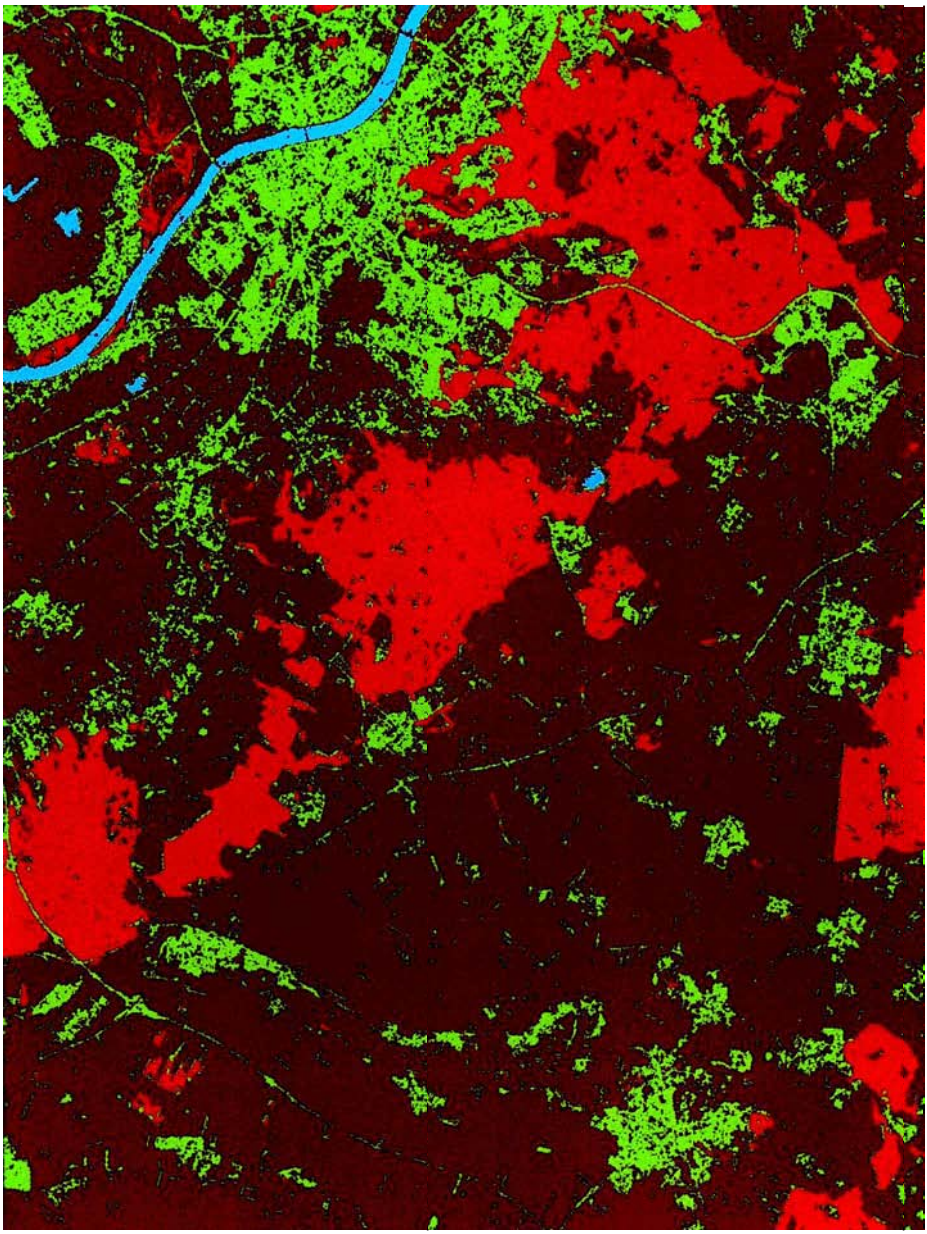

Figure 6. As figure 5 with clusters mapped to the four principal land cover categories of figure 4, see table 4.

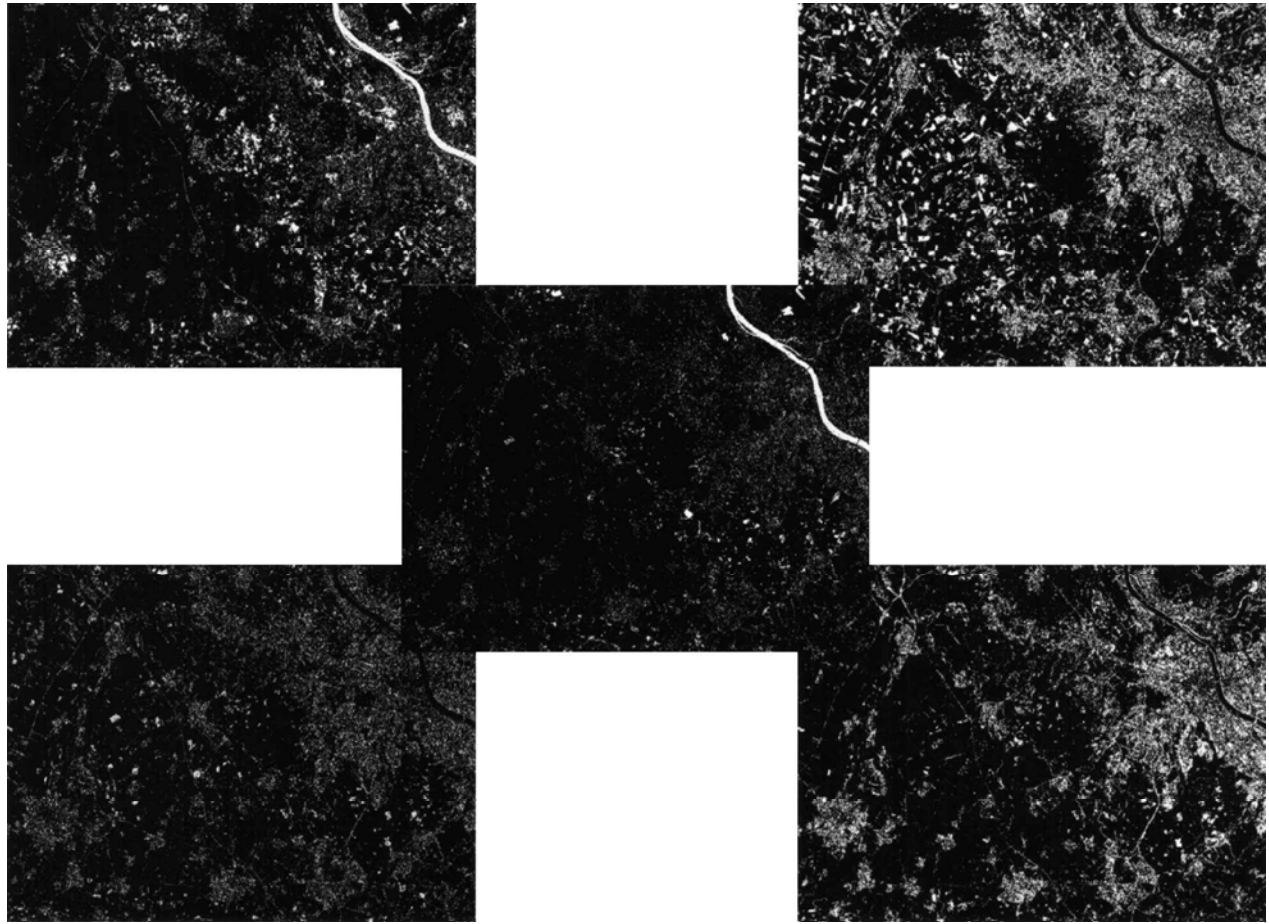

Figure 7. Comparison of unsupervised classification algorithms. Top left KM, top right EKM, lower left AHC, lower right FML, centre FKM. Differences with respect to the supervised classification result of figure 4 are shown as bright pixels.

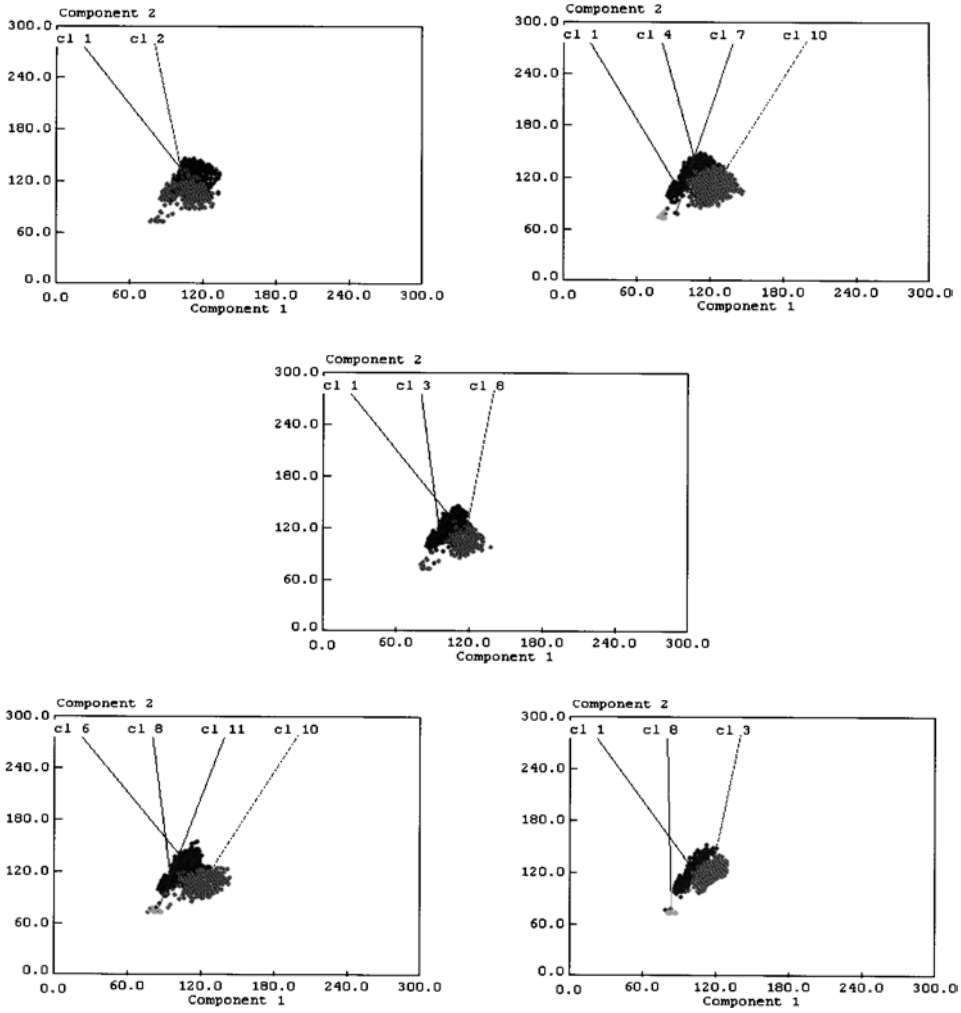

Figure 8. Forest, water and built-up clusters in the plane of the first and second principal axes. Top left KM, top right EKM, lower left AHC, lower right FML, centre FKM.

On the whole, the fuzzy K-means clustering algorithm would appear to correspond most closely with ground data in this example. It is therefore to be favoured as an aid for ascertaining ground cover categories and for designing training areas for supervised classification in scenes of similar complexity. Agglomerative hierarchical clustering also performs satisfactorily, and might be preferred on the basis of its ability to discriminate water surfaces.

Finally, it should be emphasized that, although our conclusions are essentially based on the quantitative disagreement values in table 4, these values may be influenced by the initial—and essentially subjective—choice of 12 clusters for comparing the five algorithms, and to their assignment to the four land cover categories.

Acknowledgement

The authors would like to express their appreciation to Gintuatas Palubinskas for his comments and suggestions on the material in §2 and 3.

Table 4. Comparison of clustering algorithms.

Algorithm	Cluster	Class	Disagreement (%)
K-means (KM)	3, 4, 5, 6, 7, 8, 9, 11, 12 1 – 2	1 (farmland) 2 (forest) 3 (water) 4 (built-up)	11.2
Extended K-means (EKM)	2, 3, 5, 6, 8, 9, 11, 12, 13 1, 4 7 10	1 (farmland) 2 (forest) 3 (water) 4 (built-up)	11.6–26.9
Agglomerative hierarchical (AHC)	1, 2, 3, 4, 5, 7, 9, 12 6, 8 11 10	1 (farmland) 2 (forest) 3 (water) 4 (built-up)	9.4
Fuzzy K-means (FKM)	2, 4, 5, 6, 7, 9, 10, 11, 12 1, 3 – 8	1 (farmland) 2 (forest) 3 (water) 4 (built-up)	7.2
Fuzzy maximum likelihood (FML)	1, 2, 3, 5, 6, 7, 9, 10, 12 4 11 8	1 (farmland) 2 (forest) 3 (water) 4 (built-up)	13.0

References

Bishop, C. M., 1995, *Neural Networks and Pattern Recognition* (Oxford: Oxford University Press).

Canty, M. J., 1999, *Fernerkundung mit Neuronalen Netzen* (Renningen: expert-Verlag).

Chuvieco, E., and Congalton, R. G., 1988, Using cluster analysis to improve the selection of training statistics in classifying remotely sensed data. *Photogrammetric Engineering and Remote Sensing*, **54**, 1275–1281.

Cohen, J., 1960, A coefficient of agreement of nominal scales. *Educational and Psychological Measurement*, **10**, 37–46.

Duda, R. O., and Hart, P. E., 1973, *Pattern Classification and Scene Analysis* (New York: Wiley).

Dunn, J. C., 1973, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, **3**, 32–57.

Fraley, C., 1996, Algorithms for model-based gaussian hierarchical clustering. Technical Report No. 311, Department of Statistics, University of Washington, Seattle.

Gath, I., and Geva, A. B., 1989, Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAM1-1**, 773–781.

Palubinskas, G., 1998, K-means clustering algorithm using the entropy. SPIE *European Symposium on Remote Sensing (EUROPTO'98), Conference on Image and Signal Processing for Remote Sensing), September 21–24, 1998, Barcelona, Spain*, edited by S. B. Serpico, vol. 3500 (Bellingham: SPIE), pp. 63–71.

Richards, J. A., 1995, *Remote Sensing Digital Image Analysis,* 2nd edn (Berlin: Springer).

Schowengerdt, R. A., 1997, *Remote Sensing: Models and Methods for Image Processing*, 2nd edn (London: Academic Press).