8 Image Classification

Image classification belongs to a very active field in computing research, that of *pattern recognition*. Image pixels can be classified either by their multivariable statistical properties, such as the case of multi-spectral classification (clustering), or by segmentation based on both statistics and spatial relationships with neighbouring pixels. In this chapter, we will look at multi-variable statistical classification techniques for image data.

8.1 Approaches of statistical classification

Generally, statistical classification can be catalogued into two major branches: *unsupervised and supervised classifications.*

8.1.1 Unsupervised classification

This is entirely based on the statistics of the image data distribution, and is often called *clustering*. The process is automatically optimized according to cluster statistics without the use of any knowledge-based control (i.e. ground truth). The method is therefore objective and entirely data driven. It is particularly suited to images of targets or areas where there is no ground truth knowledge or where such information is not available, such as in the case of planetary images. Even for a well-mapped area, unsupervised classification may reveal some spectral features which were not apparent beforehand. The result of an unsupervised classification is an image of statistical clusters, where the thematic contents of the clusters are not known. Ultimately, such a classification image still needs interpretation based on some knowledge of ground truth.

8.1.2 Supervised classification

This is based on the statistics of *training areas* representing different ground objects selected subjectively by users on the basis of their own knowledge or experience. The classification is controlled by users' knowledge but, on the other hand, is constrained and may even be biased by their subjective view. The classification can therefore be misguided by inappropriate or inaccurate training area information and/or incomplete user knowledge.

Realizing the limitations of both major classification methods, a *hybrid classification* approach has been introduced. In the hybrid classification of a multi-spectral image, firstly an unsupervised classification is performed, then the result is interpreted using ground truth knowledge and, finally, the original image is reclassified using a supervised classification with the aid of the statistics of the

Essential Image Processing and GIS for Remote Sensing By Jian Guo Liu and Philippa J. Mason

unsupervised classification as training knowledge. This method utilizes unsupervised classification in combination with ground truth knowledge as a comprehensive training procedure and therefore provides more objective and reliable results.

8.1.3 Classification processing and implementation

A classification may be completed in one step, as a *single pass classification*, or in an iterative optimization procedure referred to as an *iterative classification*. The single pass method is the normal case for supervised classification while the iterative classification represents the typical approach to unsupervised classification (clustering). The iterative method can also be incorporated into a supervised classification algorithm.

Most image processing software packages perform image classification in the image domain by *image scanning classification*. This approach can classify very large image datasets with many spectral bands and very high quantization levels, with very low demands on computing resources (e.g. RAM) but it cannot accommodate sophisticated classifiers (decision rules). Image classification can also be performed in feature space by *feature space partition*. In this case, sophisticated classifiers incorporating data distribution statistics can be applied but the approach demands a great deal of computer memory to cope with the high data dimensionality and quantization. This problem is being overcome by increasingly powerful computing hardware and dynamic memory management in programming.

8.1.4 Summary of classification approaches

These are as follows:

- Unsupervised classification
- Supervised classification
- Hybrid classification
- Single pass classification
- Iterative classification
- Image scanning classification
- Feature space partition.

8.2 Unsupervised classification (iterative clustering)

8.2.1 Iterative clustering algorithms

For convenience of description, let $\mathbf{X}$ be a n-dimensional feature space of n variables ($\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$), Y_i be an object of an object set Y (an image) defined by measurements of the n variables (e.g. DNs of n spectral bands), $Y_i = (y_{i1}, y_{i2}, \ldots, y_{in})$, $i = 1, 2, \ldots, N$. N is the total number of objects in Y or the total number of pixels in an image. As shown in Figure 8.1, in the feature space $\mathbf{X}$, the object Y_i is represented by an observation vector, that is a data point $\mathbf{X}_j \in \mathbf{X}$ at the coordinates (x_{j1}, x_{j2}, $\ldots$, x_{jn}), $j = 1, 2, \ldots, M$. M is the total number of data points representing N objects. If $\mathbf{X}$ is a Euclidean space, then $x_{jh} \sim y_{ih}$, $h = 1, 2, \ldots, n$. Obviously, a data point $\mathbf{X}_j$ in the feature space $\mathbf{X}$ can be shared by more than one image pixel Y_i and therefore $M \leq N$.

The goal of the clustering process is to identify the objects of set Y in m classes. This is equivalent to the partition of the relevant data points in feature space $\mathbf{X}$ into m spatial clusters, $\omega_1, \omega_2, \ldots, \omega_m$. Generally, there are two principal iterative clustering algorithms labelled α and β (Diday and Simon, 1976), as follows:

Algorithm α

1. *Initialization*
   Let m elements $Y_q \in$ Y, chosen at random or by a selection scheme, be the 'representation'

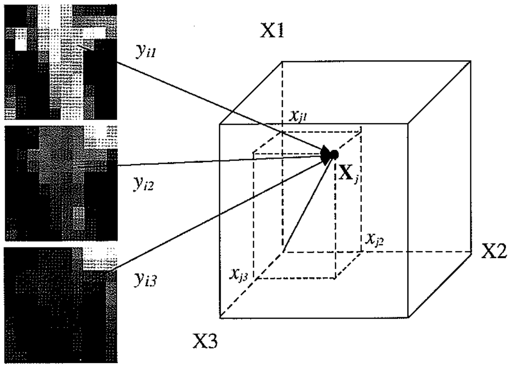

Figure 8.1 A 3D illustration of the relationship between a feature space point $\mathbf{X}_j$ and a multi-spectral image pixel $Y_i = (y_{i1}, y_{i2}, y_{i3})$

of m clusters denoted as $\omega_1, \omega_2, \ldots, \omega_k, \ldots, \omega_m$.

2. *Clustering*
   For all i, assign any element Y_i $(Y_i \in Y)$ to a cluster ω_k, if the dissimilarity measurement $\delta(Y_i, \omega_k)$ is minimal.
3. *Update statistical representation*
   For all k, new statistics of cluster ω_k are computed as the renewed representation of the cluster ω_k.
4. *Stability*
   If no ω_k has changed above the given criteria then stop, else go to 2.

Algorithm β

1. As in step 1 of algorithm α.
2. One element Y_i $(Y_i \in Y)$ is assigned to cluster ω_k, if $\delta(Y_i, \omega_k)$ is minimal.
3. A new representation of ω_k is computed from all the elements of cluster ω_k, including the last element.
4. If all elements Y_i $(Y_i \in Y)$ have been assigned to a cluster then stop, else go to step 2.

Algorithm α may not necessarily converge if the criterion for terminating the iteration is too tight. Algorithm β ends when the last pixel is reached. Algorithm α is more commonly used for image classification because of its self-optimization mechanism and processing efficiency. Cluster splitting and merging functions can be added to algorithm α after step 4, which allows the algorithm to operate more closely with the true data distribution and to reach more optimized convergence. During the progress of the clustering iteration, the initial cluster centres move towards the true data cluster centres via the updating of their statistical representations at the end of each iteration. The only user control on clustering is the initial parameter setting, such as number and position of the starting centres of clusters, iteration times or termination criteria, maximum and minimum number and size of clusters, and so on. The initial setting will affect the final result. In this sense, the clustering iteration mechanism can only ensure local optimization, the optimal partition of clusters for the given initial parameter setting, but not the global optimization, because the initial parameter setting cannot be optimal for the best possible clustering result.

For most image processing packages, image clustering using either of the two algorithms is executed on an object set Y, that is the image. The processing is on a pixel-by-pixel basis by scanning the image but, with advances in computing power, the very demanding feature space partition clustering in the feature space $\mathbf{X}$ becomes feasible.

One of the most popular clustering algorithms for image classification, the ISODATA algorithm (Ball, 1965; Ball and Hall, 1967), is a particular case of algorithm α in which the dissimilarity measure $\delta(Y_i, \omega_k)$ in step 2 is the square Euclidean distance. The assumption underlying this simple and efficient technique is that all the clusters have equal variance and population. This assumption is generally untrue in image classification, and as a result classification accuracy may be low. To improve ISODATA, more sophisticated measures of dissimilarity, such as maximum likelihood estimation and population weighted measurements, have been introduced. For all these different decision rules, within the ISODATA frame, the processing is performed by image scanning.

8.2.2 Feature space iterative clustering

As mentioned earlier, image classification can be performed by image scanning as well as by feature space partition. Most multi-variable statistical classification algorithms can be realized by either approach but, for more advanced decision rules, such as optimal multiple point reassignment (OMPR), which will be introduced later, feature space partition is the only feasible method because all pixels sharing the same DN values in each image band must be considered simultaneously. Here we introduce a three-dimensional feature space iterative clustering method, the 3D-FSIC method, an algorithm which can be easily extended to further dimensions.

Three-dimensional feature space iterative clustering (3D-FSIC)

Step 1. Create a 3D Scattergram of the Input Image
Read the input image, Y, pixel by pixel and record the pixel frequencies in a 3D scattergram, that is a 3D array (Figure 8.2a)

$$G(d1 \times d2 \times d3)$$

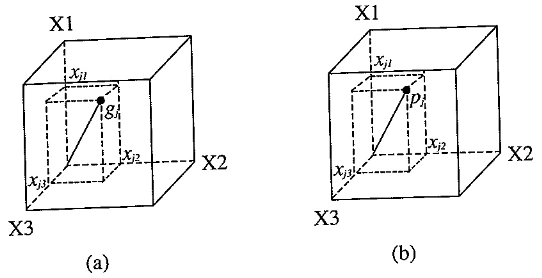

Figure 8.2 (a) A 3D array G of a scattergram; and (b) a 3D array P of a feature space partition

where $d1$, $d2$ and $d3$ are the array sizes in the three dimensions or the maximum DN values of the three bands of the image Y.

The value of any element g_j in G indicates how many pixels share the point X_j at the coordinates (x_{j1}, x_{j2}, x_{j3}) in the 3D feature space $\mathbf{X}$, or the number of pixels with the same DN values as pixel Y_i in the image Y, where $y_{ih} \sim x_{jh}$, $h = 1, 2, 3$.

Step 2. Initialization Select m points in the 3D feature space $\mathbf{X}$ as the 'seeds' of m clusters and call them ω_k, $k = 1, 2, \ldots, m$. The choice could be made at random or via an automatic seed selection technique.

Step 3. Feature space clustering For all j, assign any point $\mathbf{X}_j$ $(\mathbf{X}_j \in \mathbf{X}, j = 1, 2, \ldots, N)$ to cluster ω_k if the dissimilarity $\delta(\mathbf{X}_j, \omega_k)$ is minimal. Thus all the pixels sharing the point X_j are assigned to cluster ω_k simultaneously. The size of cluster ω_k, N_k, increases by the value g_j while, if it is a reassignment, the size of the cluster to which $\mathbf{X}_j$ was formerly assigned decreases by the same value. The cluster sequential number k of point $\mathbf{X}_j$ is recorded by a 3D feature space partition array P $(d1 \times d2 \times d3)$ in the element p_j at coordinates (x_{j1}, x_{j2}, x_{j3}) (Figure 8.2b).

Step 4. Update the statistical representation of each cluster For all k $(k = 1, 2, \ldots, m)$, statistical parameters, such as mean vector $\boldsymbol{\mu}_k$, covariance matrix $\sum_k$ and so on are calculated. These parameters comprise the new representation of the cluster ω_k.

Step 5. Stability For all k $(k = 1, 2, \ldots, m)$, if the maximum spatial migration of the mean vector $\boldsymbol{\mu}_k$ (the kernel of the cluster) is less than a user-controlled criterion, go to step 7, else go to step 6.

Step 6. Cluster splitting and merging Split the overlarge and elongate clusters and merge clusters which are too small and/or too close to each other, according to user-controlled criteria; then update the statistical representations of the new clusters. Go to step 3.

Step 7. Transfer the Clustering Result from Feature Space to an Image Read the input image Y, pixel by pixel. For all i, assign a pixel Y_i $(Y_i \in Y)$ to cluster ω_k if its relevant data point $\mathbf{X}_j$ in feature space $\mathbf{X}$ is assigned to this cluster, according to the record in the feature space partition array P, that is

$$Y_i \rightarrow \omega_k \text{ if } P_j = k$$

where P_j is at coordinates (x_{j1}, x_{j2}, x_{j3}) in P and $y_{ih} \sim x_{jh}$, $h = 1, 2, 3$.

Then assign the class number to the corresponding pixel in the output classification image Y_{class}.

8.2.3 Seed selection

The initial kernels (seeds) for unsupervised classification can be made randomly, evenly or by particular methods. Here we introduce an automatic seed selection technique (ASST) for 3D-FSIC.

In the 3D scattergram of the three images for classification, data will exhibit peaks (the points of high frequency) at the locations of spectral clusters. It is thus sensible to use these points as the initial kernels of clusters to start iterative clustering. Such a peak point has two properties:

- Higher frequency than all its neighbouring points in the feature space.
- Relatively high frequency in the 3D scattergram.

These two properties are to be used to locate peak points. It is important to bear in mind that the multi-spectral image data and scattergram are discrete and that the DN value increment of an image may not necessarily be unity, especially after contrast enhancement. For instance, when an image of 7 bit

DN range [0, 127] is linearly stretched to 8 bit DN range [0, 255], the increment of DN values becomes 2 instead of 1. In this case, any non-zero frequency DN level in the original image will have two adjacent zero-frequency DN levels in the stretched image (Figure 8.3), appearing as a pseudo peak caused by data discontinuity. With these considerations in mind, ASST is composed of the following operations:

1. Locate and rank the first N points of highest frequency from the 3D scattergram to form a sequential set $\mathbf{X}_c$. N can be decided by setting a criterion frequency on the basis of experience and experimentation. For an 8 bit full-scene TM image, 1024 is suggested. This operation will prevent the selection of isolated low-frequency points (often representing noise) as seeds and thus satisfy the second property.
2. The first element in the set $\mathbf{X}_c$ must be nominated as a seed because it cannot be surrounded by any elements of a higher frequency. Then, for the second element of $\mathbf{X}_c$, check if the first element is in its given neighbourhood range (the neighbourhood is used to avoid pseudo peaks in the image with DN increment greater than 1) and, if not, the second element is also selected as a seed. In general, for any element $\mathbf{X}_j$ in $\mathbf{X}_c$, check the coordinates of those elements ranked with higher frequency; $\mathbf{X}_j$ is selected as a seed if none of the higher frequency elements are within its neighbourhood. This operation makes the seed selection satisfy the first property.

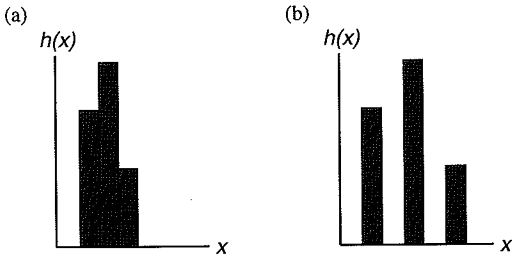

Figure 8.3 Illustration of pseudo peaks in an image histogram (a) caused by linear stretch (b)

8.2.4 Cluster splitting along PC1

In unsupervised classification (cluster partition) very large clusters may be generated. Such large clusters may contain several classes of ground objects. A function for cluster splitting is therefore necessary to achieve optimal convergence. In ISODATA, an overlarge and elongated cluster ω is split according to the variable with greatest standard deviation. The objects (image pixels) in cluster ω are reassigned to either of the two new clusters, $\omega 1$ and $\omega 2$, depending on whether their splitting variable values are above or below the mean of the splitting variable. As shown by the 2D case in Figure 8.4, splitting in this way may cause incorrect assignments of those objects in the shaded area of the data ellipse. They are assigned to a new cluster which is farther away from them rather than closer. This error can be avoided if the cluster ω is split along its first principal component (PC1). Since PC1 can be found without performing a principal component transformation, not too many calculations are involved. The technique of cluster splitting based on PC1 (Liu and Haigh, 1994) includes two steps: finding PC1 followed by cluster splitting based on PC1.

8.2.4.1 *Find the first principal component PC1*

The covariance matrix Σ of the cluster ω is a non-negative definite matrix. Thus the first eigenvalue and eigenvector of Σ, λ_1 and $\mathbf{a} = (a_1, a_2, \ldots, a_n)^{\mathrm{T}}$,

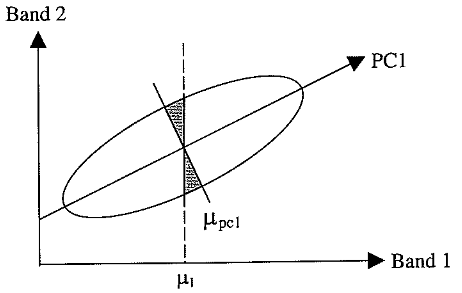

Figure 8.4 A 2D example of cluster splitting based on band 1 (the variable with maximum standard deviation) and PC1. The shaded areas indicate the misclassification resulting from the cluster splitting based on band 1

can be found by the iteration

$$\begin{aligned}\Sigma\mathbf{a}^{(s)} &= \lambda_1^{(s+1)}\mathbf{a}^{(s+1)} \\ \mathbf{a}^{(0)} &= \mathbf{I}\end{aligned} \qquad (8.1)$$

where s denotes the number of iterations and **I** is an identity vector.

As an eigenvector, **a** is orthogonal; thus for each iteration s, we have

$$(\mathbf{a}^{(s)})^{\mathrm{T}}\mathbf{a}^{(s)} = 1. \qquad (8.2)$$

Then

$$\begin{aligned}(\Sigma\mathbf{a}^{(s)})^{t}\Sigma\mathbf{a}^{(s)} &= \lambda_1^{(s+1)}(\mathbf{a}^{(s+1)})^{t}\lambda_1^{(s+1)}\mathbf{a}^{(s+1)} \\ &= (\lambda_1^{(s+1)})^2.\end{aligned}$$

Thus,

$$\begin{aligned}\lambda_1^{(s+1)} &= [(\Sigma\mathbf{a}^{(s)})^{\mathrm{T}}\Sigma\mathbf{a}^{(s)}]^{1/2} \\ a^{(s+1)} &= \frac{\Sigma\mathbf{a}^{(s)}}{\lambda^{(s+1)}}.\end{aligned} \qquad (8.3)$$

After 5–6 iterations convergence with an accuracy higher than 10^{-5} can be achieved and the first eigenvalue λ_1 and eigenvector **a** are found. Consequently, the first principal component of cluster ω in the n-dimensional feature space **X** is derived as

$$\mathrm{PC1} = (\mathbf{a})^{\mathrm{T}}\mathbf{X} = \sum_{h=1}^{n} a_h\mathbf{x}_h. \qquad (8.4)$$

8.2.4.2 *Cluster splitting*

According to (8.4), the PC1 coordinate of the mean vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^{\mathrm{T}}$ of cluster ω is

$$\mu_{\mathrm{pc1}} = (\mathbf{a})^{\mathrm{T}}\boldsymbol{\mu} = \sum_{h=1}^{n} a_h\mu_h. \qquad (8.5)$$

For every data point $\mathbf{X}_j \in \omega$, we calculate its PC1 coordinate:

$$x_{j,\mathrm{pc1}} = (\mathbf{a})^{\mathrm{T}}\mathbf{X}_j = \sum_{h=1}^{n} a_h x_{jh}. \qquad (8.6)$$

We assign $\mathbf{X}_j$ to $\omega 1$ if $x_{j,\mathrm{pc1}} > \mu_{\mathrm{pc1}}$, otherwise we assign $\mathbf{X}_j$ to ω_2.

Cluster splitting can also be performed on the objects (image pixels) instead of data points by replacing $\mathbf{X}_j$ by Y_i $(i = 1, 2, \ldots, N)$ in formula (8.6). After cluster splitting, the statistics of the two new clusters are calculated as the representations for the next clustering iteration.

8.3 Supervised classification

8.3.1 Generic algorithm of supervised classification

A supervised classification comprises three major steps, as follows:

Step 1. Training Training areas representing different ground objects are manually and interactively defined on the image display. Statistics of the training areas are calculated to represent the relevant classes ω_k $(k = 1, 2, \ldots, m)$.

Step 2. Classification For all i, assign any element Y_i $(Y_i \in Y)$ to a class ω_k, if the dissimilarity measurement $\delta(Y_i, \omega_k)$ is minimal.

Step 3. Class statistics Calculate the statistics of all resultant classes.

Iteration and class splitting/merging functions can also be accommodated into a supervised classification algorithm to provide an automated optimization mechanism.

8.3.2 Spectral angle mapping classification

A pixel in an n-band multi-spectral image can be considered as a vector in the n-dimensional feature space **X**. The magnitude (length) of the vector is decided by the pixel DNs of all the bands while the orientation of the vector is determined by the shape of the spectral profile of this pixel. If two pixels have similar spectral properties but are under different solar illumination because of topography, the vectors representing the two pixels will have different lengths but very similar orientation. Therefore the classification of image pixels based on the spectral angles between them will be independent of topography (illumination) as well as any unknown linear translation factors (e.g. gain and offset). The spectral angle mapping (SAM) technique, proposed by Kruse, Lefkoff and Dietz (1993), is a supervised classification based on the angles between image

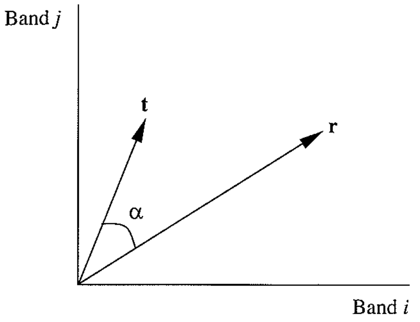

Figure 8.5 A 2D illustration of two spectral vectors and the spectral angle (α) between them

pixel spectra and training data spectra or library spectra. The algorithm determines the similarity between two spectra by calculating the spectral angle between them as shown in a 2D diagram (Figure 8.5). According to vector algebra, the angle between two vectors **r** and **t** is defined as

$$\alpha = \arccos\left(\frac{\mathbf{t}\cdot\mathbf{r}}{|\mathbf{t}|\cdot|\mathbf{r}|}\right) \tag{8.7}$$

or

$$\alpha = \arccos\left(\frac{\sum_{i=1}^{m} t_i r_i}{\left(\sum_{i=1}^{m} t_i^2\right)^{1/2}\left(\sum_{i=1}^{m} r_i^2\right)^{1/2}}\right). \tag{8.8}$$

where m is the number of spectral bands.

The value range of α is 0–π.

In general, for N reference spectral vectors $\mathbf{r}_k$ ($k = 1, 2, \ldots, N$), either from an existing spectral library or from training areas, the spectral vector **t** of an image pixel is identified as $\mathbf{r}_k$ if the angle α between them is minimal and is less than a given criterion.

The SAM classification is widely used in hyperspectral image data classification for mineral identification and mapping. It can also be used in broadband multi-spectral image classification. Within the framework of SAM, different dissimilarity functions can be implemented to assess the spectral angle, α.

8.4 Decision rules: dissimilarity functions

Dissimilarity functions, based on image statistics, formulate decision rules at the core of both supervised and unsupervised classification algorithms, and these theoretically decide the accuracy of a classification algorithm. Here we introduce several commonly used decision rules of increasing complexity.

8.4.1 Box classifier

This is also called a parallel classifier. It is used for single pass supervised classification. In principle, it is simply multi-dimensional thresholding (Figure 8.6a).

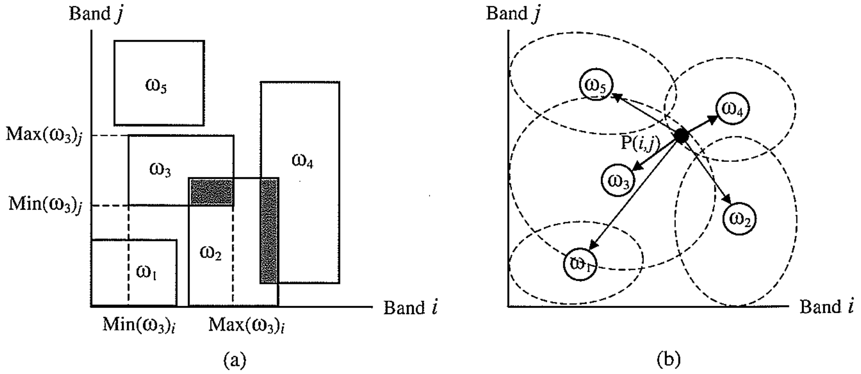

Figure 8.6 Illustrations of 2D feature space partition of the box classifier and distance-based classifications: (a) a box classifier is actually a simple multi-dimensional threshold – it cannot classify image pixels that fall in the value ranges of multiple classes as shown in the shaded areas; (b) the circles are the class centres and the ellipses represent the size of each class. The minimum Euclidean distance classification will assign the pixel P(i,j) to the class centre ω_4 whereas the maximum likelihood minimum distance classification will be more likely to assign the pixel P(i,j) to the class centre ω_3 because this class is larger

For all i, assign an element Y_i ($Y_i \in Y$) to cluster ω_k if

$$\min(\omega_k) \leq Y_i \leq \max(\omega_k). \tag{8.9}$$

The 'boxes' representing the scopes of different classes may partially overlap one another, as in the shaded areas shown in Figure 8.6a. The pixels that fall within the overlap areas are treated as unclassified. This is a very crude but fast classifier.

8.4.2 Euclidean distance: simplified maximum likelihood

The Euclidean distance is a special case of maximum likelihood which assumes equal standard deviation and population for all clusters. It is defined as follows.

For all i, assign an element Y_i ($Y_i \in Y$) to cluster ω_k if

$$d(Y_i, \omega_k) = (Y_i - \mu_k)^{\mathrm{T}}(Y_i - \mu_k) = \min\{d(Y_i, \omega_r)\} \tag{8.10}$$

for $r = 1, 2, \ldots, m$ and where μ_k is the mean vector of cluster ω_k.

The Euclidean distance lies at the core of the ISODATA minimum distance classification.

8.4.3 Maximum likelihood

The maximum likelihood decision rule is based on Bayes' theorem and assumes a normal distribution for all clusters. In this decision rule, the feature space distance between an image pixel Y_i and cluster ω_k is weighted by the covariance matrix Σ_k of ω_k with an offset relating to the ratio of N_k, the number of pixels in ω_k, to N, the total number of pixels of the image Y.

For all i, assign an element Y_i ($Y_i \in Y$) to cluster ω_k if

$$\begin{aligned}\delta(Y_i, \omega_k) &= \ln|\Sigma_k| + (Y_i - \mu_k)^{\mathrm{T}}\Sigma_k^{-1}(Y_i - \mu_k)\\ &\quad -\ln\frac{N_k}{N} = \min\{\delta(Y_i, \omega_r)\}\end{aligned} \tag{8.11}$$

for $r = 1, 2, \ldots, m$.

As shown in Figure 8.6b, the minimum Euclidean distance classification will assign the pixel P(i, j) to the class centre ω_4, whereas the maximum likelihood minimum distance classification will be more likely to assign the pixel P(i, j) to the class centre ω_3 because this class is larger.

8.4.4 *Optimal multiple point reassignment

An advantage of 3D-FSIC is that the optimal multiple point reassignment (OMPR) rule can be implemented if we let $\delta(\mathbf{X}_j, \omega_k)$ be an OMPR dissimilarity measurement at step 3 of 3D-FSIC. The OMPR (Kittler and Pairman, 1988) was developed based on the optimal point assignment rule (Macqueen, 1967). By using OMPR, the cluster sizes and the number of pixels sharing the same data point in feature space (point frequency) are taken into account when a reassignment of these pixels is made. Thus the accuracy of the clustering partition can be reasonably improved.

Suppose a data point $\mathbf{X}_j$ currently allocated to cluster ω_l is shared by H pixels; then the OMPR based on the square Euclidean distance (Euclidean OMPR) for all these pixels from cluster ω_l to cluster ω_k, shared by N_k pixels, will be achieved if ω_k satisfies

$$\begin{aligned}\frac{N_k}{N_k + H} d(\mathbf{X}_j, \mu_k) &= \min_{r \neq l} \frac{N_r}{N_r + H} d(\mathbf{X}_j, \mu_r)\\ &< \frac{N_l}{N_l - H} d(\mathbf{X}_j, \mu_l)\end{aligned} \tag{8.12}$$

where N_r is the number of pixels in any cluster ω_r and N_l that in cluster ω_l.

If the clusters are assumed to have a normal distribution (Gaussian model), the Gaussian OMPR is formed as follows.

For all j, assign a data point $\mathbf{X}_j$ in cluster ω_l to cluster ω_k if

$$\begin{aligned}\delta(X_j, \omega_k) &= \min_{r \neq l} \delta(\mathbf{X}_j, \omega_r)\\ &< \ln|\Sigma_l| - \frac{N_l - H}{H} \ln\left[1 - \frac{H}{N_l - H}\Delta(\mathbf{X}_j, \omega_l)\right]\\ &\quad -2\ln\frac{N_l}{N} - (D + 2)\frac{N_l - H}{H}\ln\frac{N_l}{N_l - H}\end{aligned} \tag{8.13}$$

where

$$\delta(\mathbf{X}_j,\omega_r)=\ln|\Sigma_r|+\frac{N_r+H}{H}\ln\left[1+\frac{H}{N_r+H}\Delta(\mathbf{X}_j,\omega_r)\right]$$
$$-2\ln\frac{N_r}{N}+(D+2)\frac{N_r+H}{H}\ln\frac{N_r}{N_r+H} \qquad (8.14)$$

and

$$\Delta(\mathbf{X}_j,\omega_r)=(\mathbf{X}_j-\mu_r)^{\mathrm{T}}\Sigma_r^{-1}(\mathbf{X}_j-\mu_r)$$

with D the dimensionally of feature space $\mathbf{X}$.

In the OMPR method, data point inertia is considered. A data point shared by more pixels ('heavier') is more difficult to move from one cluster to another than a 'lighter' point.

8.5 Post-classification processing: smoothing and accuracy assessment

8.5.1 Class smoothing process

A classification image appears to be a digital image in which the DNs are the class numbers, but we cannot perform numerical operations on class numbers. For instance, the average of class 1 and class 2 cannot be class 1.5! Indeed, the class numbers in a classification image do not have any sequential relationship; they are nominal values and can be treated as symbols such as A, B and C (see also Section 12.3). A classification image is actually an image of symbols, *not* digital numbers; it is therefore *not* a digital image in the generally accepted sense. As such we cannot apply any numerical-operation-based image processing to classification images.

A classification image often contains noise caused by the isolated pixels of some classes, within another dominant class, which can form sizeable patches (Figure 8.7a). It is reasonable to presume that these isolated pixels are more likely to belong to this dominant class rather than to the classes that they are initially assigned to; these probably arise from classification errors. An appropriate smoothing process applied to a classification image will not only 'clean up' the image, making it visually less noisy, but also improve the accuracy of classification.

Among the low-pass filters that we have described so far, the only filter you can use to smooth a classification image is the *mode (majority) filter.* The reason for this is simple, since the mode filter smoothes an image without any numerical operations. For instance, if a pixel of class 5 is surrounded by pixels of class 2, the mode filter will reassign this pixel to class 2 according to the majority class in the

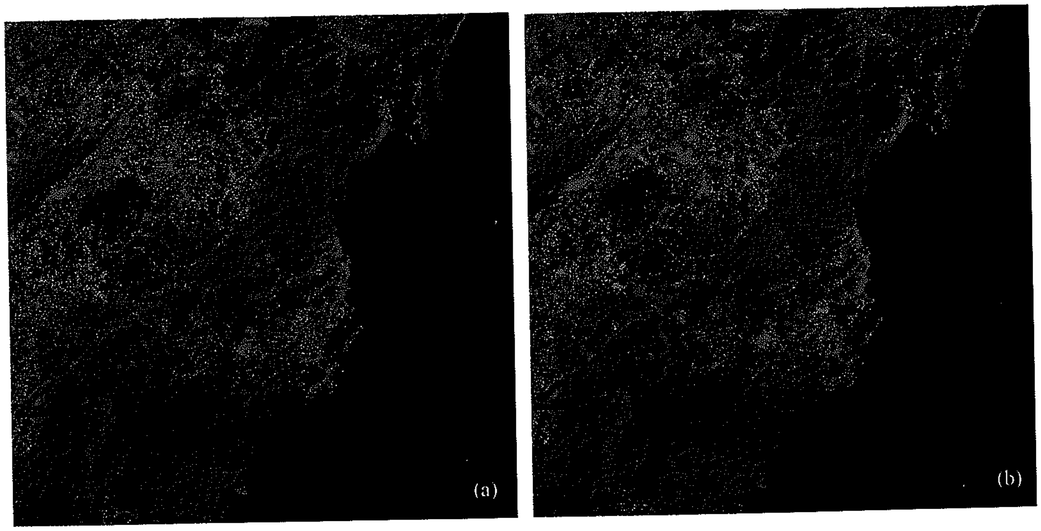

Figure 8.7 Smoothing classification images: (a) ISODATA unsupervised classification with 24 classes; and (b) the classification image smoothed using a mode filter with a majority of 5 in a 3 × 3 filtering kernel. A closer look at these images reveals the difference between them: image (b) is smoother than image (a)

filtering kernel. Figure 8.7b illustrates the effect of mode filtering applied to an unsupervised classification image in Figure 8.7a.

8.5.2 Classification accuracy assessment

Ultimately there is no satisfactory method to assess the absolute accuracy of image classification for remote sensing Earth observation applications (see also Section 17.5.1). The paradox is that we cannot conduct such an assessment without knowing 100% of ground truth on one hand, while, on the other hand, if we do have complete knowledge of ground truth, what is the point of the classification? Even an assessment or an estimate of relative accuracy of classification does, however, provide valuable knowledge for us to accept or reject a classification result at a certain confidence level. There are two generally accepted approaches to generate ground truth:

1. Use field-collected data of typical classes as samples of ground truth. For rapidly and temporally changing land cover classes, such as crops, field data should be collected simultaneously with image acquisition. For temporally stable targets, such as rocks and soils, published maps as well as field data can be used. The classification accuracy of the sample areas with known classes gives an estimate of total classification accuracy. This seemingly straightforward approach is often impractical in reality because it is often constrained by errors in the recording of field observations, limited field accessibility and temporal irrelevance.
2. Another approach relies on image training. This uses typical spectral signatures and limited field experience, where a user can manually specify training areas of various classes using a multispectral image. The pixels in these training areas are separated into two sets: one is used to generate class statistics for supervised classification and the other for subsequent classification accuracy assessment. For a given training area, we could take a selection of pixels sampled from a 2 × 2 grid as the training set while the remaining pixels are used for the verification set (ground truth reference data), as shown in Figure 8.8. The pixels in the verification set are assumed to belong to the same class as their corresponding training set. In another way, we can also select several training areas for the same class and use some of them for training and the rest for verification.

In practice the above two approaches are often used in combination.

Suppose that we have some kind of ground truth reference data; then a widely used method to describe the relative accuracy of classification is the *confusion matrix*:

$$\begin{pmatrix} C_{11} & C_{12} & \cdots & C_{im} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mm} \end{pmatrix}. \qquad (8.15)$$

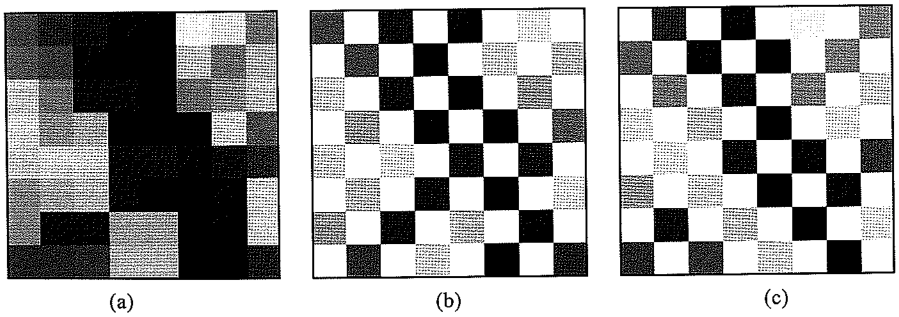

Figure 8.8 A resampling scheme for classification accuracy assessment. An image (a) is resampled to formulate two images (b) and (c); one is used as the training dataset while the other is used as the verification set

Here, each of the elements, C_{ii}, in the major diagonal represents the number of pixels that are correctly classified for class i. Any element off the major diagonal, C_{ij}, represents the number of pixels that should be in class i but which are incorrectly classified as class j. Obviously, if all the image pixels are correctly classified, we should then have a diagonal confusion matrix where all non-diagonal elements become zero. The sum of all the elements in the confusion matrix is the total number of pixels in a classification image, N:

$$N = \sum_{i=1}^{m}\sum_{j=1}^{m} C_{ij}.$$

The ratio between the summation of the major diagonal elements and the total number of pixels represents the percentage of the correct classification or *overall accuracy*:

$$\text{Ratio}_{\text{correct}}(\%) = \frac{1}{N}\sum_{i=1}^{m} C_{ii}. \tag{8.16}$$

The sum of any row i of the confusion matrix gives the total number of pixels that, according to the ground truth reference, should be in class i, Nr_i:

$$Nr_i = \sum_{j=1}^{m} C_{ij}.$$

Then the ratio C_{ii}/Nr_i is the percentage of correct classification of class i, according to the ground truth references and is often called *user's accuracy.*

The sum of any column j of the confusion matrix gives the total number of pixels that have been classified as class j, Nc_j:

$$Nc_j = \sum_{i=1}^{m} C_{ij}.$$

Then the ratio C_{ii}/Nc_i is the percentage of correct classification of class j, based on the classification result, and is often called *producer's accuracy.*

Apart from the above accuracy measurements which are based on simple ratios, another commonly used statistical measure of classification accuracy and quality is the *kappa coefficient* (κ) that combines the above two class accuracy estimations, based on the rows and columns of the confusion matrix, to produce an estimate of total classification accuracy, as follows:

$$\kappa = \frac{N\sum_{i=1}^{m} C_{ii} - \sum_{i=1}^{m} Nr_i \cdot Nc_i}{N^2 - \sum_{i=1}^{m} Nr_i \cdot Nc_i}. \tag{8.17}$$

In the case of 100% agreement between the classification and the reference data, the confusion matrix is diagonal, that is $\sum_{i=1}^{m} C_{ii} = N$. Thus,

$$\kappa = \frac{N^2 - \sum_{i=1}^{m} Nr_i \cdot Nc_i}{N^2 - \sum_{i=1}^{m} Nr_i \cdot Nc_i} = 1,$$

while if there is no agreement at all, then all the elements on the diagonal of the confusion matrix are zero, that is $\sum_{i=1}^{m} C_{ii} = 0$. In this case

$$\kappa = \frac{-\sum_{i=1}^{m} Nr_i \cdot Nc_i}{N^2 - \sum_{i=1}^{m} Nr_i \cdot Nc_i} < 0.$$

In summary, the maximum value of the kappa coefficient κ is 1, indicating perfect agreement between the classification and the reference data, while for no agreement κ becomes negative. The minimum value of κ is case dependent, but as long as $\kappa \leq 0$, it indicates zero agreement between the classification and the reference data.

As illustrated in Table 8.1, the numbers in bold italics form the confusion matrix. Nr_i and C_{ii}/Nr_i are listed in the two right-hand columns, while Nc_j and C_{jj}/Nc_j appear in the bottom two rows. The bold number in the bottom right corner is the total percentage of correct classification. The kappa coefficient can then be calculated from Table 8.1 by

$$\kappa = \frac{403 \times 308 - 33\,023}{162\,409 - 33\,023} = \frac{911\,01}{129\,386} = 0.704.$$

Despite the fact that the classification accuracy derived from the confusion matrix is very much a self-assessment and is by no means the true accuracy of classification, it does provide a useful measure of classification accuracy. The information in a confusion matrix is highly dependent on the quality of the training areas and field data. Well- selected training

Table 8.1 An example confusion matrix

Class Reference	Class 1	Class 2	Class 3	Class 4	Class 5	Row sum Nr_i	C_{ii}/Nr_i (%)
Reference 1	***56***	***9***	***5***	***2***	***8***	80	70.0
Reference 2	***10***	***70***	***7***	***3***	***5***	95	73.7
Reference 3	***0***	***3***	***57***	***10***	***6***	76	75.0
Reference 4	***0***	***6***	***0***	***79***	***4***	89	88.8
Reference 5	***8***	***4***	***3***	***2***	***46***	63	73.0
Column sum Nc_j	74	92	72	96	69	403	
C_{jj}/Nc_j (%)	75.6	76.1	79.2	82.3	66.7		**76.4**

areas can improve both the classification accuracy and the credibility of accuracy assessment, whereas poorly selected training areas will yield low classification accuracy and unreliable accuracy assessment. Strictly speaking, this method only gives an estimate of the classification accuracy of the whole image.

8.6 Summary

In this chapter, we have introduced the most commonly used image classification approaches and algorithms. These methods are essentially multi-variable statistical classifications that achieve data partition in the multi-dimensional feature space of multi-layer image data, such as a multi-spectral remotely sensed image.

The iterative clustering method of unsupervised classification enables self-optimization of a local optimal representative of the natural clusters in the data. How well the clustering converges to a local optimal depends on the dissimilarity function and clustering mechanism employed, while the quality of the local optimal is mainly affected by the initial cluster centres (the seeds) from where the iteration starts. Thus a seed selection technique, locating the peaks of data distribution, is introduced. A method for cluster splitting, based on PC1, is also proposed to improve the clustering mechanism.

Though affected by the same factors, the accuracy of a supervised classification is largely controlled by the user's knowledge. High-quality user knowledge could lead to correct classification of known targets while poor user knowledge may mislead rather than help.

There are many methods of accuracy assessment, such as the well-known confusion matrix, but it is important to know the limitations of such methods that merely give a relative assessment rather than the true accuracy of classification.

Finally, we must recognize that a classification image is not a true digital image but a symbol image presented in numbers. We could apply numerical operations to a classification but the results do not really make any sense. We can, however, use logical operations to process classification images, such as smoothing a classification image using a mode (majority) filter because it does not involve any numerical operations.

Questions

8.1 What is multi-variable statistical classification? Describe the major approaches for image classification.

8.2 What are the advantages and disadvantages of unsupervised classification? Describe the algorithm α for iterative clustering.

8.3 Explain, using a diagram, the self-optimization mechanism of iterative clustering.

8.4 Describe the main steps of the 3D-FSIC algorithm with the aid of diagrams. What are the main advantages and limitations of feature space iterative clustering?

8.5 What are the two properties for the design of the automatic seed selection technique?

8.6 What is the problem with cluster splitting along the axis of the variable with the maximum standard deviation? What is a better approach?

8.7 Describe the general steps of supervised classification.

8.8 Explain the principle of spectral angle classification and its merits.

8.9 What is a confusion matrix? Based on the confusion matrix, give definitions for overall accuracy, user's accuracy and producer's accuracy.

8.10 Comment on the issue of accuracy assessment for image classification.