Chapter 11
Image Classification in Practice

11.1 Introduction

The material in many of the previous chapters has presented techniques that can be used to classify an image and create a thematic map. What we have to do now is to see how those techniques can be put into methodologies that make the classification task efficient and accurate. Using an appropriate methodology is probably one of the most important considerations in applying classification to data recorded by imaging sensors, and is often overlooked in practical image labelling. It is the role of this chapter to consider these operational aspects of thematic mapping, including how the accuracy of the final thematic map product can be assessed. We will consider accuracy determination in detail, including guidelines for choosing the number of samples needed to measure map accuracy effectively.

It is important to distinguish between classification techniques developed in the research laboratory and those applied in practice. While new algorithms evolve regularly, in pursuit of machine learning and mapping goals, many can be difficult to use in an operational sense, not because they don’t perform well, but because they have not reached operational maturity. That can put them beyond the reach of a typical operational image analyst faced with a practical mapping task, who is more interested in outcomes than the novelty of an algorithm.

The most commonly used approach to thematic mapping over the past three or so decades has been maximum likelihood classification based on the assumption of multivariate normally distributed spectral classes. Although the neural network approach was common for a decade or so, most analysts who choose an algorithm different from maximum likelihood classification tend now to focus on the support vector machine with an appropriate kernel transformation. Consequently, this coverage will concentrate on those two methods as prototype parametric and non-parametric procedures respectively. In passing, we will comment on variants such as the minimum distance technique, but restricting our focus to the two most common approaches is sufficient to demonstrate practical matters of importance.
One of the classification methodologies we will treat later in the chapter is the use of decision trees. Those structures have many benefits, including the ability to partition features into subsets for efficient scene labelling. We will also consider collections of decision trees (sometimes referred to as forest classifiers), to see how they perform in enhancing classification results in the manner of committee classifiers and the AdaBoost process. We also examine how scientific knowledge can be used in the classification of high spectral resolution data, and how sub-pixel class mixtures can be handled.

We commence with a summary of the essential aspects of classification as the foundation on which to build the methodologies to be presented.

### 11.2 An Overview of Classification

#### 11.2.1 Parametric and Non-parametric Supervised Classifiers

The essence of supervised classification is that the analyst acquires beforehand a labelled set of training pixels for each of the classes of interest, and desirably for all classes in the image. Often that entails the analyst obtaining reference data such as aerial photographs, maps of the region of interest, or even hard copy products of the image data, from which careful photointerpretation generates a set of acceptable training pixels.

The training data is used to estimate the parameters or other constants required to operate the chosen classification algorithm. If the algorithm involves explicit parameters, such as the mean vector and covariance matrix for the multivariate normal distribution, then the technique is called parametric. For algorithms which do not involve sets of parameters of those types the term non-parametric is used even though constants have to be estimated, such as the kernel parameters in the support vector machine approach. Once the appropriate parameters or constants have been estimated using the training data, the algorithm is then ready to be employed on unseen (testing) pixels—it is then said to be trained.

As a proportion of the full image to be analysed the amount of training data will often be less than 1–5% of the image pixels. The learning phase, therefore, in which the analyst plays an important part in the labelling of pixels beforehand, is performed on a very small part of the image. Once trained on such a small portion of the image, the classifier is then asked to attach labels to all the image pixels. That is where a significant benefit occurs in thematic mapping.

The output from the supervised classification approach typically consists of a thematic map of class labels, often accompanied by a table of area estimates and, importantly, an error matrix which indicates by class the residual error, or accuracy, of the final product.
11.2.2 Unsupervised Classification

Unsupervised classification is an analytical procedure based, generally, on clustering algorithms such as those treated in Chap. 9. Clustering partitions a sample of the image data in spectral space into a number of distinct clusters or classes. It then labels all pixels of interest as belonging to one of those classes to produce a cluster map, albeit with purely symbolic labels at this stage and not labels that indicate ground cover types.

In contrast to the prior use of analyst-provided information in supervised classification, unsupervised classification is a segmentation of the spectral space in the absence of any information fed in by the analyst. Instead, analyst knowledge is used afterwards to attach class labels to the map segments established by clustering, often guided by the spatial distribution of the labels shown in the cluster map. Clearly this is an advantage of unsupervised classification. However, by comparison to most techniques for supervised classification, clustering is a time-consuming process. This can be demonstrated by comparing the multiplication requirements of the iterative clustering algorithm of Sect. 9.3 with the maximum likelihood decision rule of Sect. 8.3.3.

Suppose a particular exercise involves \( N \) bands and \( C \) classes or clusters. Maximum likelihood classification requires \( CPN(N + 1) \) multiplications\(^1\) where \( P \) is the number of pixels in the segment to be classified. Clustering the same dataset requires \( PCI \) distance measures for \( I \) iterations. Each distance calculation requires \( N \) multiplications, so that the total number of multiplications for clustering is \( PCIN \). Thus the speed comparison of the two approaches is approximately \( (N + 1)/I \). Clustering would, therefore, need to be completed within 7 iterations for a 6 band dataset to be faster than maximum likelihood classification. Frequently about 20 times this number of iterations is necessary to achieve an acceptable clustering. Training the classifier adds a loading to its time demand; however a significant time loading should also be added to clustering to account for the labelling step. Often that is done by associating pixels with the nearest cluster using a Euclidean distance measure in the spectral space, including any that were not used in clustering.

Because of the time demand of clustering algorithms, unsupervised classification is not often carried out with large image segments. Usually a representative subset of data is employed for the actual clustering phase in order to segment the spectral space. That information is then used to assign all the image pixels to one of the clusters to create the unsupervised thematic map.

When comparing the time requirements of supervised and unsupervised classification it must be remembered that a large demand on the analyst’s time is required for training the supervised procedure. That is necessary both for gathering reference data and for identifying training pixels using that data. The corresponding step in unsupervised classification is the labelling of clusters afterwards.

---

\(^1\) See Sect. 8.5.6.
While that still requires user effort to gather labelled prototype data, not as much may be needed as when training a supervised procedure. Data is only required for those classes of interest. Also, often only a handful of labelled pixels is necessary to identify a given class because we can use spatial cues to help in that process. By comparison, in supervised training sufficient training pixels per class are required to ensure that reliable estimates of class signatures are generated. That can be a particular problem with high dimensionality data such as that recorded by imaging spectrometers.

A final point that must be taken into account when contemplating unsupervised classification by clustering is that there is no facility for including prior probabilities of class membership. By comparison, many supervised algorithms allow prior knowledge to bias the result generated from spectral data alone.

Once classes have been generated for the purpose of unsupervised classification, it is of value to use separability measures to see whether some clusters are sufficiently similar spectrally that they should be combined. That is particularly the case when the classes are generated on a sample of data; separability checking and merging would be carried out before cluster maps are produced.

### 11.2.3 Semi-Supervised Classification

When insufficient labelled training samples are available it is still possible to use supervised classification algorithms by employing one of a number of techniques that use unlabelled data to supplement known training pixels. They are known as semi-supervised training procedures; we do not treat them any further in this coverage, but they feature strongly in the machine learning community.

### 11.3 Supervised Classification with the Maximum Likelihood Rule

#### 11.3.1 Outline

The underlying requirement of maximum likelihood classification is that the analyst has available a sufficient number of known training pixels to allow

---


representative signatures to be developed for each information class. When the data dimensionality is high it is frequently necessary to undertake some form of feature selection; that is almost always the case when seeking to classify hyperspectral data with traditional statistical algorithms.

The signatures required for the maximum likelihood approach to thematic mapping are the mean vectors and covariance matrices of each of the classes of interest. The analyst is effectively teaching the classification algorithm to recognise the spectral characteristics of each class by presenting the classifier with the prototype pixels from which the class signatures are estimated.

### 11.3.2 Gathering Training Data

The prior identification of training pixels, or more commonly training fields, may be an expensive exercise if field visits are required. As noted earlier, topographic maps and aerial photographs can be used, as can image products formed from selected of bands of the image data to be classified. When choosing training fields from imagery or aerial photographs, good photointerpretive skills are necessary. If the training data is to be taken from a map it is important that the scales of the map and image data be comparable. It is particularly useful if the image can be registered to the map beforehand so that selected polygons from the map can be laid over the image to identify pixels of given cover types for training.

It is necessary to collect training data at least for all classes of interest and, preferably, for all apparent classes in the segment of image to be classified. In either case, and particularly if the selection of training data is not exhaustive or fully representative, it is prudent to use some form of threshold or limit on the classification so that pixels in the image that are not well represented in the training data are excluded in a trial classification. Such a limit can be imposed in maximum likelihood classification by the use of thresholds on the discriminant functions. By limiting a classification in this way, pixels in the image that are not well characterised by the training data will not be classified. That will identify weaknesses in the selection of the training data which can be rectified; the image can then be reclassified. Repeated refinement of the training data, and reclassification, can be carried out using a representative portion of the image data. That is important if the image requiring analysis is very large.

---

4 For other supervised rules the class signatures will be different. For example, for parallelepiped classification class signatures are the upper and lower bounds of brightness in each spectral band. For the minimum distance classifier the signatures are the mean vectors of the training data in each class.

5 See Sect. 8.3.5.
11.3.3 Feature Selection

When performing a classification it is sensible to use no more features than necessary to get results at the level of accuracy required. There are two reasons for this. First, the more features involved in a classification the more costly the exercise. In the case of the maximum likelihood classifier, in particular, classification cost increases with the square of the number of features.

Secondly, and more importantly, the well-known Hughes effect alerts us to the drop in classifier performance that can be experienced when there are not enough training samples available relative to the number of features used in the classifier. Estimates of the maximum likelihood signatures in training are severely compromised if the dimensionality of the mean vector and covariance matrix is high compared with the number of pixels used. Clearly, this is most critical when using sensors with large numbers of bands, such as imaging spectrometers.

Before being overly concerned about this problem, however, one should look at the practical aspects of applying a machine-assisted classifier for the purpose of pixel identification in thematic mapping. It is questionable as to whether one would want to use all or a very large number of the bands available in a hyperspectral data set when considering machine learning approaches to pixel labelling. The benefit of the large number of high resolution spectral bands generated by an imaging spectrometer is fundamentally so that scientific methods can be brought to bear in pixel identification. In particular, the identification of fine absorption features and similar characteristics that require high spectral resolution is why imaging spectrometer data sets are so valuable. The scientific expertise of a spectroscopist can then be used and searching libraries of laboratory spectra becomes a viable recognition method. This approach is taken up in Sect. 11.9 below.

When considering the application of machine learning algorithms for thematic mapping it makes little sense to contemplate using all the bands available. Some exercises only require data in given regions of the spectrum. Moreover, adjacent bands tend to be highly correlated because of overlaps of the bandpass filters that define the individual band measurements. That is not to say that considerable diagnostic benefit is not available in the large numbers of bands from an imaging spectrometer, but in a practical exercise a consequence of injudiciously using a large number of bands, without some consideration beforehand of their relevance, is likely to prejudice the performance of the classification.

When using a maximum likelihood rule, feature selection can be guided by the application of separability measures such as those treated in Chap. 10. Those metrics have another benefit when dealing with classes that are assumed to be represented by multivariate normal distributions. By checking their statistical separability we can assess whether any pair of classes are so similar in spectral space that a significant misclassification error would occur if they were both used. If they are too close, they should be merged.
11.3.4 Resolving Multimodal Distributions

One of the attractions of the maximum likelihood approach to thematic mapping, based on the multivariate normal distribution, is that it can model class distributions that are elongated to different extents in different directions in the spectral domain. It also offers the guarantee that, if properly applied, it will lead to minimum average classification error. Further, it is a simple classifier to use. Its major limitation is that the classes must be well represented by multivariate normal distributions. Often the information classes of interest to the user will not be of that form; either the distribution of pixels in a given information class will be smeared out in spectral space or will appear as sets of moderately distinct clusters, each of which can be modelled by a single normal distribution. In the latter case, the distribution is said to be multimodal. In order to apply the maximum likelihood rule we need to resolve those modes. In the terminology of remote sensing that means finding the constituent set of spectral classes that are needed to model each information class so that good classifier performance will result. That is the step most often overlooked when using Gaussian maximum likelihood classification, as a consequence of which many users find that the maximum likelihood rule does not work as well as might otherwise be expected.

Multimodality can be identified using clustering algorithms, and that is the basis of the hybrid classification methodology presented in Sect. 11.4 below. Although data dimensionality with contemporary sensors is very high, a simple means by which multimodal behaviour can be observed is to produce scatter plots of the data in each training class. A scatter plot is a two-dimensional spectral subspace with user-defined axes. If vegetation were the information class of interest, an infrared versus visible red scatter plot could indicate the need to introduce sets of spectral classes to represent single information classes.

Another way of resolving multimodality is to use the Gaussian mixture models treated in Sect. 8.4. Although appealing theoretically, that approach is computationally more difficult than using clustering. The more general hybrid methodology, referred to as cluster space classification, and treated in Sect. 11.5, is another approach that should be considered.

11.3.5 Effect of Resampling on Classification

It is much easier to work with image data that has been registered to a map grid using the techniques of Sect. 2.18. That requires an interpolation technique with which to synthesise pixel values for placement on the map grid. The two most common interpolation procedures are nearest neighbour resampling and resampling by cubic convolution. In the former, the original image pixels are simply relocated onto a geometrically correct map grid whereas, in the latter, new pixel brightness values are created by interpolating over a group of 16 neighbouring pixels.
Clearly, it is also desirable to have thematic maps registered to a map base. That can be done by rectifying the image before classification or by rectifying the thematic map, in which case nearest neighbour resampling is the only interpolation option available.

An advantage in correcting the image beforehand is that it is often easier to relate reference data to the image if it is in correct geometric registration to a map. However, a drawback with doing this prior to classification is that some of the relative pixel brightnesses may be changed by the interpolation process used. Since cubic convolution fits an interpolating function over 16 neighbouring pixels, the brightnesses in the rectified product are not the original brightnesses recorded by the imaging sensor. As a result, the new brightnesses for some pixels may be atypical and not classify well. By comparison, the brightnesses of the pixels in a corrected image based on nearest neighbour resampling are the original pixel brightness values, simply relocated. In that case resampling cannot affect subsequent classification results.

The influence of resampling on classification accuracy has been considered since the earliest days of remote sensing image processing. Cubic convolution interpolation in particular, has been shown to have a major influence across boundaries such as that between vegetation and water, leading to uncertainties in classification.6

When images in a multitemporal sequence have to be classified to extract change information it is necessary to perform image to image registration. Since registration cannot be avoided, again nearest neighbour resampling should be used. It is most effective when the scales of the two products to be registered are comparable.

### 11.4 A Hybrid Supervised/Unsupervised Methodology

#### 11.4.1 Outline of the Method

The strength of supervised classification based on the maximum likelihood procedure is that it minimises classification error for classes that are distributed in a multivariate normal fashion. It can also label data relatively quickly. Its major drawback lies in the need to have delineated unimodal spectral classes beforehand. This, however, is a task that can be handled by clustering on a representative subset of image data. Used for this purpose, unsupervised classification performs the valuable function of identifying the existence of all spectral classes, yet it is not expected to perform the entire classification. Consequently, the rather logical

---

hybrid classification procedure outlined below has been shown to work well. It consists of five fundamental steps:

Step 1: Use clustering to determine the spectral classes into which the image resolves. For reasons of economy this is performed on a representative subset of data. Spectral class statistics are produced from this unsupervised step.

Step 2: Using available reference data associate the spectral classes, or clusters, with information classes. Frequently there will be more than one spectral class for each information class.

Step 3: Perform a feature selection to see whether all features (bands) need to be retained for reliable classification.

Step 4: Using the maximum likelihood algorithm, classify the entire image into the set of spectral classes.

Step 5: Label each pixel in the classification with the information class corresponding to its spectral class and use independent testing data to determine the accuracy of the classified product.

We now consider some of the steps in detail and introduce some practical concepts. The method depends for its accuracy, as do all classifications, on the skills and experience of the analyst. Consequently, in practice it is not unusual to iterate over sets of steps as experience is gained with the particular problem at hand.

### 11.4.2 Choosing the Image Segments to Cluster

Clustering is applied to a subset of the total image to find suitable spectral classes. Although this will depend on experience, it is recommended that about 3–6 small regions, called candidate clustering areas, be chosen for the purpose. They should be well spaced over the image, located such that each one contains several of the information classes of interest, and such that all information classes are represented in the collection of clustering areas. An advantage of choosing heterogeneous regions to cluster, as against the apparently homogeneous training areas that are used for simple supervised classification, is that mixture pixels which lie on class boundaries will be identified as legitimate spectral classes that represent mixed information classes.

With most clustering procedures the analyst has to specify a set of parameters that control the number of clusters generated. About 2–3 spectral classes per information class have been found to be useful in general; clustering parameters

---

7 This procedure is developed in M.D. Fleming, J.S. Berkebile and R.M. Hofer, Computer aided analysis of Landsat 1 MSS data: a comparison of three approaches including a modified clustering approach, Information Note 072475, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana, 1975 http://www.lars.purdue.edu/home/references/LTR_072475.pdf.
should be selected with that in mind. This number of clusters could be chosen conservatively high because unnecessary classes can be deleted or merged at a later stage.

It is of value to cluster each region separately as that has been found to produce cluster maps with more distinct boundaries than when all regions are pooled beforehand.

11.4.3 Rationalising the Number of Spectral Classes

When clustering is complete the spectral classes are then associated with information classes using the available reference data. It is then necessary to see whether any spectral classes can be discarded, or more importantly, whether sets of clusters can be merged, thereby reducing the number. Decisions about merging can be made on the basis of separability measures, such as those treated in Chap. 10.

During this rationalisation process it is useful to be able to visualise the locations of the spectral classes. For this a bispectral plot can be constructed. The bispectral plot is not unlike a two-dimensional scatter plot of the spectral space in which the data appears. However, rather than displaying the individual pixels, class means are shown by their spectral coordinates. The most significant pair of spectral bands would be chosen in order to see the relative locations of the clusters centres. Sometimes several biplots are produced using different band combinations.

11.4.4 An Example

We now present a simple example to illustrate some key features of the hybrid approach. Because of the simplicity of this illustration not all steps outlined earlier are involved; but the example highlights the value of using unsupervised classification as a means for identifying spectral classes and for generating signatures of classes for which the acquisition of training data would be difficult.8

Figure 11.1a shows a small image segment recorded by the HyVista HyMap sensor over the city of Perth in Western Australia. It is centred on a golf course. The obvious cover types are water, grass (fairways), trees, bare ground including bunkers, a clubhouse, tracks and roads. Apart from a few cases, the distribution of cover types suggests that it might be hard to generate training fields for all classes of interest.

With the direction in which the image was recorded, north is to the right. Shown on the figure are three fields that were used as clustering regions. Inspection shows that those fields among them cover all the obvious information classes.

8 A further example of the use of the hybrid approach will be found in G.E. Moreton and J.A. Richards, Irrigated crop inventory by classification of satellite image data, *Photogrammetric Engineering and Remote Sensing*, vol. 50, 1984, pp. 729–737.
Rather than cluster each field independently, which is generally preferable, the three fields were aggregated and an Isomap clustering algorithm was applied to the group with the requirement to generate 10 clusters. Because there were seven information classes, and some are clearly very homogeneous, it was felt that 10 clusters would be adequate. Five bands were chosen for the exercise: band 7 (visible green), band 15 (visible red), band 29 (near infrared), band 80 (mid infrared) and band 108 (mid infrared). The last two were chosen on the infrared maxima of the vegetation and soil curves, midway between the water absorption regions. It was felt that they would assist in discriminating among the bare ground and roadway classes.

Figure 11.1b shows the cluster maps produced for each of the three fields outlined in Fig. 11.1a. Table 11.1 shows the mean vectors for each of the 10 clusters, along with information class labels generated by observing the clusters against the image, looking at their spatial orientation and distribution, and noting

---

9 All processing for this example was carried out using the MultiSpec© package developed at Purdue University, West Lafayette, Indiana.

10 Ordinarily these labels would be generated by using available reference data. In this case, photointerpretation of the original image easily reveals the information classes. This can be supplemented by other data for the region such as that available on Google™ Earth.
where they fall in the bispectral plot seen in Fig. 11.2. Of particular note is the ease with which signatures have been generated for the two elongated classes of tracks and roads. Although only the mean vectors are shown here, the clustering processor in MultiSpec also generates the covariance matrices that are used in the subsequent classification step.

The infrared versus red bispectral plot in Fig. 11.2 shows information class labels attached to the cluster means. As observed, there are two (spectral) classes of grass, and two of sparse vegetation. There is also a thin border class of mixed vegetation and water pixels; that often happens in practice and would not get picked up in a traditional supervised classification exercise. Here it will be attributed to the water class.

Table 11.1  Cluster mean vectors and associated information class labels

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Pixels</th>
<th>7 (0.511 µm)</th>
<th>15 (0.634 µm)</th>
<th>29 (0.847 µm)</th>
<th>80 (1.617 µm)</th>
<th>108 (2.153 µm)</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,957</td>
<td>1516.7</td>
<td>1074.9</td>
<td>6772.0</td>
<td>3690.6</td>
<td>1762.8</td>
<td>Grass</td>
</tr>
<tr>
<td>2</td>
<td>1,216</td>
<td>1889.3</td>
<td>2275.3</td>
<td>4827.2</td>
<td>4766.3</td>
<td>3304.4</td>
<td>Bare</td>
</tr>
<tr>
<td>3</td>
<td>1,643</td>
<td>1301.7</td>
<td>984.6</td>
<td>6005.4</td>
<td>3299.1</td>
<td>1574.9</td>
<td>Grass</td>
</tr>
<tr>
<td>4</td>
<td>1,282</td>
<td>1592.5</td>
<td>1535.5</td>
<td>5503.9</td>
<td>4006.1</td>
<td>2275.6</td>
<td>Sparse veg</td>
</tr>
<tr>
<td>5</td>
<td>668</td>
<td>1012.9</td>
<td>830.8</td>
<td>4884.3</td>
<td>2519.4</td>
<td>1230.0</td>
<td>Sparse veg</td>
</tr>
<tr>
<td>6</td>
<td>649</td>
<td>1496.5</td>
<td>1708.4</td>
<td>3813.4</td>
<td>3369.0</td>
<td>2332.9</td>
<td>Tracks</td>
</tr>
<tr>
<td>7</td>
<td>950</td>
<td>707.7</td>
<td>540.2</td>
<td>3938.0</td>
<td>1600.1</td>
<td>722.6</td>
<td>Trees</td>
</tr>
<tr>
<td>8</td>
<td>669</td>
<td>1397.3</td>
<td>1509.3</td>
<td>2368.0</td>
<td>2135.2</td>
<td>1745.9</td>
<td>Bldg/road</td>
</tr>
<tr>
<td>9</td>
<td>462</td>
<td>642.1</td>
<td>625.2</td>
<td>2239.5</td>
<td>1194.4</td>
<td>656.9</td>
<td>Water/veg</td>
</tr>
<tr>
<td>10</td>
<td>2,628</td>
<td>438.4</td>
<td>376.7</td>
<td>469.7</td>
<td>276.0</td>
<td>182.0</td>
<td>Water</td>
</tr>
</tbody>
</table>

Fig. 11.2 Near infrared versus visible red bispectral plot showing the cluster means and information class labels; note that two information classes each consist of two spectral classes.

where they fall in the bispectral plot seen in Fig. 11.2. Of particular note is the ease with which signatures have been generated for the two elongated classes of tracks and roads. Although only the mean vectors are shown here, the clustering processor in MultiSpec also generates the covariance matrices that are used in the subsequent classification step.

The infrared versus red bispectral plot in Fig. 11.2 shows information class labels attached to the cluster means. As observed, there are two (spectral) classes of grass, and two of sparse vegetation. There is also a thin border class of mixed vegetation and water pixels; that often happens in practice and would not get picked up in a traditional supervised classification exercise. Here it will be attributed to the water class.
Figure 11.3 shows the result of a maximum likelihood classification using the full set of signatures for all 10 clusters. Implicitly, that means we are using two spectral classes in the case of the grass information class and two spectral classes in the case of the sparse vegetation information class. In the classification map they have been coloured the same since the end user is not interested in the spectral class structure.

11.5 Cluster Space Classification

Apart from the possibility of mixed pixel classes on information class boundaries, an assumption in the hybrid classification methodology just treated is that the spectral classes, by and large, map to single information classes. In other words, the overlap of a single spectral class into several information classes is assumed not to occur. When it is recalled that the spectral classes are a convenient segmentation of data in the spectral space, most likely generated by an unsupervised approach such as clustering, and that the information classes are simply user-defined labels that refer to regions on the ground, it is likely that the spectral classes and information classes might overlap substantially as illustrated in Fig. 9.8. Such a situation can be handled statistically, as a generalisation of the hybrid approach of Sect. 11.4, as shown in Sect. 9.13.
11.6 Supervised Classification Using the Support Vector Machine

11.6.1 Initial Choices

When seeking to apply the support vector classifier in thematic mapping the user needs to make two initial decisions: which kernel to use to improve separability, and what multiclass strategy to adopt.

Generally a polynomial or radial basis function kernel is chosen, as seen in Sect. 8.16. Although many other possibilities exist, they are the two most often found in remote sensing studies and are known to work well. In this section we will focus on the radial basis function kernel of (8.54), which we reproduce here

\[ k(x_i, x) = \exp \left\{ -\gamma \|x - x_i\|^2 \right\} \]  

(8.54)

As noted, it has one parameter \( \gamma \), a value for which has to be estimated to generate optimum results from the support vector classifier. We will come to that below. Similar considerations will apply if a polynomial kernel is chosen.

Section 8.17 presents several multiclass methods that allow the fundamentally binary support vector machine to work with many classes. We will choose the one-against-one (OAO) approach here for illustration. It requires the user to generate \( M(M - 1)/2 \) binary classifiers, one for each pair of the \( M \) information classes of interest to the user. Ideally that list should be exhaustive. Otherwise unseen pixels not represented among the training sets will nevertheless be labelled into one of those classes unless some measure of typicality is applied to the decision rule test in the parallel network of Fig. 8.16.

Clearly, the user also needs training data for each information class, gathered in the manner discussed in Sect. 11.3.2.

There are two central SVM equations that we now need to focus on, again reproduced here from Chap. 8. The first is the objective function

\[ \frac{1}{2} ||w||^2 + C \sum_i \xi_i \]  

(8.43b)

and the other is the decision rule, shown here with the kernel substituted

\[ \text{sgn}\left\{ \phi(w)^T\phi(x) + w_{N+1} \right\} = \text{sgn}\left\{ \sum_{i \in S} \alpha_i y_i k(x_i, x) + w_{N+1} \right\} \]  

(8.50)

We seek to minimise (8.43b): ||w|| controls the margin in Fig. 8.14 and \( C \) controls the influence of the slack variables \( \xi_i \), that permit some classification error noting, that even after kernel transformation complete, linear separability might not be possible because of a scattering of pixels on the wrong sides of the separating hyperplane. The value chosen for \( C \) determines the misclassification error that can be accepted.
Thus, there are two parameters to be found before the SVM can be applied: the regularisation parameter $C$, and the width parameter $\gamma$ in the kernel. Unfortunately, they are interdependent, so one cannot be found in isolation from the other.

### 11.6.2 Grid Searching for Parameter Determination

The best values of $C$ and $\gamma$ will vary from data set to data set and thus have to be determined anew for each classification exercise. Unfortunately they can vary over a wide range, particularly $C$, so an efficient search strategy is needed. A grid searching process is usually selected in which an initial large range of values for each parameter is chosen and the ranges discretised to give a matrix of $C$, $\gamma$ pairs. The SVM is trained on a representative set of data using each pair selected in turn, from which the best pair is selected. The grid spacing can then be narrowed in the vicinity of that pair and the process repeated, allowing more effective values to be found. This process should be used to find the best parameter pair for each binary classifier in the multiclass topology adopted, but is sometimes done for the multiclass data set as a whole.

Once the user is satisfied with those values then the OAO multiclass network of SVMs can be fully trained. During training the final set of SVMs for each binary separation, the corresponding support vectors, and corresponding values for the coefficients $a_i$ in (8.50) will be determined with the optimisation software used.\(^\text{11}\) Once they are available the decision rule in (8.50) is then fully specified and can be used to label unseen pixels.

### 11.6.3 Data Centering and Scaling

Some, although not all, authors recommend that the data be shifted to a mean of zero and scaled to a common range in each band before training, as a means for improving performance and to speed up the optimisation step.\(^\text{12}\)

---

\(^{11}\) Several packages are available for the support vector classifier, including LibSVM, described in Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, vol. 2, issue 3, 2011, pp. 27:1–27:27. The software is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm Commercial image analysis packages also include SVM classifiers.

11.7 Assessing Classification Accuracy

At the completion of a classification exercise the accuracy of the results obtained needs to be assessed. That is necessary to allow confidence to be attached to the results and will serve to indicate whether the objectives of the analysis have been achieved. There are several ways of assessing the accuracy of a thematic map, and there are several measures available for expressing that accuracy. In this section we consider the most common.

11.7.1 Use of a Testing Set of Pixels

One approach for assessing map accuracy is to select an independent random sample of pixels from the thematic map and check their labels against actual ground classes determined from reference data. That presents practical challenges because we then have to locate labelled ground reference data for those pixels randomly selected from the thematic map.

More often the analyst has set aside labelled data to be used as a testing set after the classification has been carried out. This is similar to the training set of pixels used to generate the classifier in the first instance. In most cases, the analyst labels as many pixels as practicable and then uses a subset for training and another subset for assessing the accuracy of the final product.

In principle, the testing data should be composed of pixels selected at random in order to avoid the interdependences of near neighbouring pixels. A difficulty that can arise with random sampling in this manner is that it is area-weighted. That is, large classes tend to be represented by a larger number of sample points than smaller classes; indeed it is possible that some small classes may not be represented at all. To avoid the undesirable effect that that has on the assessment of accuracy of the smaller classes, it is necessary to ensure that those classes are represented adequately. An approach that is widely adopted is stratified random sampling in which the user first decides on a set of strata into which the image is divided. Random sampling is then carried out within each stratum. The strata could be any convenient area segmentation of the thematic map, such as grid cells. However, the most appropriate stratification to use is the actual thematic classes themselves. That effectively reduces any area bias in the random sampling that could lead to inappropriate accuracy estimation for the smaller classes.

---

13 In the past, reference data was often called ground truth; that term is now less often used because sampling and interpretation inaccuracies can lead to errors in what might otherwise be regarded as a perfectly correct understanding of the ground labels. The term reference data is less dogmatic although we still tend to assume it is exact, unless we have reasons to believe differently.
11.7 Assessing Classification Accuracy

Table 11.2 Using an error matrix to summarise classifier performance and map accuracy, following the layout of Congalton and Green\textsuperscript{14}

<table>
<thead>
<tr>
<th>reference data classes</th>
<th>thematic map classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>35</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>sum</td>
<td>50</td>
</tr>
</tbody>
</table>

overall accuracy \(=\frac{35+37+41}{136} = 83.1\%\)

<table>
<thead>
<tr>
<th>producer’s accuracies</th>
<th>user’s accuracies</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (=\frac{35}{50} = 70.0%)</td>
<td>A (=\frac{35}{39} = 89.7%)</td>
</tr>
<tr>
<td>B (=\frac{37}{40} = 92.5%)</td>
<td>B (=\frac{37}{50} = 74.0%)</td>
</tr>
<tr>
<td>C (=\frac{41}{46} = 89.1%)</td>
<td>C (=\frac{41}{47} = 87.2%)</td>
</tr>
</tbody>
</table>

11.7.2 The Error Matrix

Whichever of the approaches in Sect. 11.7.1 is used, it is common to express the results in the form of an error matrix (sometimes in the past called a contingency matrix or a confusion matrix) which lists the reference data classes by column and the classes indicated on the thematic map by row, as shown in Table 11.2. The cells in the table show the number of pixels that are common between a reference class and a map class. In an ideal result the table or matrix will be diagonal, indicating that for every reference class pixel the classifier generated the correct label. For a poor classification the off diagonal terms will be larger indicating that the classifier has had trouble correctly labelling the pixels from the reference data.

The column sums in the error matrix represent the total number of labelled reference pixels available per class. The row sums represent the total number of pixels labelled by the classifier as coming from a particular class in the set of pixels chosen to assess classification accuracy. Using those, we can define errors of omission and errors of commission. Errors of omission correspond to those pixels belonging to the reference class that the classifier has failed to recognise: they are therefore the off diagonal terms down the column for a particular reference class. We can turn them into percentages by dividing the counts in those cells by the column sum. Errors of commission correspond to those pixels belonging to other reference classes that the classifier has placed in the class of interest: they are the off diagonal terms across the row for a particular thematic map class. They can be turned into percentages by dividing the counts in those cells by the row sum.

When interpreting an error matrix it is important to understand that different indications of accuracy will result according to whether the number of correct pixels for a class (those on the diagonal) is divided by the total number of reference pixels for that class (the column sum) or the total number of pixels that the classifier attributes to the class (the row sum). For example, consider class B in Table 11.2. As noted, 37 of the reference data pixels have been correctly labelled. This represents 37/40 = 92.5% of the reference data pixels for the class. This is the probability that the classifier has labelled a pixel as class B given that the actual (reference data) class is B. It is often referred to as producer’s accuracy\footnote{Ibid.} and is an indication of classifier performance. A user of a thematic map produced by a classifier is often more interested in the likelihood that the actual class is B given that the pixel has been labelled B on the thematic map by the classifier; this is an indication of map accuracy. It is called user’s accuracy and, for this example, is 37/50 = 74.0%, indicating that only 74% of the pixels labelled B on the thematic map are correct, even though the classifier correctly handled almost 93% of the class B reference data. This distinction is important and leads one to believe that user’s accuracy is the figure that should most often be adopted. Producer’s and User’s accuracies for all of the classes are indicated in Table 11.2.

Often the results of a classification exercise are expressed as a single figure of accuracy, independent of the class. In the case of the data in Table 11.2 we would say that the result is 113/136 = 83.1% accurate. While that is an acceptable summary of the situation it masks the fact that the classifier may handle some classes better than others. Knowing how the classifier performs on individual classes is important if the analyst is going to iterate through the results in order to refine the accuracy of the final product.

### 11.7.3 Quantifying the Error Matrix

In Sect. 11.7.1 we noted two alternative methods that could be used to assess how a classifier performs and how accurate a map might be. In the first, samples are taken from the thematic map and checked against the ‘true’ labels on the ground. In the second, samples are taken of ground gathered reference data (a testing set) and used to check the classifier generated map labels. The first checks the accuracy of the map and the second checks the performance of the classifier—they are not the same.\footnote{This section is based on J.A. Richards, Classifier performance and map accuracy, Remote Sensing of Environment, vol. 57, 1996, pp. 161–166.} The first is what the user wants but the second is easier in practice and generally is what is available when constructing the error matrix. The latter is equivalent to producer’s accuracy but the former is only equivalent to user’s
accuracy when the distribution of testing pixels over the classes reflects the actual
class distributions on the ground, as the following analysis shows. This is not
widely appreciated.

Suppose we use the variates $t, r$ and $m$ to represent respectively the true labels for
a pixel, labels in the reference data and labels in the thematic map; we will use the
symbols $Y$ and $Z$ to represent any of the available class labels, i.e. $Y, Z \in \{A, B, C\}$.

If we sample the map and check the accuracy of those samples on the ground
then effectively we are estimating the map accuracy probabilities: $p(t = Z|m = Z)$
This is the likelihood that the actual label for a pixel is $Z$ if the map shows it as $Z$,
which is what the user of the thematic map is interested in. More generally,
$p(t = Y|m = Z)$ is the probability that $Y$ is the correct class if the map shows $Z$.

If, instead, we select reference pixels and check the corresponding map labels
then we are computing the classifier performance probabilities $p(m = Z|r = Z)$, or
in general $p(m = Z|r = Y)$, which is the likelihood that the map label is $Z$ for a
pixel labelled as $Y$ in the reference data.

We have used the two different variates $t$ and $r$ to refer to the ground labels only
because, in the case of checking the map by sampling from it, there is strictly no
concept of reference data. We now assume they are the same, but that places an
important constraint on the reference data set—i.e. the testing set. Its labelled pixels
must be truly representative of the situation on the ground and, in particular, the
distribution of pixels per class must be representative of their distribution on the
region of the earth’s surface being imaged. If that is the case then random selections
from the testing set will yield a distribution by label the same as the prior probability
of occurrence of the labels on the ground. If we make that assumption then we can put

\[ p(r = Y|m = Z) \equiv p(t = Y|m = Z) \]

and then use Bayes Theorem to relate the map accuracy and classifier performance
probabilities

\[ p(r = Y|m = Z) = \frac{p(m = Z|r = Y)p(r = Y)}{p(m = Z)} \]

In this last expression the prior probability $p(r = Y)$ represents the likelihood that
class $Y$ exists in the region being imaged; $p(m = Z)$ is the probability that class $Z$
appears on the thematic map, which can also be generated from

\[ p(m = Z) = \sum_{Y \in \{A,B,C\}} p(m = Z|r = Y)p(r = Y) \]

To reiterate, the $p(r = Y|m = Z)$ are what we are interested in because they tell us
how accurately the map classes represent what’s on the ground, but that requires
the rather impractical step of identifying samples selected at random from the map
for identification. In contrast, using previously labelled testing data we can find the
$p(m = Z|r = Y)$, but that doesn’t explicitly tell us about the accuracy of the map
product. The concept of user’s accuracy in the previous section is meant to be a
surrogate, but that is only true if the testing set distribution reflects the relative proportions of classes on the ground as we will now show by example, using the data of Table 11.2.

Table 11.3 gives three sets of results for differing prior probability distributions for the ground class labels; here the results are left in the form of probabilities and not converted to percentages. In one, the distribution is the same as the proportion of classes in the testing set; in another equal prior probabilities are assumed; and in the third a distribution very different from the priors is used. As seen, the map accuracies are only the same as the user’s accuracies when the testing set reflects the priors.

In Table 11.3 a number of average accuracies are reported. We need now to give some thought as to how the average is computed. In its ideal form the class-wise accuracy of a thematic map is expressed by the probabilities $p(r = Z|m = Z)$. As above, these are found by sampling the thematic classes on a map and seeing how many pixels correspond to the actual classes on the ground. Once those probabilities have been determined the average accuracy of the map should be expressed as

$$\text{map accuracy} = \sum_{Z \in \{A,B,C\}} p(r = Z|m = Z)p(m = Z)$$

in which the class-wise accuracies are weighted by the probability of occurrence of those classes in the thematic map. That is important to ensure that the user of a map is not misled by the disparity in the sizes of classes that might appear on a map. For example, even though a class which has a small area is highly accurate, it will not have a significant influence on the average accuracy, and certainly not as much influence as the larger classes. Using the reciprocity of joint probabilities we can express the above formula for map accuracy in the form

<table>
<thead>
<tr>
<th>Prior probabilities</th>
<th>Classifier performance</th>
<th>User’s accuracies</th>
<th>Map accuracies</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(A)$ = 0.368</td>
<td>$p(m = A</td>
<td>r = A) = 0.700$</td>
<td>Class A: 0.987</td>
</tr>
<tr>
<td>$p(B)$ = 0.294</td>
<td>$p(m = B</td>
<td>r = B) = 0.925$</td>
<td>Class B: 0.740</td>
</tr>
<tr>
<td>$p(C)$ = 0.338</td>
<td>$p(m = C</td>
<td>r = C) = 0.981$</td>
<td>Class C: 0.872</td>
</tr>
</tbody>
</table>

Average = 0.831

<table>
<thead>
<tr>
<th>Prior probabilities</th>
<th>Classifier performance</th>
<th>User’s accuracies</th>
<th>Map accuracies</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(A)$ = 0.333</td>
<td>$p(m = A</td>
<td>r = A) = 0.700$</td>
<td>Class A: 0.987</td>
</tr>
<tr>
<td>$p(B)$ = 0.333</td>
<td>$p(m = B</td>
<td>r = B) = 0.925$</td>
<td>Class B: 0.740</td>
</tr>
<tr>
<td>$p(C)$ = 0.333</td>
<td>$p(m = C</td>
<td>r = C) = 0.981$</td>
<td>Class C: 0.872</td>
</tr>
</tbody>
</table>

Average = 0.831

<table>
<thead>
<tr>
<th>Prior probabilities</th>
<th>Classifier performance</th>
<th>User’s accuracies</th>
<th>Map accuracies</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(A)$ = 0.900</td>
<td>$p(m = A</td>
<td>r = A) = 0.700$</td>
<td>Class A: 0.987</td>
</tr>
<tr>
<td>$p(B)$ = 0.050</td>
<td>$p(m = B</td>
<td>r = B) = 0.925$</td>
<td>Class B: 0.740</td>
</tr>
<tr>
<td>$p(C)$ = 0.050</td>
<td>$p(m = C</td>
<td>r = C) = 0.981$</td>
<td>Class C: 0.872</td>
</tr>
</tbody>
</table>

Average = 0.831

Average = 0.721
map accuracy $= \sum_{Z \in \{A,B,C\}} p(r = Z, m = Z)\\
= \sum_{Z \in \{A,B,C\}} p(m = Z, r = Z)\\
= \sum_{Z \in \{A,B,C\}} p(m = Z | r = Z) p(r = Z)\\$

This last expression tells us that the true average map accuracy can be obtained from the classifier performance, provided the classifier probabilities are weighted by the prior probabilities.

There is one final lesson we need to learn from the entries in Table 11.3. The last example shows a case with extreme prior probabilities, which is indicative of a situation where we may have one large class and two smaller classes. The way this example has been set up is such that the classifier performs better on the smaller classes than it does on the large class. When the map accuracies are computed it is seen that the results for the two smaller classes are disappointingly low notwithstanding that the classifier worked well on those classes. The reason is that the errors the classifier made on the large class commit themselves to the smaller classes thereby causing the confusion. In practice, therefore, even though one might be interested in a set of smaller classes and be happy that a classifier performs well on them, it is nevertheless important that any larger classes are also well recognised by the classifier so that errors of commission do not distort the results for the smaller classes.

### 11.7.4 The Kappa Coefficient

The Kappa Coefficient is a measure of classifier performance derived from the error matrix but which, purportedly, is free of any bias resulting from chance agreement between the classifier output and the reference data.\(^{17}\) It was proposed initially for checking the chance agreement between the results of two independent classifiers. In our case, we have only one classifier and a set of reference data but, nevertheless, the reference data is also regarded as a sample of the true situation on the ground for this purpose. Although, as discussed below, there is dispute over the efficacy of the method, it is nevertheless, rightly or wrongly, widely used in remote sensing to report classification results. We will demonstrate its use by reference to

\(^{17}\) The Kappa Coefficient is generally attributed to J. Cohen, A coefficient of agreement for nominal scales, *Educational and Psychological Measurement*, vol. 20, no. 1, 1960, pp. 37–46, although apparently its use has been traced back to the late 1800s. Its use in remote sensing is covered extensively in R.G. Congalton and K. Green, *loc. cit.*
the error matrix in Table 11.2. If we look at the classifier output in Table 11.2 we see that

- The classifier places:
  - 39/136 = 0.287 of the pixels in class A
  - 50/136 = 0.368 of the pixels in class B
  - 47/136 = 0.346 of the pixels in class C

- The reference data places:
  - 50/136 = 0.368 of the pixels in class A
  - 40/136 = 0.294 of the pixels in class B
  - 46/136 = 0.338 of the pixels in class C

The probability that they would both place a pixel at random in class A is $0.287 \times 0.368 = 0.106$; similarly the probability that they would both place a pixel in class B is $0.368 \times 0.294 = 0.108$, and in C is $0.346 \times 0.338 = 0.117$. Overall, the probability that they place a pixel at random in the same class is the sum of the three probabilities, viz. $0.106 + 0.108 + 0.117 = 0.331$, which is the random chance of their agreeing on the label for a pixel. On the other hand, the probability of a correct classification determined from the agreement of the classifier output and the reference data is $(35 + 37 + 41)/136 = 0.831$.

Now, the Kappa Coefficient is defined, in words, as

$$\kappa = \frac{\text{probability of correct classification} - \text{probability of chance agreement}}{1 - \text{probability of chance agreement}}$$

which, for this example, is $\kappa = (0.831 - 0.331)/(1 - 0.331) = 0.747$. We now need to do two things: express the Kappa Coefficient directly in terms of the elements of the error matrix, and gain some understanding of what certain levels of Kappa Coefficient mean.

The commonly used measure of the probability of correct classification is given by the sum of the diagonal elements of the error matrix divided by the global total. If we represent an entry in the error matrix by $n_{ij}$, the total number of pixels by $N$, and the number of classes by $M$ then the probability of correct classification is

$$p_o = \frac{1}{N} \sum_{i=1}^{M} n_{ii} \quad (11.1)$$

If we now define the sum over the rows of the error matrix and the sum over the columns, respectively, as

$$n_{+i} = \sum_{k=1}^{M} n_{ki} \quad n_{i+} = \sum_{k=1}^{M} n_{ik} \quad (11.2a)$$

then the probabilities that the reference data and classifier respectively place a pixel at random into class $i$ are

---

18 See Cohen, loc. cit.
\[ p_{+i} = \frac{1}{N} \sum_{k=1}^{M} n_{ki} \quad p_{i+} = \frac{1}{N} \sum_{k=1}^{M} n_{ik} \]  

(11.2b)

so that the probability of their agreeing by chance on any of the available labels for a pixel is

\[ p_c = \sum_{i=1}^{M} \left\{ \frac{1}{N} \sum_{k=1}^{M} n_{ki} \frac{1}{N} \sum_{k=1}^{M} n_{ik} \right\} \]  

(11.2c)

By definition

\[ \kappa = \frac{p_o - p_c}{1 - p_c} \]  

(11.3a)

This is the ratio of (i) the agreement between the map and reference data, expressed as classification accuracy, and the chance agreement (sometimes called the beyond-chance agreement) and (ii) the probability that there is no chance agreement between the map and the reference data. Said another way, it is the proportion of labels that are in agreement between the map and reference data after chance agreement is excluded. Substituting (11.1) and (11.2) into (11.3) and multiplying throughout by \( N^2 \) gives

\[ \kappa = \frac{N \sum_{i=1}^{M} n_{ii} - \sum_{i=1}^{M} \left\{ \sum_{k=1}^{M} n_{ki} \sum_{k=1}^{M} n_{ik} \right\} }{N^2 - \sum_{i=1}^{M} \left\{ \sum_{k=1}^{M} n_{ki} \sum_{k=1}^{M} n_{ik} \right\} } \]  

(11.3b)

which is sometimes written as

\[ \kappa = \frac{N \sum_{i=1}^{M} n_{ii} - \sum_{i=1}^{M} n_{+i} n_{i+} }{N^2 - \sum_{i=1}^{M} n_{+i} n_{i+} } \]  

(11.3c)

We now turn our attention to the values of Kappa. How does the user of a thematic map, with accuracy assessed on the basis of the Kappa Coefficient, know if the result is good or not? That is not an easy question to answer in general and is one reason why some analysts still prefer to use classification accuracy, or even a presentation of the full error matrix, in favour of using Kappa. Based on empirical results over several authors the guidelines on Kappa in Table 11.4 have been proposed, noting that its theoretical maximum is 1 (in the ideal case when there can be no chance agreement and there are no off diagonal elements in the error matrix) and its minimum can be large and negative.

There is now significant concern that the Kappa Coefficient is a misleading measure because (i) in an endeavour to be a single measure it masks importantly-different sources of error, (ii) its dependence on a comparison with chance agreement.

---

agreement is not informative and (iii) being a ratio, similar values can be generated by differing combinations of the numerator and denominator. A simple illustration of this last point is shown in Fig. 11.4, which is a plot of (11.3a) for ranges of $p_0$ and $p_c$. Note the different combinations that correspond to the same value of Kappa, such as at 0.5.

As a result of these concerns a return to measures more directly related to the entries in the error matrix has been advocated. Two recently recommended measures are quantity disagreement and allocation disagreement. To derive these measures we note that the expressions in (11.2b) can be written

$$p_{+i} = \sum_{k=1}^{M} p_{ki} \quad p_{i+} = \sum_{k=1}^{M} p_{ik}$$

### Table 11.4 Suggested ranges for the Kappa Coefficient

<table>
<thead>
<tr>
<th>Kappa Coefficient</th>
<th>Classification can be regarded as</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 0.4</td>
<td>Poor</td>
</tr>
<tr>
<td>0.41–0.60</td>
<td>Moderate</td>
</tr>
<tr>
<td>0.61–0.75</td>
<td>Good</td>
</tr>
<tr>
<td>0.76–0.80</td>
<td>Excellent</td>
</tr>
<tr>
<td>0.81 and above</td>
<td>Almost perfect</td>
</tr>
</tbody>
</table>

**Fig. 11.4** Kappa coefficient as a function of the probability of correct classification (in the legend) and probability of chance agreement

---

20 *ibid.*
in which the \( p_{ij} \) are the error matrix entries expressed as proportions or probabilities; \( p_{ij} \) is shorthand for the joint occurrence \( p(m = i, r = j) \) when the relative class proportions in the reference data reflect the true proportions on the ground. Taking the difference between \( p_{+i} \) and \( p_{i+} \) is tantamount to the difference between the number of class \( i \) pixels in the reference data and the number in the thematic map. Summing the absolute difference over all classes gives the discrepancy in the proportions of all classes between the reference data and the map\(^{21}\):

\[
Q = \frac{1}{2} \sum_{i=1}^{M} |p_{+i} - p_{i+}|
\]  

(11.4a)

That is called the *quantity disagreement*. As another interpretation it effectively measures the differences in the areas allocated to the classes in the reference data and map—it is one error measure. A second error measure is the *allocation disagreement* defined by\(^{22}\)

\[
A = \sum_{i=1}^{M} \min\{(p_{+i} - p_{ii}), (p_{i+} - p_{ii})\}
\]  

(11.4b)

The first of the arguments in the minimum function is the proportion of class \( i \) pixels in error in the map—the errors of commission, whereas the second is the proportion indicated from the reference data as errors of omission. This measure is intended to assess the aggregated misallocation of individual pixels for the same level of quantity agreement. If there is a specific error of commission there is a corresponding error of omission, which is recognised in the minimum operation of (11.4b).

From (11.1) \( 1 - p_o \) is the total error, or disagreement between the thematic map and the reference data. Interestingly it can be shown that\(^{23}\)

\[1 - p_o = A + Q\]  

(11.4c)

indicating that the total disagreement can be disaggregated into quantity and allocation disagreements.

### 11.7.5 Number of Testing Samples Required for Assessing Map Accuracy

We now turn to the question of how many testing pixels should be chosen to ensure that the assessed accuracy of the thematic map is a reliable estimate of the real accuracy of the map. Clearly, choosing too few samples per class will lead to a

---

\(^{21}\) The \( \frac{1}{2} \) is needed because operations involving both the row sums and columns sums will have a maximum of \( 2N \).

\(^{22}\) Pontius and Millones, *loc. cit.*

\(^{23}\) *ibid.*
poor estimate of map accuracy. To illustrate this point a single testing sample from a particular class can only indicate an accuracy of 100 or 0% depending on its match, or otherwise, to the reference data. A larger sample will clearly give a more realistic estimate.

The problem we have to consider is shown in Fig. 11.5. In Fig 11.5a we have the pixels in a thematic map indicated as correctly (white) and incorrectly (grey) labelled. Of course we don’t know which are correct and incorrect a priori; that’s what we have to find using the labelled testing data. Figure 11.5b shows the locations of a random set of testing pixels, and Fig. 11.5c shows the testing pixels sampling the map. Note that some pixels in error have been detected while others have been missed. What we need to do is choose enough testing pixels so that the correct and incorrect labels sampled on the map are sufficient to reveal the map’s accuracy to the precision required by the user.

\[ y_i \quad i = 1 \ldots N \]

\[ g_i \quad i = 1 \ldots n \]
If we select a pixel at random from the thematic map it can only be correct or incorrect. The chance of it being one or the other is described by a binomial probability distribution with properties established by the numbers of correct and incorrect labels in the map. The likelihood of our choosing a correctly labelled pixel will be high if there are few pixels in error in the map, and vice versa.

For the thematic map we let $N$ be the total number of pixels, and $y_i, i = 1 \ldots N$ be a property of the $i$th pixel, which has the value 1 if the pixel is correctly labelled and 0 if it is incorrectly labelled.

Thus the value of the sum
\[ \sum_{i=1}^{N} y_i \]

is the number of correctly labelled pixels, and
\[ P = \frac{1}{N} \sum_{i=1}^{N} y_i \quad (11.5) \]
is the proportion of pixels that are correctly labelled. This is the overall accuracy of the map\(^{24}\), which is what we want to determine using the testing pixels. It is also the mean of the binomial distribution describing the correct labels in the map.

For the testing set let $n$ be the total number of pixels, and $g_j, j = 1 \ldots n$ be the property of the $j$th testing set pixel which has the value 1 if a correctly labelled pixel is detected in the map and 0 if an incorrectly labelled pixel is found. Note $n \leq N$.

Thus the value of the sum
\[ \sum_{j=1}^{n} g_j \]

is the number of correctly labelled map pixels found by using the testing set and
\[ p = \frac{1}{n} \sum_{j=1}^{n} g_j \quad (11.6) \]
is the proportion of pixels that are correctly labelled in the testing set and is thus an estimate of the accuracy of the map. Our task is to find the value of $n$ that makes $p$ in (11.6) an acceptable estimate of $P$ in (11.5).

The sum of the binomial random variables in (11.6) is itself a variate, as is $p$. Since the individual random variables come from the same underlying binomial distribution it can be shown that their expected value is the mean of that distribution, so that

\(^{24}\) Here the accuracy is described by a proportion between 0 and 1, rather than the more usual form between 0 and 100%.
The variate $p$ also has a standard deviation that tells us the range about the mean $P$ within which the sample value $p$ is likely to occur. Since we are sampling a set of $n$ pixels from a finite number of pixels $N$ in the thematic map the variance of $p$ about its mean $P$ is given by

$$\text{var}(p) = \frac{P(1-P)}{n} \frac{(N-n)}{(N-1)}$$

(11.7)

If we want confidence in the value of $P$ estimated from $p$ then we need to ensure this variance is small. Before proceeding to do that, there are a couple of useful observations we can make from (11.7). First, if we just use one testing pixel then the variance is $P(1-P)$ which is in the range $(0,0.25)$. We can conclude nothing about the map from the outcome of the single test, unless the map were near perfect ($P \approx 1$) or near imperfect ($P \approx 0$). That is the situation alluded to in the opening paragraph of this section. Secondly, if $n = N$, i.e. we test every pixel in the thematic map, then the variance in the estimate $p$ is zero, meaning that $p$ is exactly $P$, which is logical. Thirdly, if $N \gg n$, which is generally the case in remote sensing—i.e. the number of testing pixels is generally a small part of the overall scene, then (11.7) reduces to

$$\text{var}(p) = \sigma^2 = \frac{P(1-P)}{n}$$

(11.8)

The factor $\frac{(N-n)}{(N-1)}$ between (11.7) and (11.8) is sometimes called the finite population correction.

---

To a very good approximation\textsuperscript{26} the variate \( p \) can be assumed to be normally distributed about its mean \( P \), as illustrated in Fig. 11.6, which shows graphically the range within which the estimate we generate from testing data is likely to occur. Note that 95% of the time our estimate is within \( \pm 1.96 \), or approximately \( \pm 2 \), standard deviations of the mean; so with 95% confidence we can say the sampled map accuracy is in that range. If we were happy with a lower precision then we can give a smaller range for the estimate of the map accuracy.

We are now in the position to use (11.8) to estimate the number of testing pixels \( n \) needed to check the accuracy of a thematic map. We can be 95% confident it lies within two standard deviations of its true value. We now have to consider what range about the true value we are prepared to accept as an error, because that range specifies the value of the standard deviation. Suppose we are happy for the estimate to be within \( \pm e \) of what we think is the true map accuracy; i.e. \( p = P \pm e \). Then from (11.8) we have at the 95% confidence level and for \( N \gg n \)

\[
2\sqrt{\frac{P(1-P)}{n}} = e
\]

Rearranging gives as the minimum number of samples required

\[
n = \frac{4P(1-P)}{e^2}
\] \hspace{1cm} (11.9)

As an illustration, suppose we are happy for an estimate to be within \( \pm 0.04 \) of a true proportion which is thought to be about 0.85 (i.e. we are happy if the map accuracy estimated from the testing set is in the range 0.81–0.89) then from (11.9) at the 95% confidence level \( n = 319 \).

Thus randomly selecting 319 testing pixels will allow a thematic map accuracy of about 85% to be checked with an uncertainty of \( \pm 4\% \) with 95% confidence. Note that (11.7) and (11.8) can be used in percentage terms as well as proportions since we obtain percentages by multiplying proportions by 100; that applies also to the standard deviation, because it refers to a proportion or percentage as appropriate.

Table 11.5 gives further examples for a range of likely thematic map accuracies, this time expressed in percentages.

We can make a couple of observations from (11.8) to help guide in the selection of the number of testing pixels. First, more samples will narrow the variance about the mean; however, since the variance is inversely proportional to \( n \) there is a diminishing value in choosing many more samples than the minimum required. Secondly, the term \( P(1-P) \) is largest in the vicinity of \( P = 0.5 \). Thus the error (variance) is greatest when the accuracy of the map is poor, and correspondingly more testing samples are required, than when the overall thematic map accuracy is high.

\textsuperscript{26} ibid.
The results given above, and summarised in Table 11.5, have been developed by assuming that we are testing the overall accuracy of the thematic map, independent of class. We have made no specific reference to class or otherwise in this development so the results apply equally at the category level if required. If there were a large number of classes in a thematic map then clearly the requirements of Table 11.5 will lead to an excessively large number of testing pixels. Table 11.6 shows the number of testing pixels required for an individual class, in this case with 10% uncertainty in the accuracy of the class.27

<table>
<thead>
<tr>
<th>Thematic map accuracy (%)</th>
<th>Number of testing pixels required</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>525</td>
</tr>
<tr>
<td>75</td>
<td>469</td>
</tr>
<tr>
<td>80</td>
<td>400</td>
</tr>
<tr>
<td>85</td>
<td>319</td>
</tr>
<tr>
<td>90</td>
<td>225</td>
</tr>
<tr>
<td>95</td>
<td>119</td>
</tr>
</tbody>
</table>

The calculations of the previous section were focused on the need to establish a sufficient number of testing pixels so that the overall accuracy of a thematic map could be assessed. Those calculations had their foundation in binomial statistics since, for each pixel in a thematic map, only two outcomes are possible—a correctly labelled pixel or an incorrectly labelled pixel. With reference to Table 11.2, those results are sufficient for evaluating the sum of the diagonal entries compared with the total number of testing pixels, but not sufficient for

<table>
<thead>
<tr>
<th>Table 11.5 Number of testing pixels required for evaluating thematic map accuracies, with an error ±4% and with 95% confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thematic map accuracy (%)</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>70</td>
</tr>
<tr>
<td>75</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>90</td>
</tr>
<tr>
<td>95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thematic class accuracy (%)</th>
<th>Number of testing pixels required</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>45</td>
</tr>
<tr>
<td>75</td>
<td>40</td>
</tr>
<tr>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td>85</td>
<td>19</td>
</tr>
</tbody>
</table>


### 11.7.6 Number of Testing Samples Required for Populating the Error Matrix

The calculations of the previous section were focused on the need to establish a sufficient number of testing pixels so that the overall accuracy of a thematic map could be assessed. Those calculations had their foundation in binomial statistics since, for each pixel in a thematic map, only two outcomes are possible—a correctly labelled pixel or an incorrectly labelled pixel. With reference to Table 11.2, those results are sufficient for evaluating the sum of the diagonal entries compared with the total number of testing pixels, but not sufficient for
generating accurate estimates of the individual entries in the error matrix simultaneously.

When our interest is in the performance of a classifier when labelling a pixel into each of the available classes, testing data needs to be used to estimate the proportions of pixels by class. That means we need good estimates of all of the elements of the error matrix and, as would be expected, this will require more testing pixels.

Perhaps the simplest approach, particularly when stratified random sampling is used to avoid area bias in the results, is to choose samples within each class, using the guidelines in Table 11.6, although perhaps with a tighter error bound.

Another approach is to use the multinomial probability distribution to describe the multiple outcomes possible with simple, as against stratified, random sampling when using testing data to check the accuracy of the actual class to which a pixel is allocated in the thematic map.28 The results parallel those of the binomial development of the previous section. When the total number of pixels in the thematic map is large we obtain the following estimate of the necessary sample size, based on the tolerable error and expected mean for the $i$th class for the pixel:

$$n = \frac{BP_i(1 - P_i)}{e_i^2}$$

(11.10)

$P_i$ is the population proportion for the class, $e_i$ is the error we can tolerate in the accuracy of the proportion estimate for that class and $B$ is the upper $\beta$ percentile for the $\chi^2$ distribution with one degree of freedom, where $\beta$ is the overall precision needed, divided by the total number of classes.

A version of (11.10) can also be derived for cases when the number of pixels in the thematic map is not large enough to ignore a finite population correction.29

To find a satisfactory value for $n$, (11.10) would be evaluated for each class and the largest value of $n$ selected for the required size of the testing set, in the sense that that is the most demanding requirement. That number could then be divided by the total number of classes to find how many testing pixels per class are needed, noting that this is a simple and not stratified random sampling strategy, which assumes implicitly that the classes are comparable in size (numbers of pixels).

Following our observations with the binomial approach we can see that the worst case class in (11.10) would be one where the population proportion is 0.5. We can therefore derive a conservatively high estimate for $n$ by putting $P_i = 0.5$ in (11.10) to give the simple expression

$$n = \frac{B}{4e^2}$$

(11.11)

in which we have also assumed the same tolerable error for each class.


29 ibid.
As an example\(^{30}\), suppose we require our estimates of the proportions of each of 8 classes in the thematic map to be within the range ±5% and that we want our results to be at the 95% confidence level. Then \(b\) is the upper \(0.05/8 = 0.00625\) (0.625 percentile) of the distribution and has the value 7.568, giving

\[
n = \frac{7.568}{4(0.05)^2} = 757
\]

Thus we need about 757 testing pixels in total, with slightly fewer than 100 per class to get good estimates of the elements of the error matrix at the precision level specified. Although this is based on simple random sampling, we could assume about 100 per class in general, unless it was known beforehand that some classes are very different in size.

11.7.7 Placing Confidence Limits on Assessed Accuracy

Once accuracy has been estimated using testing data it is important to place some confidence on the actual figures derived. If the number of testing pixels has been determined using the guidance of the previous sections then those limits have been set in the process. If not, we can use straightforward statistics to express the interval within which the true map accuracy lies, say, with 95% certainty. That interval can be determined using the expression derived from the normal distribution\(^{31}\)

\[
-z_{a/2} < \frac{x - nP}{\sqrt{nP(1 - P)}} < z_{a/2}
\]

in which \(n\) is the number of testing pixels, as before, \(x(\equiv np)\) is the number that were correctly labelled, and \(P\) is the thematic map accuracy, which we are estimating by \(p = x/n\), \(z_{a/2}\) is the value of the normal distribution beyond which on both tails \(x\) of the population is excluded. As in the previous examples, if we want the normalised statistic \((x - nP)/\sqrt{nP(1 - P)}\) to lie in the 95% portion of the normal curve then \(z_{a/2} = \pm 1.96(\approx \pm 2)\). Using this value, it is a relatively straightforward matter to solve the two inequalities above to show that the extremes of \(P\) estimated by \(p\), at the 95% confidence level are

\[
x + 1.921\pm 1.960\sqrt{x(n-x)/n + 0.960} = 7.842
\]

which for \(n\) and \(x\) large, and for reasonable accuracies, is approximated by

---

\(^{30}\) Taken from Congalton and Green, loc. cit.

\[
\frac{x \pm 1.960 \sqrt{x(n-x)/n + 0.960}}{n} = p \pm \frac{1.960}{n} \sqrt{x(n-x)/n + 0.960}
\]

Choosing an example of 400 testing pixels from Table 11.5 to assess an accuracy of 80% we would expect to find 320 of those pixels in agreement with the map. From (11.12) the bounds on the estimated map accuracy are \( P = p \pm 0.039 \) or, in percentage terms, the map accuracy is between 76 and 84%.

11.7.8 Cross Validation Accuracy Assessment 
and the Leave One Out Method

As an alternative to using a separate testing set of pixels, an effective method for assessing accuracy is based on cross validation. This involves taking a single labelled set of pixels and dividing it into \( k \) separate, equally sized, subsets. One subset is put aside for accuracy testing and the classifier is trained on the pixels in the remaining \( k - 1 \) sets. The process is repeated \( k \) times with each of the \( k \) subsets excluded in rotation. At the end of those \( k \) trials, \( k \) different measures of classification accuracy have been generated. The final classification accuracy is the average of the \( k \) trial outcomes.

A variation of the cross validation method is when each subset consists of a single pixel. In other words, one pixel from the training set is excluded and the classifier trained on the remainder. The pixel which has been excluded is then labelled. In this case there are as many trials as they are training pixels, in each case with a separate pixel left out during training. The average classification accuracy is then the average over the labelling of the pixels left out in each trial. Provided the original training pixels are representative, this method produces an unbiased estimate of classification accuracy.32 This is called the Leave One Out (LOO) method.

11.8 Decision Tree Classifiers

Classifiers such as the maximum likelihood rule and the support vector machine are single stage processes. They make a single decision about a pixel when labelling it as belonging to one of the available classes, or it is left unclassified. Multistage classification techniques are also available, in which a series of decisions is taken to determine the most appropriate label for a pixel. The committee classifiers of Sect. 8.18 are examples.

The most commonly encountered multistage classifier is the decision tree, such as that shown in Fig. 11.7. Decision trees consist of a number of connected classifiers (called decision nodes in the terminology of trees) none of which is expected to perform the complete segmentation of the image data set. Instead, each component classifier only carries out part of the task as indicated. The simplest is the binary decision tree in which each component classifier is expected to perform a segmentation of the data into one or two possible classes or groups of classes. It is the most commonly encountered tree in practice, and has the topologies shown in Fig. 11.8.

There is some ambiguity in the terminology of decision trees; in this treatment we adopt the following:

- **root node**: this is where the tree commences
- **decision node**: intermediate node (and the root node)
terminal or leaf node  final node, which usually represents a single class
link or branch  connection between nodes
tree depth  number of layers from the root node to the most distant leaf
antecedent node immediately above a node of interest; sometimes called a parent node
descendant node immediately following a node of interest; sometime called a child node
split  the result of a decision to create new descendant nodes

The advantages of the decision tree approach are that:

- different sets of features can be used at each decision node; this allows
  - feature subsets to be chosen that optimise segmentations
  - reduced feature subsets at individual decisions, so that the Hughes phenomenon might be avoided
- simpler segmentations than those needed when a decision has to be made among all available labels for a pixel in a single decision
- different algorithms can be used at each decision node
- different data types can be used at each decision node

Decision tree design is usually not straightforward. Sometimes the analyst can design a tree intuitively. For example, near infrared data might be used to segment between land and water bodies; subsequently thermal infrared data might be used to map temperature contours within the water. In principle, tree design involves finding the structure of the tree, choosing the subset of features to be used at each node, and selecting the decision rule to use at each node. If we restrict the range of possibilities for the last two requirements some automated procedures are possible, as developed in the following.

**11.8.1 CART (Classification and Regression Trees)**

The CART tree growing methodology is possibly the most commonly encountered because it restricts, and thus simplifies, the possible options for how the

---

decision nodes function. Only one feature is involved in each decision step and a simple threshold rule is used in making that decision. As with other supervised classification procedures it uses labelled training data to construct the tree. Once the tree has been built it can then be used to label unseen data.

At each node in CART, including at the root node, a decision is made to split the training samples into two groups; the aim is to produce sub-groups that are purer class-wise than in the immediately preceding node. All of the training data from all classes is fed to the root node. We then evaluate all possible binary partitions of the training pixels and choose that partition which minimises the class mixture in the two groups produced. For example, if there were five separate classes in the training set then we would expect the sub-groups to have pixels from fewer than five classes and, in some cases, hope that one sub-group might have pixels from one class only. We keep subdividing the groups as we go down the tree so that ultimately we end up with groups containing pixels from only one class—i.e. “pure” groups. That happens at the leaf nodes.

To be able to implement the process just described we have to have some way of measuring how mixed the training classes are in a particular group. We do that by using an impurity measure, several of which are used in practice. A common metric is the Gini impurity, defined at the $N$th node as

$$i(N) = \sum_{i \neq j} P(\omega_j)P(\omega_i)$$

$$= 1 - \sum_j P(\omega_j)^2$$

(11.13)

in which $P(\omega_j)$ is the fraction of the training pixels at node $N$ that are in class $\omega_j$ and $P(\omega_i)$ is the proportion not in class $\omega_j$. If all the pixels at the node were from a single class then $P(\omega_j) = 1$ and $P(\omega_i) = 0$, for $i \neq j$ so that $i(N) = 0$, indicating no impurity. If there were $N$ equally distributed classes in the training set then $i(N)$ is a maximum and equal to $1 - 1/N^2$, which is larger for larger $N$, as would be expected.

Another impurity measure is based on entropy, defined as

$$i(N) = - \sum_j P(\omega_j)\log_2 P(\omega_j)$$

(11.14)

Again, this is zero if all the training pixels are from the same class, and is large when the group is mixed.

In splitting the training pixels as we go down the tree we are interested in that split which gives the greatest drop in impurity from the antecedent to the descendent nodes—in other words, the split that generates the most pure descendent groups. We can measure the reduction in impurity by subtracting the impurities of the descendent nodes from the impurity of their antecedent node, weighted by the relative proportions of the training pixels in each of the descendent nodes. Let $N$ refer to a node and $N_L$ and $N_R$ refer to its left and right
descendants; let $P_L$ be the proportion of the training pixels from node $N$ that end up in $N_L$. Then the reduction in impurity in splitting $N$ into $N_L$ and $N_R$ is

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$  (11.15)

To see how this is used in building a decision tree consider the training data shown in Fig. 11.9. This consists of three classes, each of which is described by two features (bands). The Gini impurity is used. Table 11.7 shows the original impurity for the complete set of data and the subsequent drops in impurity with various candidates splits. Not all possible splits are given because the number of combinations is excessive; only those that are clearly the most favoured are shown. The table is segmented by successive layers in the decision tree as it is built, showing splits by layer until the leaf nodes are reached. There are several split options later in the tree; only two are given to demonstrate that trees are often not unique but will still segment the data as required. The resulting segmentations of the training set and the corresponding decision trees are shown in Fig. 11.10.

As well as determining which splits should occur at each node, we have to set a value for the feature threshold that tests whether the remaining training patterns should go to the left or right descendent node. The simplest choice for the threshold is the value that lies midway between the nearest neighbours from opposite classes. That position can be weighted according to the relative proportions of the pixels going to each of the descendent nodes. In the example of Fig. 11.10 the simple midpoint threshold has been chosen. Note that a split could just as easily involve decisions based on qualitative attributes as on numerical features; that is another of the benefits of the CART process which makes it potentially valuable in the processing of mixed, GIS-like, data sets.

In practice, we also often have to choose a stopping criterion to avoid the tree becoming too deep and complex. In simple cases, of course, the tree growth stops automatically when all nodes have terminated in leaves. However, we could test the tree at various stages in its growth to see whether it achieves a desired classification accuracy on a separate testing set. Alternatively, a lower bound on the
impurity could be set such that no further tree growth is required if all nodes have achieved that level.

One of the problems with splitting based on the simple thresholding of individual features is that quite complicated trees can be generated compared with what should be possible if more flexibility is introduced into the decision functions and thus the decision boundaries in the spectral space. For example, inspection of Fig. 11.9 suggests that the data could easily be split into the three classes by two inclined linear surfaces, one between class A and B pixels, and the other between class B and C pixels. While it is feasible to develop a tree design methodology that implements linear decisions of that nature at each node, it is sometimes simpler to transform the data prior to tree growth.\textsuperscript{36} For example, if the data of Fig. 11.9 is used to generate its principal components, the principal axes will provide a simpler tree. Figure 11.11 shows the principal axes of the data from Fig. 11.9 along with a decision tree generated with the CART methodology.

Once a tree has been grown it can be examined to see if it can be simplified by pruning; that involves removing nodes or sets of nodes such that the tree is simpler but still gives acceptable accuracy on a testing set of pixels, i.e. so that it still

\begin{table}[ht]
\centering
\caption{Impurity calculations and splits leading to the decision trees of Fig. 11.10 based on the single feature shown in the left descendent column in each case; only the most likely splits are shown to illustrate the process; shaded boxes highlight the greatest reduction in impurity and thus the best splits, noting that two equally favourable splits are possible at the second stage leading to different outcomes and thus illustrating that the resulting tree is not unique}
\small
\begin{tabular}{llllll}
\toprule

Original unsplit training set & A1 A2 A3 B1 B2 B3 C1 C2 C3 & \(i(N) = 0.667\) \\
\midrule

First split candidates & \\

\begin{tabular}{llllll}
\textit{Left descendent} & \textit{Right descendent} & \(i(N_L)\) & \(i(N_R)\) & \(\Delta i(N)\) \\
A1 A2 A3 B1 B2 B3 (\(x_1\)) & C1 C2 C3 (leaf node) & 0.500 & 0 & 0.334 \\
A2 A3 (\(x_2\)) & A1 B1 B2 B3 C1 C2 C3 & 0 & 0.612 & 0.191 \\
C2 C3 (\(x_2\)) & C1 A1 A2 A3 B1 B2 B3 & 0 & 0.612 & 0.191 \\
A1 (\(x_1\)) & A2 A3 B1 B2 B3 C1 C2 C3 & 0 & 0.656 & 0.084 \\
\end{tabular} \\

Second split candidates from A1 A2 A3 B1 B2 B3 | C1 C2 C3 first split & \\

\begin{tabular}{llllll}
B1 B2 (\(x_2\)) & A1 A2 A3 B3 & 0 & 0.375 & 0.250 \\
A2 A3 (\(x_2\)) & A1 B1 B2 B3 & 0 & 0.375 & 0.250 \\
A1 (\(x_1\)) & A2 A3 B1 B2 B3 & 0 & 0.480 & 0.100 \\
\end{tabular} \\

Third split from B1 B2 | A1 A2 A3 B3 second split & \\

\begin{tabular}{llllll}
A1 A3 (\(x_1\)) & A2 B3 & 0 & 0.500 & 0.125 \\
\end{tabular} \\

Fourth split from A1 A3 | A2 B3 third split & \\

\begin{tabular}{llllll}
A2 (leaf node) (\(x_2\)) & B3 & 0 & 0 & 0.500 \\
\end{tabular} \\

Third split from A2 A3 | A1 B1 B2 B3 second split & \\

\begin{tabular}{llllll}
A1 (leaf node) (\(x_1\)) & B1 B2 B3 (leaf node) & 0 & 0 & 0.375 \\
\end{tabular} \\
\bottomrule
\end{tabular}
\end{table}

\textsuperscript{36} See Duda, Hart and Stork, \textit{loc. cit.}
Fig. 11.10 Two alternative tree segmentations of the training data in Fig. 11.9

Fig. 11.11 Simpler decision tree generation after principal components transformation of the data
generalises well. Several strategies for pruning exist,\(^{37}\) including just working upwards through the tree by layers and noting the drop in generalisation accuracy.

### 11.8.2 Random Forests

In Sect. 8.18 we considered committee classifiers as typical of an ensemble approach that led to strong classification decisions using component classifiers that in themselves may not perform well. We can also form ensembles of decision trees with the same goal in mind. One particularly successful decision tree committee is the Random Forest.\(^{38}\) As its name implies it is a collection of trees (a “forest”) that are somehow random in their construction.

In common with other supervised classifiers we assume we have available a labelled set of training pixels. Those pixels are not used as a complete set as would be the case with single stage supervised classifiers such as the maximum likelihood decision rule. Instead, bootstrapped samples are used in the following manner.\(^{39}\) If there are \(K\) pixels in the training set we randomly select \(K\) with replacement. In other words the first training pixel is selected and then returned to the set; then the second a pixel is selected, and so on. In this manner it is possible for one or more training pixels to be replicated in the bootstrapped sample chosen to train the first tree in the random forest. Using this training set a CART style decision tree is developed. Before that can happen though a decision has to be made as to the feature set that will be used to grow the tree. If there are \(N\) features in the spectral domain a small number\(^{40}\) \(n \ll N\) is selected randomly for the first tree. That small set is used at each node in the CART process. Typically, the Gini impurity is employed to find the best split and the best one of the \(n\) features to use in that split.

We then need to assess how well the (first) tree generalises. The classic approach is to have a separate testing set of pixels which would be run through the tree to see what errors are made. In the case of a random forest, however, those pixels in the original training set that are not picked up in the bootstrapped sample used to develop a particular tree can be used as testing pixels. It turns out that taking a sample \(K\) with replacement from the available pixels, leaves out about one third of the original set. They are the pixels that are used to check tree performance.

Clearly, the first tree, trained in this manner, would not be expected to perform well. As a consequence, a second tree is grown using another bootstrapped sample

---

\(^{37}\) See Breiman et al., *loc. cit*  
\(^{39}\) See also Sect. 8.18.1  
\(^{40}\) A suggested value for \(n\) is \(\sqrt{N}\). For hyperspectral data sets this will most likely be too big.
from the available training pixels along with a second random selection of features, but with the same dimension $n$ as used in the growth of the first tree. A third tree is developed in the same manner. The newly grown trees are tested using the pixels from the original training set left over after the bootstrapped samples were chosen for training. We now have three trees capable of performing a classification on the same dataset. In order to combine their results we use a majority vote logic, in that the actual label allocated to an unknown pixel is given by the most favoured label among the trees. Sometimes this is called modal logic.

The process just described is continued through the addition of as many randomly generated trees as necessary in order to reduce classification error and thus produce results with the desired accuracy. It would not be uncommon for several hundreds to thousands of trees to be generated randomly in pursuit of such a goal.

There are two requirements for the random forest methodology to work well. First, the trees generated randomly have to be uncorrelated; the choice of the bootstrapped training sets (with replacement) provides that. Secondly, the individual trees should be strong classifiers. Generally, classifier strength will increase with the number of features used at each decision node. However, that increases substantially the complexity of tree growth so that the number of features is nevertheless kept small and weak classifiers are generally used.

### 11.8.3 Progressive Two-Class Decision Classifier

Another tree classifier is shown in Fig. 11.12. It makes sequential binary decisions, operating similar to the one-against-one multiclass strategy of Sect. 8.17. As indicated in the figure, at the first decision node a separation is made between classes 1 and 2; pixels from all other classes will not be separated and may appear in both subsets. At the left hand node in the second layer, class 2 pixels are not considered any further since they were split off in the first decision; instead a split is now made between class 1 and another class, in this case class 3. In the right hand node in the second layer class 3 is also split off. Ultimately, all pixels will be split into the constituent set of classes as a result of the progressive set of decisions, each based on two of the training classes. In Fig. 11.12 the overbars indicate which classes are not considered any further at any decision node, having been handled earlier.

---

Since pairs of classes are considered at each node, and not class subsets, the algorithm to be used and the set of features for separating that pair can be chosen optimally for those classes and could thus be different at each decision node.

The method has been tested on four difficult-to-separate classes, leading to an overall accuracy of 72.5% when using the maximum likelihood classifier as the decision rule, compared with a performance of 64.4% when the four classes are treated in a single step.42

11.9 Image Interpretation through Spectroscopy and Spectral Library Searching

One of the great benefits of recording many, finely spaced spectral samples for a pixel by an imaging spectrometer is that a scientific approach to interpretation can be carried out, as against a machine learning route to image understanding. Effectively, the latter looks for separable patterns in the data in spectral or feature space, whereas a scientific analysis seeks to associate observed spectral features with known chemical characteristics. That is the basis of spectroscopy used in many fields and is why hyperspectral sensors are known as imaging spectrometers.

Absorption features in recorded spectra, seen as localised dips, usually provide the information needed for identification, and are referred to as **diagnostically significant features**. Characterisation and recognition of those features is of paramount importance when taking an expert spectroscopic approach to the analysis of hyperspectral imagery. They are described by their locations along the spectrum, and their relative depths and widths. Feature identification by the analysis of

---

spectral properties has been used in many remote sensing applications, particularly for soils and rocks.\textsuperscript{43}

The absorption features result from photon interaction with the atomic structure of the chemicals that make up the material being imaged. To be able to quantify them it is necessary, first, to separate them from the background continuum of the spectrum that derives from light transmission and scattering.

The importance of continuum removal is illustrated in Fig. 11.13. Often the continuum will not be horizontal, which makes the measurement of the properties of an absorption feature, especially its depth, difficult. If the continuum in the vicinity of the feature is defined by a line of best fit between those parts of the spectrum either side of the feature then a reasonably consistent measure of band depth results.

In such an approach to interpretation the spectrum is generally divided into several spectral regions, usually under the guidance of a domain expert. Absorption features are then detected in each of those regions. An unknown pixel is labelled into a class for which its diagnostically significant features match those of prototype spectra from that same class stored in a spectral feature library.

The process can be complicated when pure substances are embedded in mixtures. Also some materials have very similar absorption features. Those complexities make the approach less than straightforward\textsuperscript{44} but quite sophisticated techniques have been devised to make it practical.

\textsuperscript{43} See Imaging Spectroscopy Special Issue, Remote Sensing of Environment, vol. 113, 2009 for several examples which demonstrate the value of the spectroscopic approach

Not all spectra have identifiable absorption characteristics so that the continuum removal method outlined in Fig. 11.13 may lead to identification failures for some cover types. Recent library searching methods have been based on features from both continuum included and continuum removed spectra. Thematic mapping is carried out by similarity matching recorded data, often an a field rather than pixel basis, with library prototypes.45

11.10 End Members and Unmixing

A challenge that has faced image analysts throughout history of remote sensing has been need to handle mixed pixels.46 They represent a mixture of cover types or information classes; they occur whenever there are indistinct boundaries between cover types and whenever the classes of interest to the user exist implicitly in mixtures, such as in the analysis of geological regions.

Early on several efforts were directed to resolving the proportions of pure cover types within mixed pixels by assuming that the measured radiance is a linear combination of the radiances of the “pure” constituents in each of the imaging wavebands used. With low spectral resolution data that approach generally did not meet with a great deal of success because most cover types are not well differentiated in the small number of wavebands of the instruments available in the 1980s and 1990s. However, with hyperspectral data, the prospect for uniquely characterising a vast number of cover types, and thus differentiating them from each other spectroscopically, suggests that the mixing approach be revisited as a means for establishing mixture proportions of pure cover types. This has particular relevance in mineral studies where abundance maps for the minerals of interest could be produced, based on the proportions determined for all the pixels in an image.

If we make the assumption that the total radiance recorded for a given pixel is a linear combination of the radiances of its constituents, the process can be developed mathematically in the following manner. This assumption is supportable if we accept that the incident energy is scattered only once to the sensor from the landscape and does not undergo multiple scatterings from among, for example, foliage components. Clearly that assumption will be violated in some situations; then the following simple approach may not work well.

Assume there are $M$ pure cover types in the image of interest. In the nomenclature of mixing models these are referred to as end members. We assume they are known to us, perhaps because of regions of pure cover in the image.


46 They were even referred to as “mixels” in some image processing systems.
Let the proportions of the various end members in a pixel be represented by
$f_m, m = 1 \ldots M$. They are the unknowns in the process which we need to find, based
on observing the hyperspectral reflectance of the pixel.

Let $R_n, n = 1 \ldots N$, be the observed reflectance of the pixel in the $n$th band
recorded by the sensor, and $a_{mn}$ be the spectral reflectance in the $n$th band for the
$m$th end member. Then, as above, we assume

$$R_n = \sum_{m=1}^{M} f_m a_{mn} + \xi_n$$

in which $\xi_n$ accounts for any errors in band $n$. In words, this equation says that the
observed reflectance in each hyperspectral band is the sum of the reflectances of the end members in that band. The extent to which it does not work exactly is provided for in the error term. In matrix form the mixing equation can be
expressed

$$\mathbf{R} = \mathbf{A}\mathbf{f} + \mathbf{\xi}$$

in which $\mathbf{f}$ is a column vector of mixing proportions, of size $M. \mathbf{R}$ is the spectral
reflectance vector and $\mathbf{\xi}$ is the error vector; they are column vectors of size $N. \mathbf{A}$ is
an $N \times M$ matrix of end member spectral responses by column.

Spectral unmixing, as the process is called, involves finding a set of end member proportions that minimise the error vector $\mathbf{\xi}$. On the assumption that the
correct set of end members has been chosen, the problem is then one of solving the
error free equation

$$\mathbf{R} = \mathbf{A}\mathbf{f}$$

Normally there are more equations than unknowns so that simple inversion to
find the vector of mixing proportions is not possible. Instead, a least squares
solution is found by using the Moore–Penrose pseudo inverse

$$\mathbf{f} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{R}$$

There are two constraints that the mixing proportions must satisfy, and that
need to be taken into account during the inversion process. The first is that the
proportions are all positive and less than one, and the second is that they must sum
to unity:

$$0 \leq f_m \leq 1 \text{ and } \sum_{m=1}^{M} f_m = 1$$

In practice, these constraints are sometimes found to be violated, particularly if the
end members are derived from average cover type spectra or the end member
selection is poor.\(^{47}\)

\(^{47}\) See H.N. Gross and J.R. Schott, Application of spectral mixture analysis and image fusion
In this development we have assumed that the end member matrix $A$ is known. Sometimes that will not be the case and the unmixing process needs not only to find the mixing proportions but an acceptable set of end members as well. There are several methods by which that can be approached, including the adoption of independent component analysis or independent factor analysis \[48\] and use of the Gaussian mixture model approach of Sect. 8.4.

11.11 Is There a Best Classifier?

This question has been in the minds of image analysts ever since alternative classifiers became available. It also drives research in classifier design, in which comparative results are often quoted to demonstrate that new algorithms perform better than their predecessors. Is it conceivable that better performing algorithms and methodologies will always be found? \[49\] In this section we examine that question and draw the conclusion that, provided a particular algorithm does not suffer any theoretical deficiencies, if properly applied it should deliver results as good as any other classifier. Outcomes are driven more by how the algorithm is used rather than by any inherent superiority of the algorithm itself.

As noted in Sect. 3.3 classification is a mapping from measurement space to a set of labels in the form of a thematic map. Most algorithms perform this mapping using the measurement attributes of the pixels, although some processes exploit context information as well, as we saw in Sect. 8.20. Here we will not add that complication and focus just on the pixel-specific classification task.

How large is the measurement space? For a sensor with $C$ channels and a radiometric resolution of $b$ bits per measurement there are $2^{bC}$ cells or individual sites in the discrete measurement space. This can be an enormous number, but for the moment suppose we had an instrument with just two bands, and one bit of radiometric resolution. The measurement space then consists of just four cells. Conceivably, based on training data or some other form of reference information, we could attach a label to each of those four cells so that any subsequent measurement could be classified by reference into which cell the measurement fell. That is the basis of the table look up classifier of Sect. 8.9.

It could be argued that the ideal classifier is one that is able to attach a label uniquely to each of the cells in the measurement space. Clearly, the huge size of that space for most modern instruments, along with the fact that many cells are


unoccupied, means that labelling at the individual cell level is totally impracticable. As a consequence, we use algorithms that segment the measurement space rather than label each of its cells. The suitability of a classifier is then very much related to how well the measurement space is segmented, and that in turn depends on how well the analyst understands the measurement space.

In the very early days of pattern recognition the linear discriminant function was a common algorithm. The available training data was used to locate a hyperplane between the training classes which thus segmented the measurement space into regions that were associated with the training classes. Whether that classifier then generalised well, by performing accurately on testing data, depended on how good the location of the hyperplane was. By comparison to labelling each of the individual cells in the measurement space, we can see that there is a trade-off between ease of training (finding a hyperplane rather than trying to label every cell) and performance. More generally, we can describe the trade-off as between how easily a classifier can be trained and used, and the performance it delivers. Such trade-offs are important in operational thematic mapping.

Clearly, the simple hyperplane of the previous paragraph would not be used now, except in the simplest of circumstances. Modern classifiers segment the data space into much more flexible regions so that good generalisation is possible. Nevertheless, among all the approaches available, the one essential requirement that the user should look for is how well an algorithm accommodates the actual distribution of pixels in the measurement space; the representativeness of the training data gets expressed ultimately in the accuracy of the final classification. That emphasises the importance of data representation when using any of the techniques. How the training data, as a window into the distribution of pixels in the measurement space, is handled by each approach will set the ultimate performance of the classifier. As a simple illustration, if multimodality in training data is not resolved when applying the Gaussian maximum likelihood classification, good results cannot be expected.

It is, of course, difficult to envisage the data space explicitly, except in simple cases where we might create a scatter plot from two of the available channels. Instead, we often have to infer the properties of the space from other observations. We do that by clustering in the hybrid classification methodology in Sect. 11.4. More generally, we have to ensure that any algorithm matches the data structure during training. For the three common supervised approaches this suggests that spectral classes must be identified when using the maximum likelihood rule, the kernel function and the regularisation parameter used with the support vector machine need to be chosen carefully and the selection of the hidden layer for a neural network approach can be critical.

We know that the Gaussian maximum likelihood approach works well when feature reduction has been properly applied and multimodality is resolved. The support vector machine using polynomial or radial basis function kernels has been seen to work well, presumably because the higher order space, although not observed directly creates, at minimum, high order polynomial and quadratic boundaries that we know work well because of our experience with the maximum
likelihood rule. The neural network situation is more difficult to understand because of the connection complexity; nevertheless, the example in Sect. 8.19.4 suggests it is capable of implementing a powerful piecewise linear decision surface. It is the flexibility of such a surface that renders the neural network an effective approach, even though beforehand we cannot induce performance by prior assessment of how well its properties are matched to the structure of the measurement space. In many ways, the neural network is an overkill brought about by over-specifying the size and complexity of the hidden layer. Some analysts even use more than the single hidden layer to achieve a given outcome.

The outputs from the support vector and neural network approaches do not interface well with those other procedures, nor do they naturally provide relative measures of likelihood. Although coding methods can be devised in each case partly to overcome those limitations, they are by no means standardised, nor readily comparable to the posteriors from maximum likelihood classification.

From this performance perspective we could infer that each of these well-recognised approaches, when used properly, performs acceptably and that, on the basis of generalisation accuracy, one would probably not be preferred over any other. However there are other comparators that we should take into account, particularly from an operational perspective. To consider those we will continue the comparison of the maximum likelihood rule, the support vector machine and the neural network, examining the methodologies that would be employed in practice with each algorithm. Table 11.8 presents such a comparison.

Against this outline, Table 11.9 summarises the practical considerations in each case. As is to be expected there are always trade-offs. Where one approach excels, the others may have limitations. Nonetheless, it is important in comparative studies that the methodologies summarised Table 11.8 are used for each procedure. Too often, one sees comparisons that conclude that certain approaches are superior to others and yet care has not been taken to ensure that the algorithmic methodologies have been properly applied. The very reason for placing an algorithm within its operational methodology is to ensure that the measurement space is segmented in such a manner as to ensure that the algorithm performs as well as possible. Even moderately primitive algorithms such as the parallelepiped classifier can be made to perform well provided the data space is appropriately segmented.

There is some very sobering guidance provided by the No Free Lunch Theorem from the field of machine learning, which we state as

If the goal is to obtain good generalisation performance, there are no context-independent or usage-independent reasons to favour one learning

---

50 See J.A. Richards, Is there a best classifier? SPIE Conference on Remote Sensing, Bruges, Belgium 20-22 Sep 2005, for a set of comparative results, some using the same data set, in which the variability in reported accuracy is small, and one approach seems not to rise above the others

51 From Richards, ibid.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian maximum likelihood</td>
<td>1. Obtain labelled training data, noting concerns about poor generalisation if the Hughes phenomenon is encountered.</td>
</tr>
<tr>
<td></td>
<td>2. Apply feature reduction. With hyperspectral data that can be a non-trivial task. Standard divergence-based techniques cannot be applied because the necessary class statistics cannot be estimated reliably. Nonparametric weighted feature extraction is a viable candidate, but it is computationally demanding.</td>
</tr>
<tr>
<td></td>
<td>3. Resolve multimodality, presumably by clustering. Information classes have now been resolved into (spectral) sub-classes.</td>
</tr>
<tr>
<td></td>
<td>4. Compute the statistics (mean vector and covariance matrix) for each spectral class.</td>
</tr>
<tr>
<td></td>
<td>5. Apply the discriminant function of equation (8.7) to perform multi-class labelling into the spectral classes.</td>
</tr>
<tr>
<td></td>
<td>6. Generate the thematic map, and map information classes to spectral classes.</td>
</tr>
<tr>
<td>Support vector machine</td>
<td>1. Obtain labelled training data.</td>
</tr>
<tr>
<td></td>
<td>2. Determine the multiclass strategy to be used.</td>
</tr>
<tr>
<td></td>
<td>3. Determine the kernel function to be used.</td>
</tr>
<tr>
<td></td>
<td>5. Train the machine through optimisation while also optimising kernel parameters and the regularisation parameter. May require several trials to find the best parameters. This step can be time consuming.</td>
</tr>
<tr>
<td></td>
<td>6. Generate the thematic map, via a tree of binary decisions.</td>
</tr>
<tr>
<td>Neural network</td>
<td>1. Obtain labelled training data.</td>
</tr>
<tr>
<td></td>
<td>2. Choose the input and output layer codings; that is generally straightforward.</td>
</tr>
<tr>
<td></td>
<td>3. Choose the number of hidden layers (one is usually sufficient in remote sensing) and the number of processing elements in the hidden layer. That is a difficult task in principle; it is generally handled by over-specifying the layer in case poor results are generated by not building enough flexibility into that centrally important element.</td>
</tr>
<tr>
<td></td>
<td>4. Train the network iteratively.</td>
</tr>
<tr>
<td></td>
<td>5. Refine the network through pruning unwanted hidden layer nodes.</td>
</tr>
<tr>
<td></td>
<td>6. Generate the (multiclass) thematic map.</td>
</tr>
</tbody>
</table>

or classification method over another. If one classification algorithm seems to outperform another in a particular situation, it is a consequence of its fit to the particular pattern recognition problem, not the general superiority of the algorithm.

This does not imply that there are no poor classifiers, but it reminds us that the classifiers we have become to consider as our benchmarks in remote sensing do not inherently perform well—they perform well when the analyst uses them properly in a given task, which is the central message of this section. Overall, there is probably no best classifier.
Table 11.9 Qualitative comparison of the three common classifier types

<table>
<thead>
<tr>
<th>Comparator</th>
<th>Gaussian maximum likelihood</th>
<th>Neural network</th>
<th>Support vector machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of training samples</td>
<td>Does not need to be large, but Hughes phenomenon is a consideration.</td>
<td>Often a large number is used given the number of unknown weights to be found.</td>
<td>Usually a large number is used, from which the support vectors are selected.</td>
</tr>
<tr>
<td>Training time</td>
<td>Small, since it only depends on estimating sample statistics. Clustering required to resolve multimodality will add to the time.</td>
<td>Can be very large given the number of epochs required for convergence.</td>
<td>Usually large and quadratically dependent on the number of training samples. Optimal kernel and regularisation parameters need to be found for each constituent classifier through grid searching.</td>
</tr>
<tr>
<td>Resolving multimodality</td>
<td>Required, otherwise poor class modelling will result. Number of modes can be over-specified if necessary.</td>
<td>Not a consideration since the algorithm is already able to handle the primary data characteristics.</td>
<td>Not a consideration since the algorithm is already able to handle the primary data characteristics.</td>
</tr>
<tr>
<td>Multiclass capability</td>
<td>Inherent in the algorithm.</td>
<td>Inherent in the algorithm.</td>
<td>Requires the binary decisions to be incorporated into a tree structure. This adds to training complexity.</td>
</tr>
<tr>
<td>Feature selection</td>
<td>Essential for high dimensional data sets, otherwise poor generalisation results. This can be a time-consuming.</td>
<td>Not essential but will expedite training.</td>
<td>Not essential but will expedite training.</td>
</tr>
<tr>
<td>Classification time</td>
<td>Generally not high, but depends quadratically on dimensionality.</td>
<td>Generally not high.</td>
<td>Can be large because of the tree decisions required. However it is only linearly dependent on dimensionality.</td>
</tr>
</tbody>
</table>

54 Recent research has shown that the maximum likelihood rule, among all common classifiers, is more robust to unbalanced training data sets, in which the different classes are represented by widely different numbers of pixels. This is consistent with the observation that maximum likelihood is robust to errors in the assumption of normality. See K. Song, Tackling Uncertainties and Errors in the Satellite Monitoring of Forest Cover Change, Ph.D. Dissertation, The University of Maryland, 2010. http://drum.lib.umd.edu/handle/1903/10523.
We repeat here a point that is material in some circumstances. If classifier results are to be incorporated into some other fundamentally-statistical process, such as a multisource statistical analysis procedure, a Markov random field model or post classification processing like relaxation labelling, then the posterior probabilities generated from the maximum likelihood rule are already in the form required for use in those other techniques. Furthermore, the set of posterior probabilities can be used before the maximum selection step in the maximum likelihood rule to give measures of relative likelihoods of the less favoured classes.

11.12 Bibliography on Image Classification in Practice

An important early paper that seemed to be the first to address a real image classification methodology, and which is now available in readily accessible form, is

M.D. Fleming, J.S. Berkebile and R.M. Hofer, Computer aided analysis of Landsat 1 MSS data: a comparison of three approaches including a modified clustering approach, Information Note 072475, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana, 1975.

http://www.lars.purdue.edu/home/references/LTR_072475.pdf

An easily read account of the sampling and quantitative procedures used to assess the performance of a classifier and the accuracy of a thematic map is given in


while the following supplements that treatment with the necessary statistical material


Other recognised historical accounts on assessing accuracy are


Although the kappa coefficient has been a popular measure of map accuracy, its efficacy is now seriously questioned, and alternative metrics are proposed, for which see

As well as in Congalton and Green above, a significant discussion of the kappa coefficient, including its variants, will be found in


Decision tree material will be found in many of the standard image processing textbooks. A specialized treatment is


Some significant research results of historical importance are


The interesting field of random forest classifiers is covered in


while their application to remote sensing problems is demonstrated in


Spectroscopic and library matching techniques for the analysis of hyperspectral data is covered in

An excellent review spectral unmixing procedures, with references to many of the important works in this area, and the need for care in selecting end members, will be found in


A good range of unmixing applications is covered in the


11.13 Problems

11.1. What is the difference between an information class and a spectral class? The notion of spectral class has more relevance to optical and thermal remote sensing imagery than to other data types such as radar. When analysing co-registered data sets it is often necessary to determine the class structures for each data set independently, before they are combined into the information classes of interest to the user. In such cases the term data class may be more relevant than spectral class; see Sect. 12.3.1.

11.2. Four analysts use different qualitative methodologies for interpreting spectral imagery. They are summarised below. Comment on the merits and shortcomings of each approach and indicate which one do you think is most effective.

Analyst 1

1. Chooses training data from homogeneous regions for each cover type.
2. Develops statistics for a maximum likelihood classifier.
3. Classifies the image.

Analyst 2

1. Performs a clustering of the whole image and attaches labels to each cluster type afterwards

Analyst 3

1. Chooses several regions within the image, each of which includes more than one cover type.
2. Clusters each region and identifies the cluster types.
3. Uses statistics from the clustering step to perform a maximum likelihood classification of the whole image.

Analyst 4

1. Chooses training fields within apparently homogeneous regions for each cover type.
2. Clusters those regions to identify spectral classes.
3. Uses statistics from the clustering step to perform a maximum likelihood classification of the whole image.

11.3. For the method you have identified as preferable in Prob. 11.2 comment on how separability measures would be used to advantage. What if the data were of hyperspectral dimension?

11.4. The spectral classes used with the maximum likelihood decision rule in supervised classification are assumed to be representable by single multivariate normal probability distributions. Geometrically, this implies that they will have hyperellipsoidal distributions in the spectral domain. Do you think that clustering by the iterative moving means algorithm will generate spectral classes of that nature? Given this observation, how might you best generate spectral classes for maximum likelihood classification, the minimum distance classifier and parallelepiped classification?

11.5. How can thresholds be used to help in the identification of spectral classes when using the maximum likelihood decision rule?

11.6. Is there a preferred kernel for use with the support vector machine when classifying high dimensional remote sensing image data?

11.7. Just before the labelling step when generating a thematic map using a classifier, each pixel has associated with it a set of measures that guide the allocation of the preferred label. For example, in the case of maximum likelihood classification it is the posterior probabilities. What measures of typicality would apply in the case of the support vector machine and the neural network?

11.8. Suppose a particular image contains just two cover types—vegetation and soil. A pixel identification exercise is carried out to label each pixel as either soil or vegetation and thus generate an estimate of the proportion of vegetation in the region being imaged. For homogeneous regions the labelling exercise is straightforward. However, the image will also contain a number of mixed pixels so that end member analysis would be considered as a means for resolving their soil/vegetation proportions. Is the additional work warranted if the approximate proportion of vegetation to soil is 1:100, 50:50, or 100:1?

11.9. This question relates to the effect on classification accuracy of resampling that might be used to correct imagery before analysis. For simplicity, consider a single line of infrared image data over a region that is vegetated to the left and water to the right. Imagine the vegetation/water boundary is sharp. Resample your line of data onto a grid with the same centres as the original, using both nearest neighbour and cubic convolution interpolation.
Clearly, in the latter case the interpolation is just along the line, rather than over a neighbourhood of 16 surrounding pixels. Comment on the results of classifying each of the resampled pixels given that a classifier would have been trained on classes that include those with responses between vegetation and water.

11.10. Sometimes the spectral domain for a particular sensor and scene consists of a set of distinct clusters. An example would be a near infrared versus red spectral domain of an image in which there are regions of just water, sand and vegetation. The spectral domain would then have three distinct groups of pixels. More often than not, particularly for images of natural vegetation and soil, the spectral domain will be a continuum because of the differing degrees of mixing of the various cover types that occur in nature. One is then led to question the distinctness and uniqueness, not only of spectral classes, but information classes as well. Comment on the issues involved in the classification of natural regions both in terms of the definition of the set of information classes and how spectral classes might be identified to assist in training.

11.11. Manually design a simple decision tree that could be used efficiently with ETM+ data for classification into deep water, shallow water, green vegetation and soil.

11.12. A particular study requires the mapping of water temperature in the effluent from an industrial complex located on the side of a river. There is no interest in the land itself. Using ETM+ data design a strategy, most likely using a decision tree that, first, separates the soil and water and then maps the temperature variation within the water.

11.13. The error matrix is a very comprehensive summary of the accuracy of the classes on a map produced from a classification exercise. Perhaps its only drawback is the number of elements it contains, particularly for a large number of classes. That is why in many cases we turn to single measures such as total classification accuracy and Kappa coefficient. Discuss the benefits and disadvantages of single metrics as against retaining the complete error matrix.

11.14. Generate a CART decision tree for the two class data shown in Fig. 10.6. Then perform a principal components transformation and repeat the exercise.