Reference

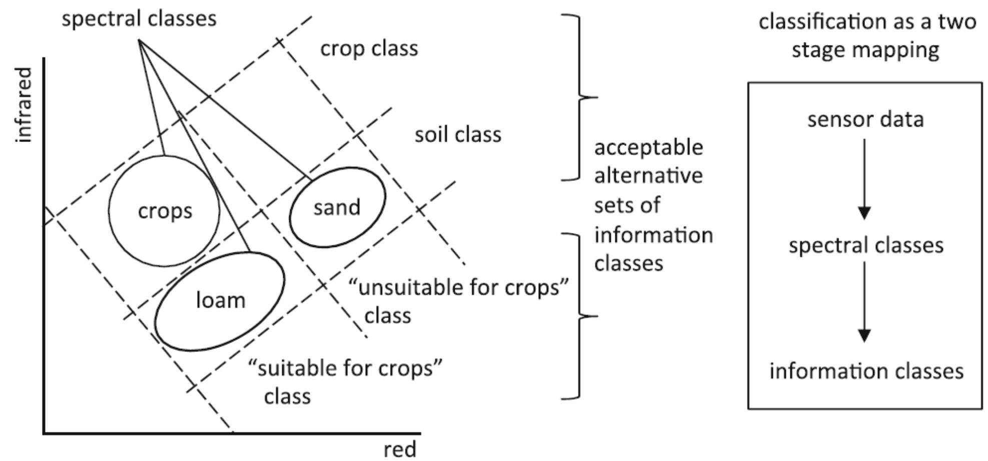
John A. Richards. 2013. *Remote Sensing Digital Image Analysis: An Introduction.* Springer-Verlag, Berlin, Heidelberg, 5th edition.

Supervised classification

- Supervised classification is the technique most often used for the quantitative analysis of remote sensing image data.
- At its core is the concept of segmenting the spectral domain into regions that can be associated with the ground cover classes of interest to a particular application.
- A variety of algorithms is available for the task.
- The different methods vary in the way they identify and describe the regions in spectral space.
  - Some seek a simple geometric segmentation.
  - Others adopt statistical models with which to associate spectral measurements and the classes of interest.
  - Some can handle user-defined classes that overlap each other spatially and are referred to as *soft classification* methods.
  - Others generate firm boundaries between classes and are called *hard classification* methods.

The Essential Steps in Supervised Classification

- Supervised classification is essentially a mapping from the measurement space of the sensor to a field of labels that represent the ground cover types of interest to the user.
- It depends on having enough pixels available, whose class labels are known, with which to train the classifier.
- *Training* refers to the estimation of the parameters that the classifier needs in order to be able to recognize and label unseen pixels.
- The labels represent the classes on the map that the user requires.
- The map is called the thematic map, meaning a map of themes.



The Essential Steps in Supervised Classification

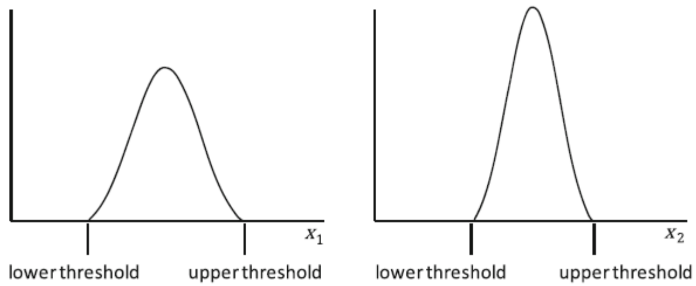
- The classes of interest to the user should occupy unique regions in spectral space.
- There is a one-to-one mapping between the measurement vectors and class labels.
- *Spectral classes*: the classes into which the data naturally groups the (in this simple example the crops, loam and sand).
- *Information classes*: those that match user requirements.

Essential Practical Steps in Applying a Classifier

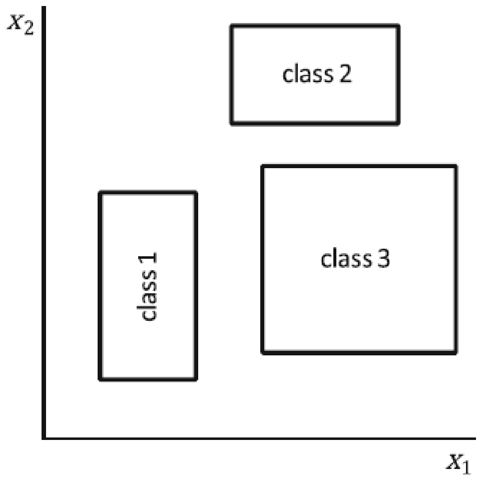
1. Deciding on the set of ground cover type classes into which to segment the image.
2. *Training data*: Choosing known, representative pixels for each of the classes.
3. Training sets for each class can be established using site visits, maps, air photographs or even photointerpretation of image products formed from the data.
   - Using the training data to estimate the parameters of the particular classifier algorithm to be employed.
4. Using the trained classifier to label every pixel in the image (i.e. whole image) as belonging to one of the classes specified in step 1.
5. Producing thematic (class) maps and tables which summarize class memberships of all pixels in the image, from which the areas of the classes can be measured.
6. Assessing the accuracy of the final product using a labelled *testing data* set.

Parallelepiped Classification

- It is trained by finding the upper and lower brightness values in each spectral dimension.
- Often that is done by inspecting histograms of the individual spectral components in the available training data.
- Together the upper and lower bounds in each dimension define a multidimensional box or *parallelepiped*.
- Unknown pixels are labelled as coming from the class of the parallelepiped within which they lie.
- It is a very simple and fast classifier.

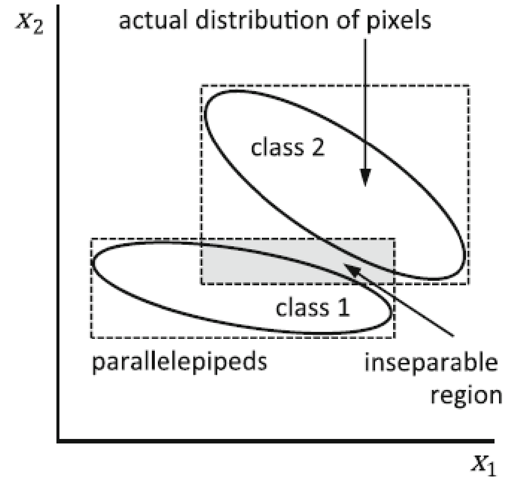


Set of 2D Parallelepipeds



- There can be considerable gaps between the parallelepipeds in spectral space; pixels in those regions cannot be classified.
- For correlated data some parallelepipeds can overlap, because their sides are always parallel to the spectral axes.
  - As a result, there are some parts of the spectral domain that can't be separated.
- There is no provision for prior probability of class membership with the parallelepiped rule.

Classification of Correlated Data Showing Regions of Inseparability



Minimum Distance Classification

- Minimum distance to class means classifier.
- Training data is used only to determine class means.
- Classification is then performed by placing a pixel in the class of the nearest mean.

Discriminant Function

Suppose $\mathbf{m}_i; i = 1 \dots M$ are the means of the M classes determined from training data.
$\mathbf{x}$ is the position of the pixel in spectral space to be classified.
Compute the set of squared Euclidean distances of the unknown pixel to each of the class means:

$$d(\mathbf{x}, \mathbf{m}_i)^2 = (\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}}(\mathbf{x} - \mathbf{m}_i) = (\mathbf{x} - \mathbf{m}_i) \cdot (\mathbf{x} - \mathbf{m}_i) \; i = 1 \dots M$$

Expanding the dot product form gives

$$d(\mathbf{x}, \mathbf{m}_i)^2 = \mathbf{x} \cdot \mathbf{x} - 2\mathbf{m}_i \cdot \mathbf{x} + \mathbf{m}_i \cdot \mathbf{m}_i$$

Classification is performed using the decision rule

$$\mathbf{x} \in \omega_i \text{ if } d(\mathbf{x}, \mathbf{m}_i)^2 < d(\mathbf{x}, \mathbf{m}_j)^2 \text{ for all } j \neq i$$

Thresholds

- Thresholds can be applied to minimum distance classification by ensuring not only that a pixel is closest to a candidate class but also that it is within a prescribed distance of that class in spectral space.
- The distance threshold is usually specified in terms of number of standard deviations from the class mean.

kNN (Nearest Neighbour) Classification

- It assumes that pixels close to each other in spectral space are likely to belong to the same class.
- An unknown pixel is labelled by examining the available training pixels in the spectral domain.
- Choosing the class most represented among a pre-specified number of nearest neighbours.
- The comparison essentially requires the distances from the unknown pixel to all training pixels to be computed.

Suppose there are k_i neighbours labelled as class ω_i among the k nearest neighbours of a pixel vector $\mathbf{x}$; noting that $\sum_{i=1}^{M} k_i = k$ where M is the total number of classes. Discriminant function for the i^{th} class is define as:

$$g_i(\mathbf{x}) = k_i$$

and the decision rule is define as

$$\mathbf{x} \in \omega_i \text{ if } g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j \neq i$$

Limitations

- In the kNN algorithm as many spectral distances as there are training pixels must be evaluated for each unknown pixel to be labelled.
- That requires an impractically high computational load, particularly when the number of spectral bands and/or the number of training samples is large.
- The method is not well-suited for hyperspectral datasets.

Unsupervised Classification

- Supervised classification require the availability of labelled training data with which the parameters of the respective class models are estimated.
- In a sense, the analyst supervises an algorithm's learning about those parameters.
- Sometimes labelled training data is not available and yet it would still be of interest to convert remote sensing image data into a thematic map of labels.
- Such an approach is called unsupervised classification since the analyst, in principle, takes no part in an algorithm's learning process.
- Several methods are available for unsupervised learning.
- Perhaps the most common in remote sensing is based on the use of clustering algorithms, which seek to identify pixels in an image that are spectrally similar.
- There are many clustering methods.

Why Unsupervised Classification [1]

- Very Expensive: Collecting and labeling a large set of sample patterns.
- Quality of training labels.
- Reverse direction: train with large amounts of (less expensive) unlabeled data, and only then use supervision to label the clusters found.
- To find features, that will then be useful for classification.
- Insight into the nature or structure of the data.
- First step for different purposes.
- Large amount of data.

[1] Duda, Hart and Stork (2000)

Clustering

Given a *representation* of n objects, find k groups based on a measure of *similarity* such that the similarities between objects in the same group are high while the similarities between objects in different groups are low.[2]

[2] Jain (2010)

Similarity Metrics and Clustering Criteria

- In clustering we try to identify groups of pixels because they are somehow similar to each other.
- The only real attributes that we can use to check similarity are the spectral measurements recorded by the sensor used to acquire the data.
- Here, therefore, clustering will imply a grouping of pixels in the spectral domain.
- Pixels belonging to a particular cluster will be spectrally similar.
- In order to quantify their spectral proximity it is necessary to devise a measure of similarity.
- Many similarity measures, or metrics, have been proposed but those used commonly in clustering procedures are usually simple distance measures in spectral space.
- The most frequently encountered are Euclidean distance L_2 and the city block or Manhattan L_1 distance.

Similarity Metrics

Euclidean distance

$$\begin{aligned} d(\mathbf{x}_1, \mathbf{x}_2) &= \| \mathbf{x}_1 - \mathbf{x}_2 \| \\ &= \{(\mathbf{x}_1 - \mathbf{x}_2) \cdot (\mathbf{x}_1 - \mathbf{x}_2)\}^{1/2} \\ &= \{(\mathbf{x}_1 - \mathbf{x}_2)^{\mathrm{T}} (\mathbf{x}_1 - \mathbf{x}_2)\}^{1/2} \\ &= \left\{ \sum_{n=1}^{N} (x_{1n} - x_{2n})^2 \right\}^{1/2} \end{aligned}$$

Manhattan distance

$$d_{L_1}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{n=1}^{N} | x_{1n} - x_{2n} |$$

Minkowski L_p distance metric

$$d_{L_p}(\mathbf{x}_1, \mathbf{x}_2) = \left\{ \sum_{n=1}^{N} | x_{1n} - x_{2n} |^p \right\}^{1/p}$$

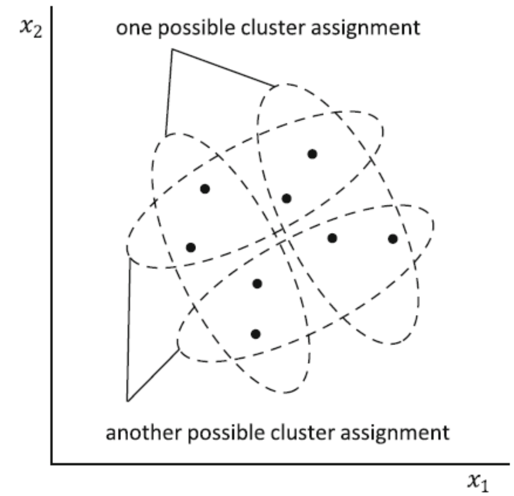
Clustering Criteria

sum of squared error (SSE) measure

$$SSE = \sum_{C_i} \sum_{x \in C_i} \| \mathbf{x} - \mathbf{m}_i \|^2 = \sum_{C_i} \sum_{x \in C_i} (\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}} (\mathbf{x} - \mathbf{m}_i)$$

- $\mathbf{m}_i$ is the mean vector of the ith cluster.
- $\mathbf{x} \in C_i$ is a pixel assigned to that cluster.
- Inner sum computes the aggregated distance squared of all the pixels in the cluster to the respective cluster mean.
- outer sum adds the results over all the clusters.
- SSE will be small for tightly grouped clusters and large otherwise, thereby allowing an assessment of the quality of clustering.

Two Acceptable Clusterings



k–Means Clustering

- Also called migrating means and iterative optimization.
- Widely used approach.
- k–Means requires an initial assignment of the available measurement vectors into a user-specified number of clusters.
- with arbitrarily specified initial cluster centres that are represented by the means of the pixel vectors assigned to them.
- This will generate a very crude set of clusters.
- The pixel vectors are then reassigned to the cluster with the closest mean, and the means are recomputed.
- The process is repeated as many times as necessary such that there is no further movement of pixels between clusters.
- In practice, with large data sets, the process is not run to completion and some other stopping rule is used.

Illustration of clustering with the k–means

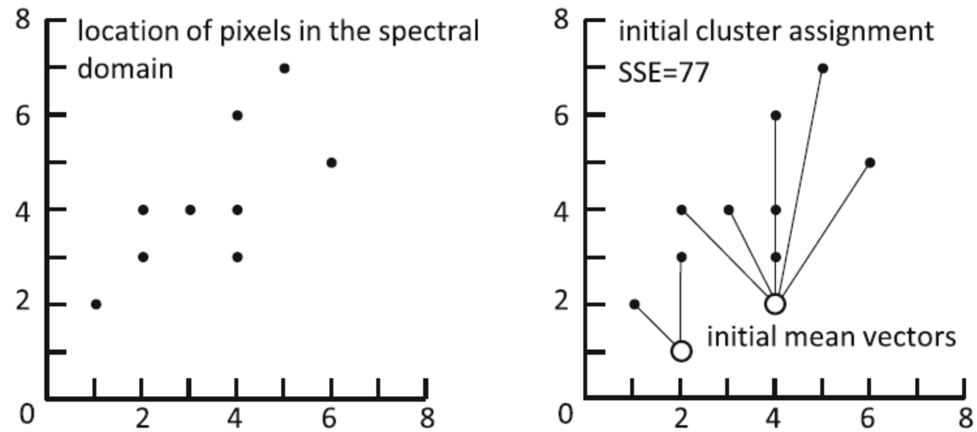


Illustration of clustering with the k–means

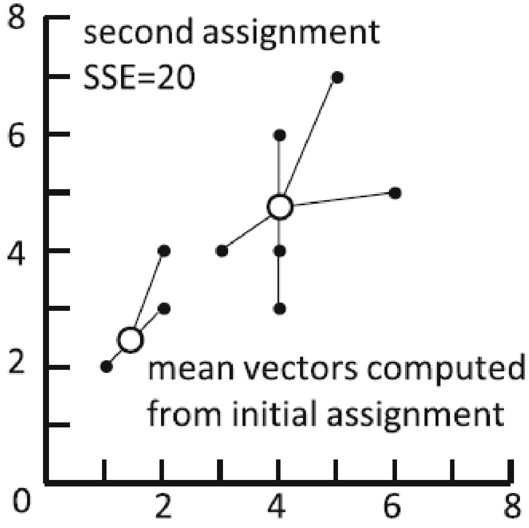


Illustration of clustering with the k–means

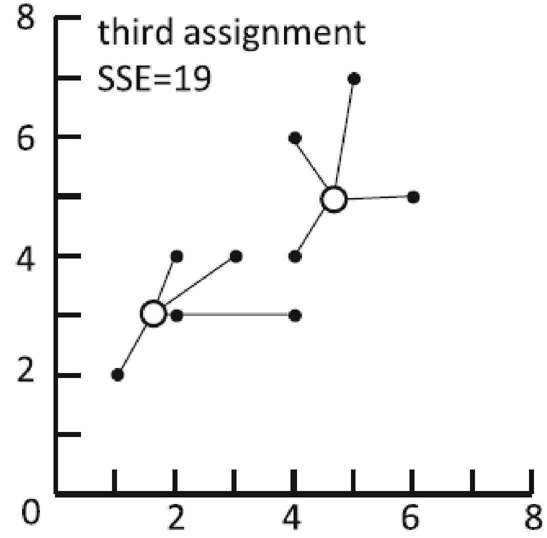


Illustration of clustering with the k–means

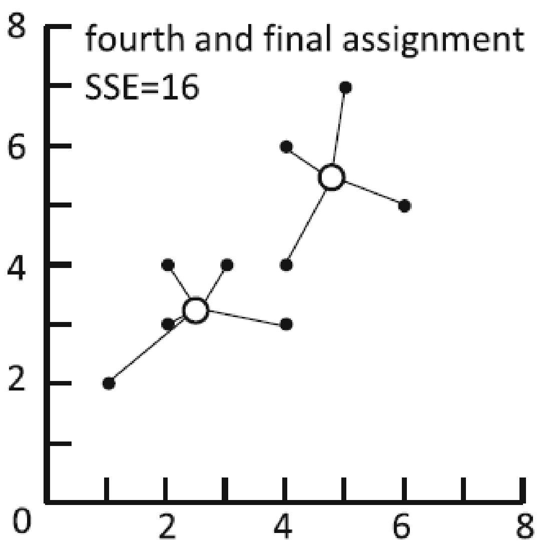
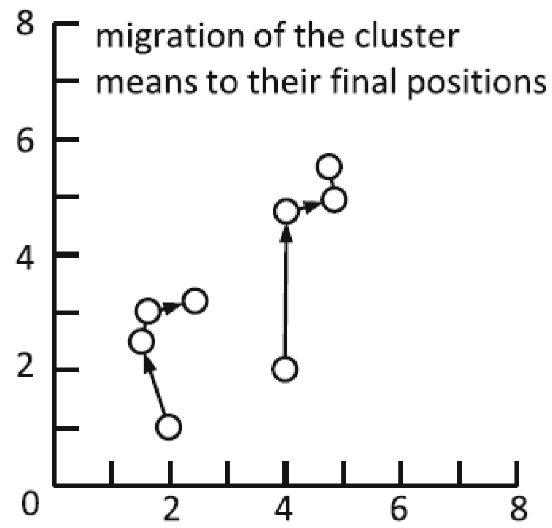


Illustration of clustering with the k–means



The k–means Algorithm

1. Select a value for C; the number of clusters into which the pixels are to be grouped.
2. Initialize cluster generation by selecting C points in spectral space to serve as candidate cluster centres.

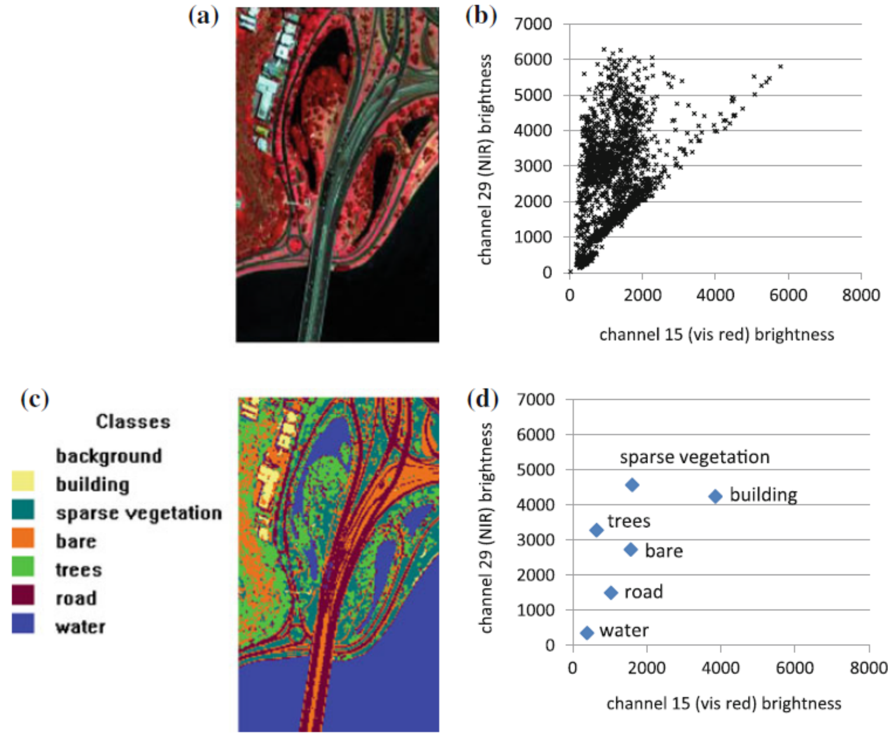
$$\hat{\mathbf{m}}_c \quad c = 1 \ldots C$$

3. Assign each pixel vector $\mathbf{x}$ to the candidate cluster of the nearest mean using an appropriate distance metric in the spectral domain between the pixel and the cluster means.
4. Compute a new set of cluster means from the groups formed in Step 3; call these

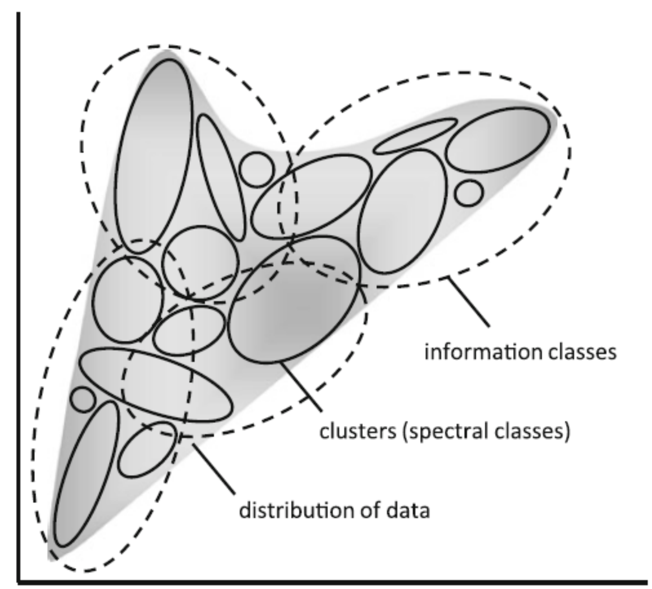
$$\mathbf{m}_c \quad c = 1 \ldots C$$

5. If $\mathbf{m}_c = \hat{\mathbf{m}}_c$ for all c then the procedure is complete. Otherwise the $\hat{\mathbf{m}}_c$ are set to the current values of $\mathbf{m}_c$ and the procedure returns to step 3.

		Cluster mean vectors (on 16 bit scale)				
Cluster	Label	Channel 7	Channel 15	Channel 29	Channel 80	Channel 108
1	Building	3511.9	3855.7	4243.7	4944.2	4931.6
2	Sparse veg	1509.6	1609.3	4579.5	3641.7	2267.0
3	Bare	1333.9	1570.7	2734.3	2715.1	2058.7
4	Trees	725.6	650.6	3282.4	1676.2	866.6
5	Road	952.3	1037.1	1503.7	1438.5	1202.3
6	Water	479.2	391.1	354.8	231.0	171.6



Relationship between Data, Clusters and Information Classes



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.
- More often the analyst has set aside labelled data to be used as a testing set after the classification has been carried out.
- 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.

The Error Matrix

- Also called a contingency matrix or a confusion matrix.
- It lists the reference data classes by column and the classes indicated on the thematic map by row.
- 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.

Error Matrix

reference data classes

thematic map classes		A	B	C	sum
	A	35	2	2	39
	B	10	37	3	50
	C	5	1	41	47
	sum	50	40	46	136

overall accuracy =(35+37+41)/136=83.1%

producer's accuracies		user's accuracies	
A	35/50=70.0%	A	35/39=89.7%
B	37/40=92.5%	B	37/50=74.0%
C	41/46=89.1%	C	41/47=87.2%

The Error Matrix

- 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.
- Errors of omission correspond to those pixels belonging to the reference class that the classifier has failed to recognize: they are therefore the off diagonal terms down the column for a particular reference class.
- 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.

- *Producer's accuracy*: is an indication of classifier performance.
- This is the probability that the classifier has labelled a pixel as class B given that the actual (reference data) class is B.
- *User's accuracy*: 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.