Environmental Remote Sensing
GEOG 2021

Lecture 4

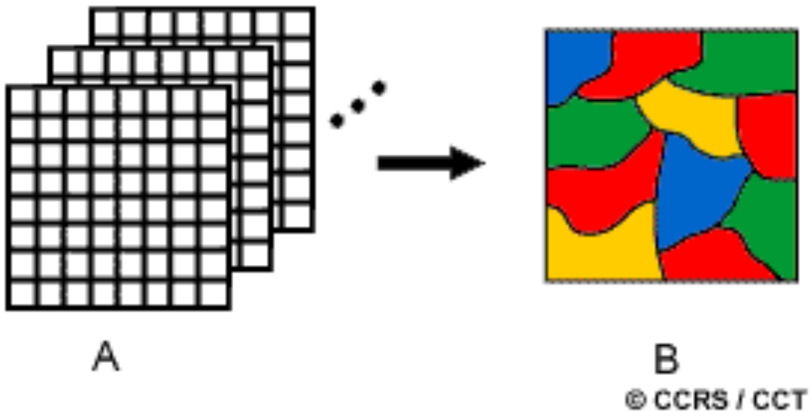
Image classification

Purpose

– categorising data
– data abstraction / simplification
– data interpretation
– mapping
  - for land cover mapping
  - use land cover class as a surrogate for other information of interest (ie assign relevant information/characteristics to a land cover class)

Multispectral image classification

- Very widely used method of extracting <u>thematic information</u>
- Use multispectral (and other) information
- Separate different land cover classes based on spectral response, texture, ….
- i.e. separability in “feature space”



Basis for 'classifying'

- method: pattern recognition
- use any/all of the following properties in an image to differentiate between land cover classes:-
  - spectral
  - spatial
  - temporal
  - directional
  - [time / distance-resolved (LIDAR)]

Spatial pattern recognition

- use spatial context to distinguish between different classes
  - e.g. measures of image texture, spatial context of 'objects ' derived from data.

Temporal pattern recognition

- the ability to distinguish based on spectral or spatial considerations may vary over the year
- use variations in image DN (or derived data) over time to distinguish between different cover types
  - e.g. variations in VI over agricultural crops

Directional pattern recognition

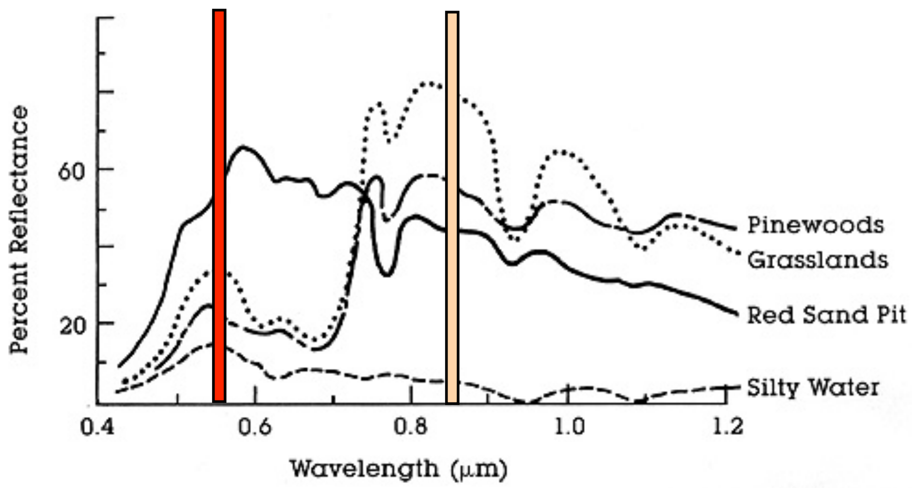
- surface with different structures will tend to give different trends in reflectance as a function of view and illumination angles

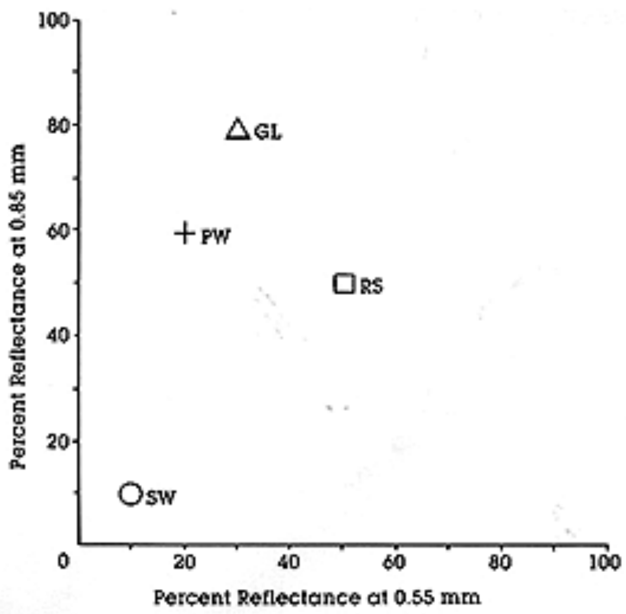
Spectral pattern recognition

- most widely used
- distinguish between different land cover classes from differences in the spectral reflectance (or more typically, image DN) in different wavebands

i.e. separate in feature space

- Use different spectral response of different materials to separate e.g. plot red v NIR DN values….





Approaches to Classification

- We need some form of automated (rule-based) classification algorithm to allow us to distinguish one surface type from another
  - Supervised Classification
  - Unsupervised Classification

Supervised classification

- training stage (significant user input/expertise)
- Identify areas of cover types of interest (map, ground survey, spectral characteristics) in bands of an image

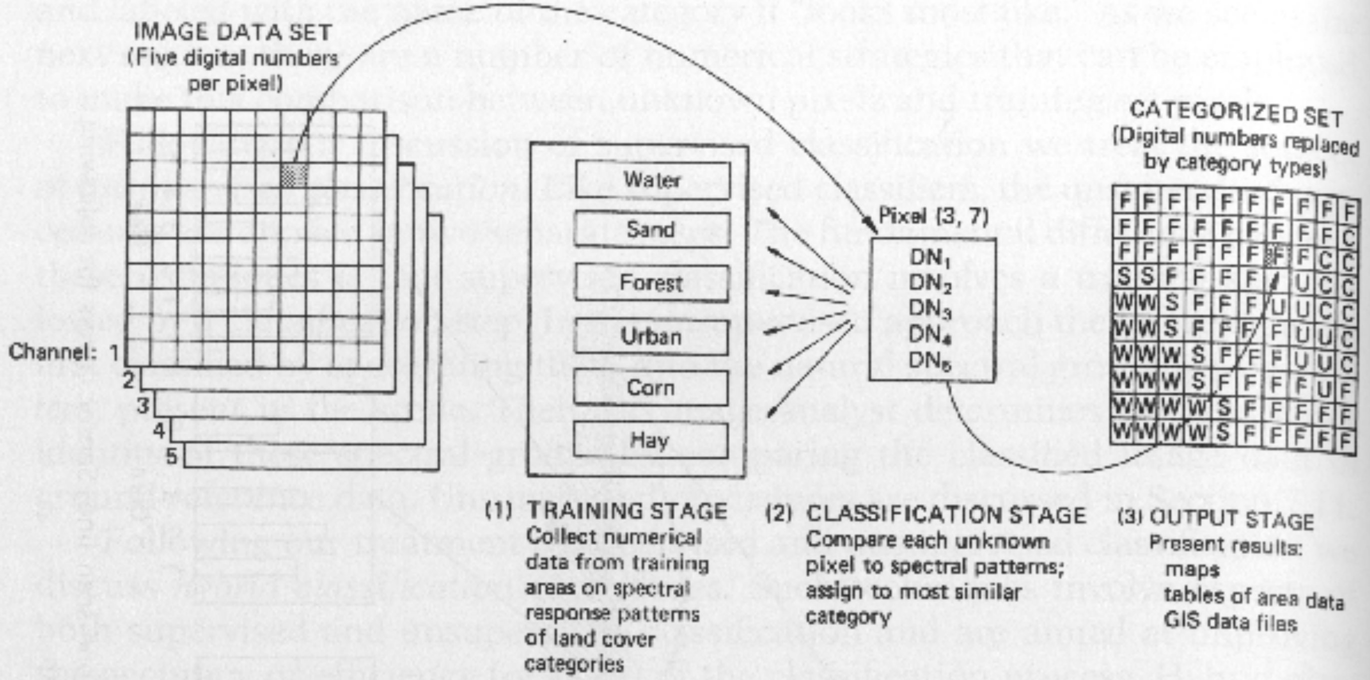


Figure 7.38 Basic steps in supervised classification.

From Lillesand, Kiefer and Chipman (2004)

Supervised classification: training stage

- areas of interest delineated by user
- spectral information on the cover types is gathered for these areas
  - Training data (subset of whole)
  - These are “classes” we will place all remaining pixels in according to their DN values
- Can plot in feature space – do we see clusters?

Supervised classification: classification stage

- Need rule(s) to decide into which class we put given pixel
- e.g. Minimum distance to means (MDM)
  - for each land cover class, calculate the mean vector in feature space (i.e. the mean value in each waveband)
  - Put every pixel into nearest class/cluster
  - define a limit beyond which a pixel remains unclassified
- a simple and fast technique but has major limitations…

Supervised classification

- Feature space clusters
- E.g. 2 channels of information
- Are all clusters separate?

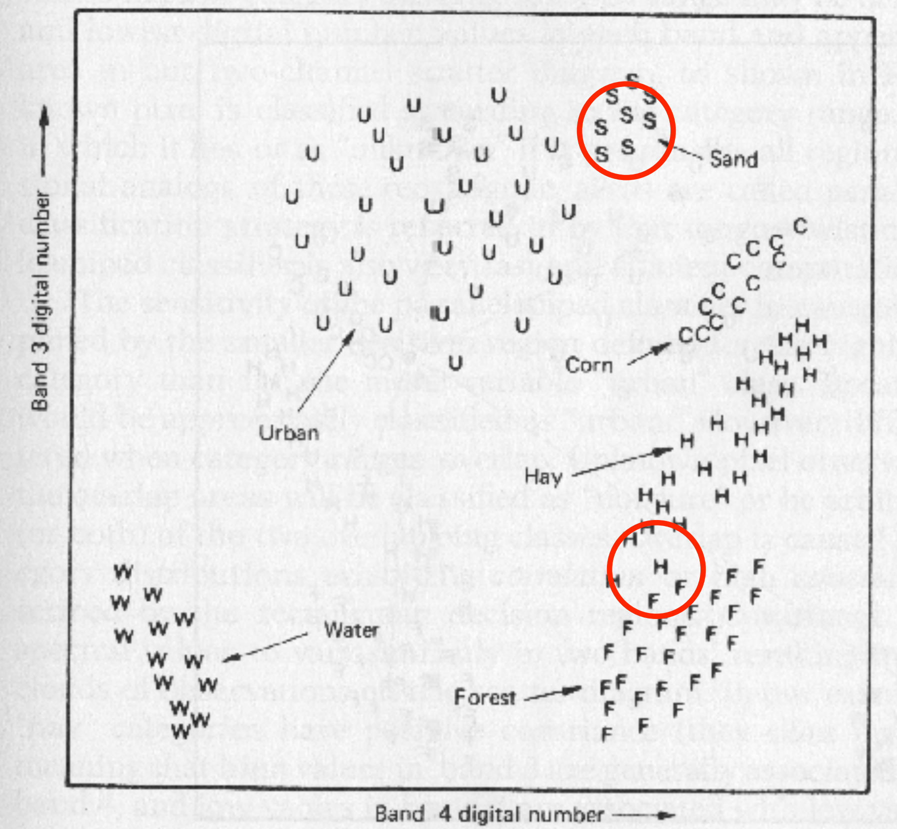


Figure 7.39 Pixel observations from selected training sites plotted on scatter diagram.

Supervised classification: MDM

- Find closest cluster mean for each pixel
- Simple and quick BUT what about points 1, 2?
- i.e. MDM insensitive to variance of clusters
- Can we improve?

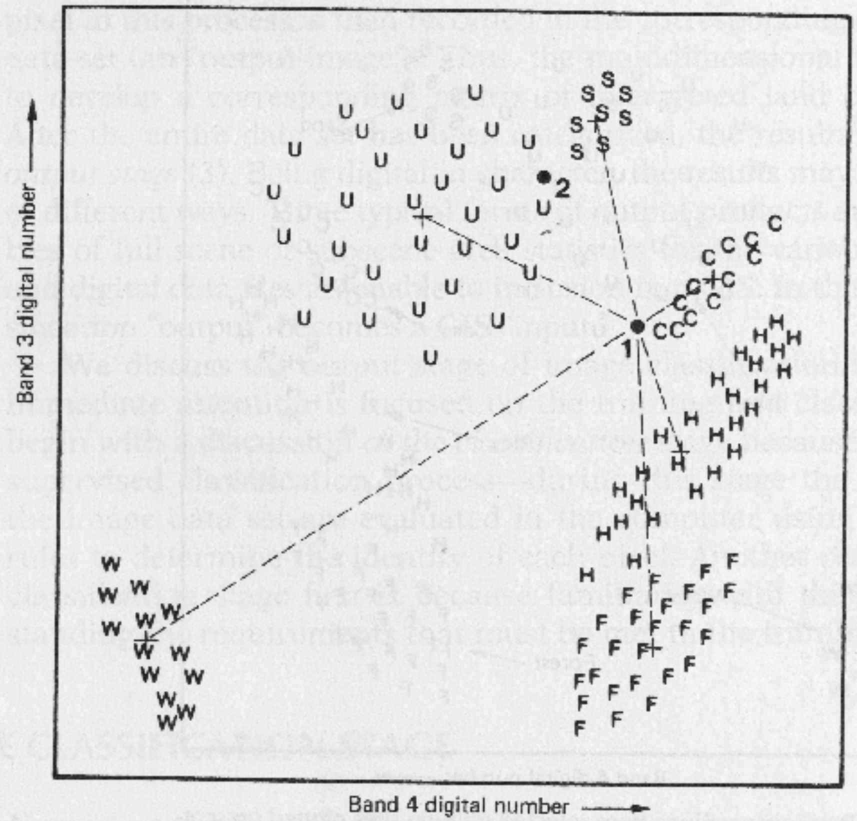


Figure 7.40 Minimum distance to means classification strategy.

Supervised classification: parallelepiped ('box')

- Assign boundaries around the spread of a class in feature space i.e. take account of variance
- typically use minimum/maximum of DN in a particular class to define limits, giving a rectangle in 2D, box in 3D (if we have > 2 bands) etc.
- pixels outside of these regions are unclassified (which is good or bad, depending on what you want!!)
- problems if class regions overlap or if high covariance between different bands (rectangular box shape inappropriate)
  - can modify algorithm by using stepped boundaries with a series of rectangles to partially overcome such problems
- simple and fast technique
- takes some account of variations in the variance of each class

Supervised classification: parallelepiped ('box')

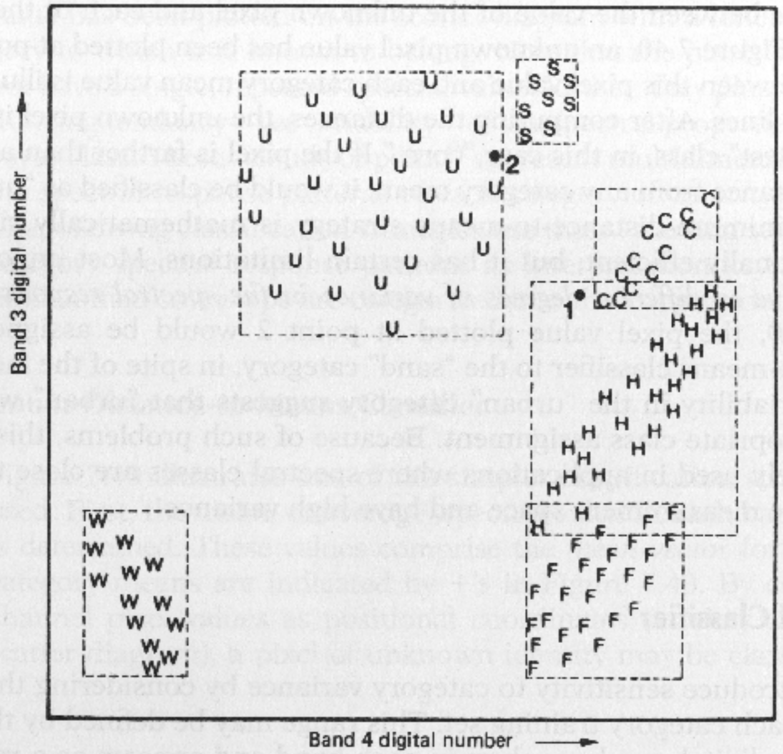


Figure 7.41 Parallelepiped classification strategy.

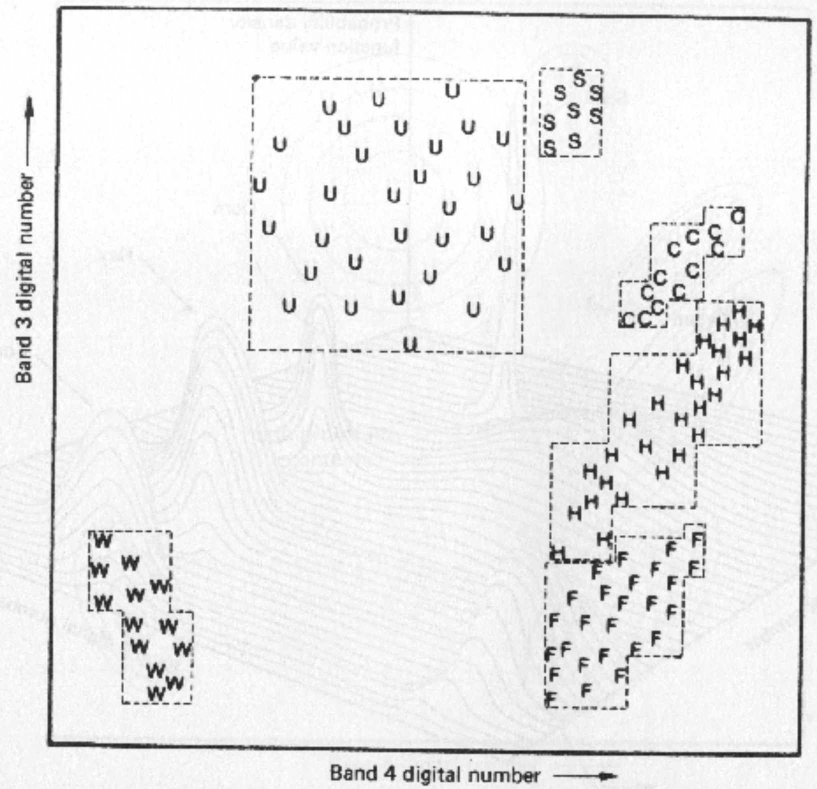


Figure 7.42 Parallelepiped classification strategy employing stepped decision region boundaries.

Simple boxes defined by min/max limits of each training class. But overlaps……..?

…so use stepped boxes

Supervised classification: Gaussian maximum likelihood

- assumes data in a class are (unimodal) Gaussian (normal) distributed
  - class then defined through a mean vector and covariance matrix
  - calculate the probability of a pixel belonging to any class using probability density functions defined from this information
  - we can represent this as equiprobability contours & assign a pixel to the class for which it has the highest probability of belonging to

Supervised classification: Gaussian maximum likelihood

- Now we use probability rather than distance in feature space
- Which class is each pixel "most likely" to belong to??

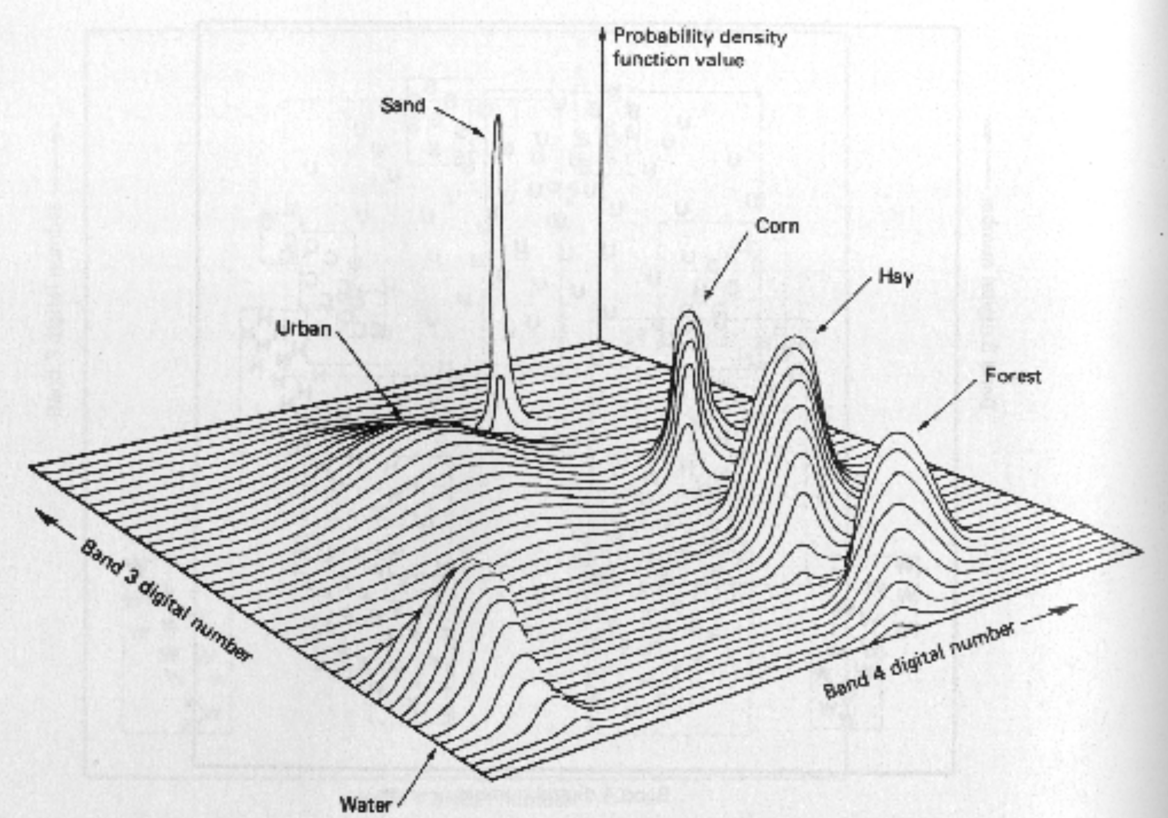


Figure 7.43 Probability density functions defined by a maximum likelihood classifier.

Supervised classification: Gaussian maximum likelihood

- Now pixel 1 correctly assigned to corn class
- Much more sophisticated BUT is computationally expensive compared to distance methods

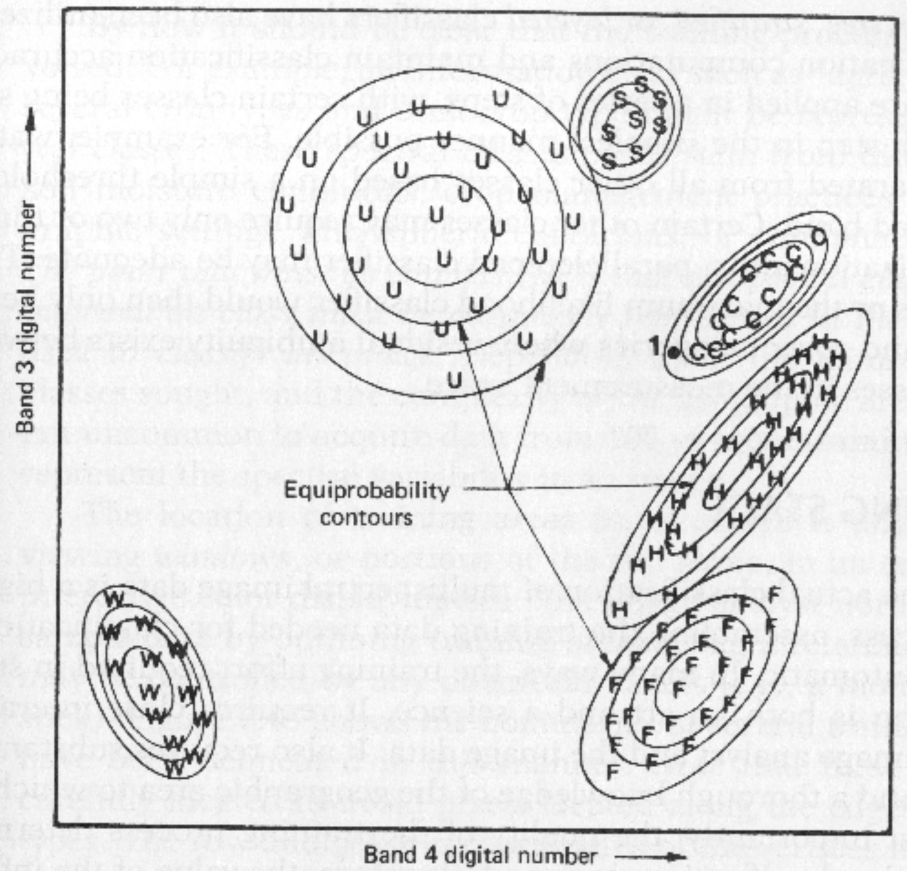


Figure 7.44 Equiprobability contours defined by a maximum likelihood classifier.

Supervised classification: decision tree

- Classify in steps, where the classifier has only to be able to distinguish between two or more classes at each step
  - can combine various types of classifiers as appropriate using such methods

Classification Accuracy

- How do we tell if classification is any good?
  - Classification error matrix (aka confusion matrix or contingency table)
  - Need “truth” data – sample pixels of known classes
    - How many pixels of KNOWN class X are incorrectly classified as anything other than X (<u>errors of omission</u>)?
      - » So-called Type 2 error, or false negative
    - Divide correctly classified pixels in each class of truth data by COLUMN totals (Producer’s Accuracy)
    - How many pixels are incorrectly classified as class X when they should be some other known class (<u>errors of commission</u>)?
      - » So-called Type 1 error, or false positive
    - Divide correctly classified pixels in each class by ROW totals (User’s Accuracy)

Classification Accuracy

TABLE 7.3 Error Matrix Resulting from Classifying Training Set Pixels

Errors of comission for class U

Errors of omission for class U

	Training Set Data (Known Cover Types)[a]						
	W	S	F	U	C	H	Row Total
Classification data							
W	480	0	5	0	0	0	485
S	0	52	0	20	0	0	72
F	0	0	313	40	0	0	353
U	0	16	0	126	0	0	142
C	0	0	0	38	342	79	459
H	0	0	38	24	60	359	481
Column total	480	68	356	248	402	438	1992

Producer's Accuracy

W = 480/480 = 100%
S = 052/068 = 76%
F = 313/356 = 88%
U = 126/248 = 51%
C = 342/402 = 85%
H = 359/438 = 82%

User's Accuracy

W = 480/485 = 99%
S = 052/072 = 72%
F = 313/353 = 87%
U = 126/142 = 89%
C = 342/459 = 74%
H = 359/481 = 75%

Overall accuracy = (480 + 52 + 313 + 126 + 342 + 359)/1992 = 84%

[a]W, water; S, sand; F, forest; U, urban; C, corn; H, hay.

- Can use original training data to test BUT….
- …this only tells us how well the classifier can classify the training areas
- Ideally, use an independent set of samples to give a better 'overall' accuracy estimate

Unsupervised Classification (clustering)

- Little input from user required (few assumptions)
  - BUT means results hard to interpret (may not represent classes we recognise)
- cluster pixels in feature space based on some measure of their proximity
- interpretation of results / assigned classes
  - can be useful, e.g. in picking up variations within what would otherwise be distinguished as a single class e.g. stressed/unstressed crop in a single field)
  - clusters can be of little intrinsic value in themselves
    - e.g. sunlit trees, shaded trees is perhaps not a useful discrimination if one simply wants to classify 'trees', and so clusters may have to be combined

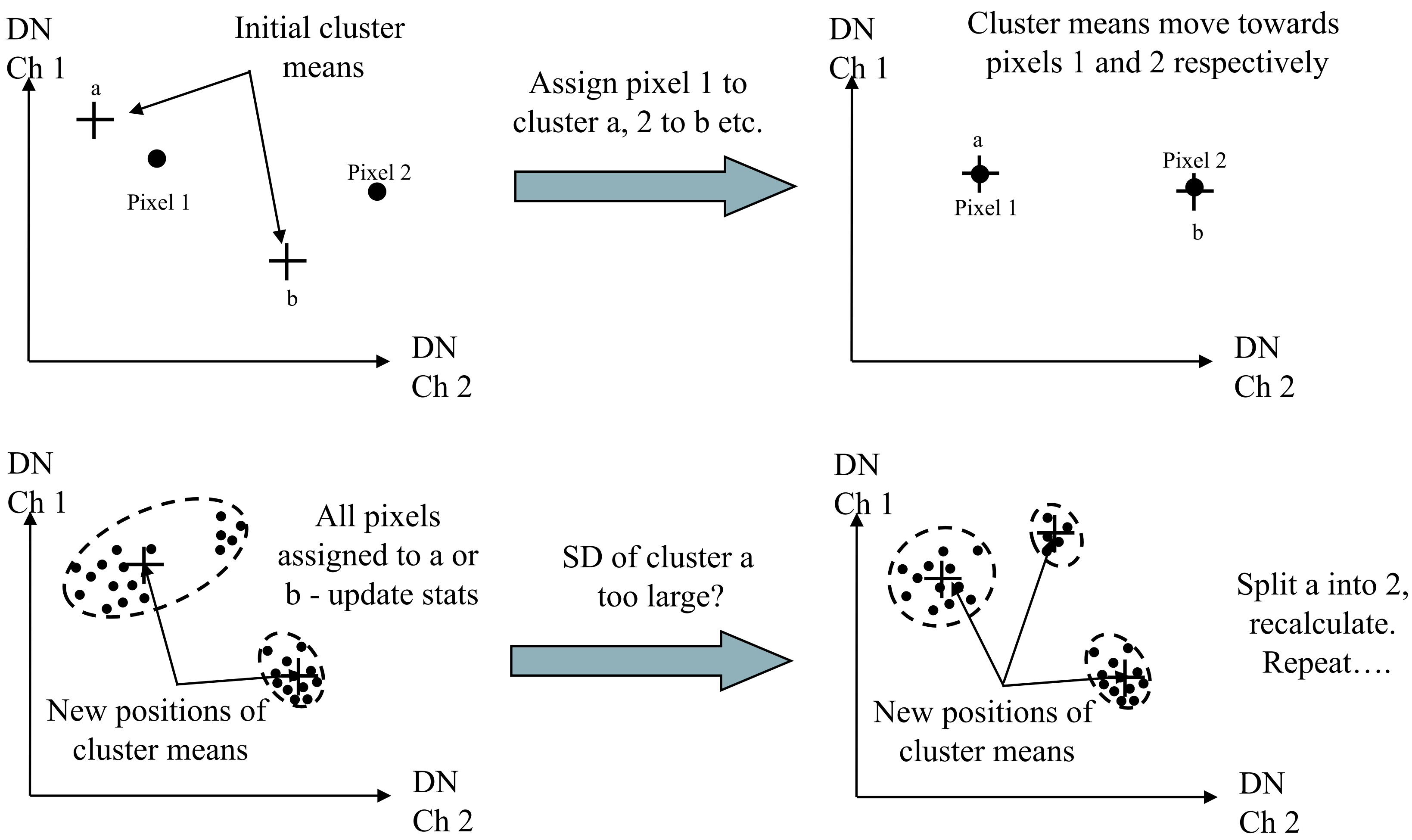
Unsupervised Classification: K-means

- A large number of clustering algorithms exist
- K-means
  - input number of clusters desired
  - algorithm typically initiated with arbitrarily-located 'seeds' for cluster means
  - each pixel then assigned to closest cluster mean
  - revised mean vectors are then computed for each cluster
  - repeat until some convergence criterion is met (e.g. cluster means don't move between iterations)
  - computationally-expensive because it is iterative

Unsupervised classification: ISODATA (Iterative self-organising data analysis) algorithm

- Same as K-means but now we can vary number of clusters (by splitting / merging)
  - Start with (user-defined number) randomly located clusters
  - Assign each pixel to nearest cluster (mean spectral distance)
  - Re-calculate cluster means and standard deviations
  - If distance between two clusters < some threshold, merge them
  - If standard deviation in any one dimension > some threshold, split into two clusters
  - Delete clusters with small number of pixels
  - Re-assign pixels, re-calculate cluster statistics etc. until changes of clusters < some fixed threshold

ISODATA example: 2 classes, 2 bands



Hybrid Approaches

- useful if large variability in the DN of individual classes
- use clustering concepts from unsupervised classification to derive sub-classes for individual classes, followed by standard supervised methods.
- can apply e.g. K-means algorithm to (test) subareas, to derive class statistics and use the derived clusters to classify the whole scene
- requirement that all classes of interest are represented in these test areas
- clustering algorithms may not always determine all relevant classes in an image e.g. linear features (roads etc.) may not be picked-up by the textural methods described above

Postclassification filtering

- The result of a classification from RS data can often appear rather 'noisy'
- Can we aggregate information in some way?
- Simplest & most common way is majority filtering
  – a kernel is passed over the classification result and the class which occurs most commonly in the kernel is used
- May not always be appropriate; the particular method for spatial aggregation of categorical data of this sort depends on the particular application to which the data are to be put
  – e.g. successive aggregations will typically lose scattered data of a certain class, but keep tightly-clustered data

Postclassification filtering