IMAGE DATA ANALYSIS (6CFU)

MODULE OF

REMOTE SENSING

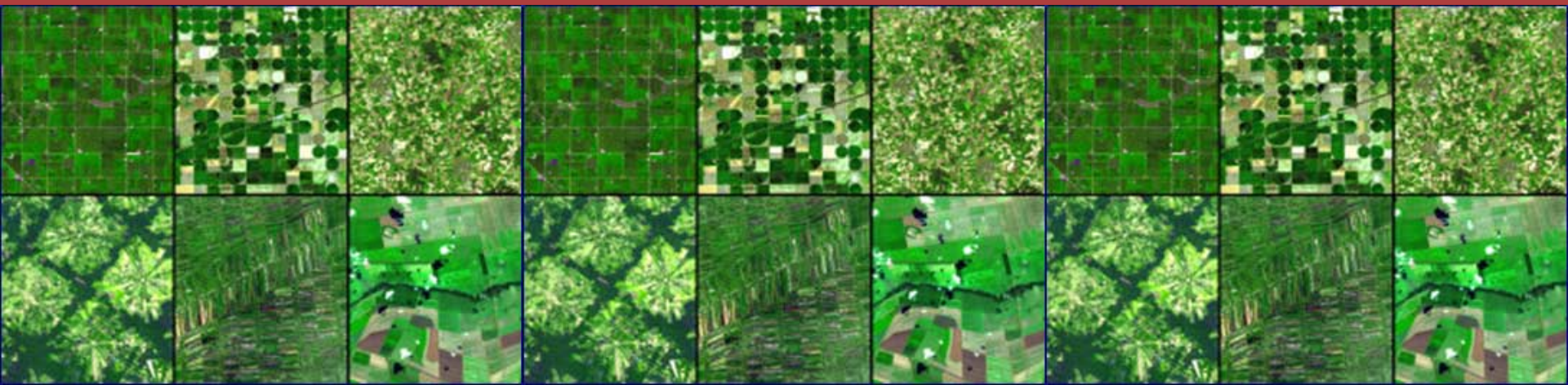
(9 CFU)

A.Y. 2022/23

MASTER OF SCIENCE IN COMMUNICATION TECHNOLOGIES AND MULTIMEDIA
MASTER OF SCIENCE IN COMPUTER SCIENCE, LM INGEGNERIA INFORMATICA

PROF. ALBERTO SIGNORONI

SUPERVISED (STATISTICAL) CLASSIFICATION TECHNIQUES



MAXIMUM LIKELIHOOD CLASSIFICATION

Maximum likelihood classification is one of the most common supervised classification techniques historically used with remote sensing image data and many other fields, and it was the first rigorous algorithm to be employed widely.

In the following, it is developed in a statistically acceptable manner. Although the present approach is sufficient for most practical purposes, more rigorous and more general derivations have been formulated.

We present algorithms used for the supervised classifications of data coming from a **single (imaging) sensor**. When data from a variety of sensors or sources require analysis, more sophisticated *Multisource Classification* tools may be required.

Note: Equations are numbered likewise in Chapter 8 of the RS Image Analysis textbook, i.e. (8.x)

Bayes' Classification

- Let the spectral classes for an image be represented by ω_i , $i = 1, \ldots M$ where M is the total number of classes. In trying to determine the class or category to which a pixel vector $\boldsymbol{x}$ belongs it is strictly the **conditional probabilities**

$$p(\omega_i \,|\boldsymbol{x}),\ i = 1, \ldots M$$

  that **are of interest**.

- The measurement vector $\boldsymbol{x}$ is a column of brightness values for the pixel.
  - It describes the pixel as a point in multispectral (or more in general multicomponent) space with co-ordinates defined by the brightnesses.
  - The probability $p(\omega_i \,|\boldsymbol{x})$ gives the likelihood that the correct class is ω_i for a pixel at position $\boldsymbol{x}$ (in the pattern space).

- Classification is performed according to

$$\boldsymbol{x} \in \omega_i \text{ , if } p(\omega_i \,|\boldsymbol{x}) > p(\omega_j \,|\,\boldsymbol{x}) \quad \text{for all } j \neq i \tag{8.1}$$

  i.e., the pixel at $\boldsymbol{x}$ belongs to class ω_i if $p(\omega_i \,|\,\boldsymbol{x})$ is the largest.

- This intuitive decision rule is a special case of a more general rule in which the decisions can be biased according to different degrees of significance being attached to different incorrect classifications.
  - The general approach is called Bayes' classification and is the subject of the treatment in Appendix F (in the textbook, 4th ed.).

The Maximum Likelihood Decision Rule

- Despite its simplicity, the $p(\omega_i|\boldsymbol{x})$ in (8.1) are unknown.
- Suppose however that sufficient training data is available for each ground cover type.
  - This can be used to estimate a probability distribution for a cover type that describes the chance of finding a pixel from class ω_i , say, at the position $\boldsymbol{x}$ in the pattern space.
  - Later the form of this distribution function will be made more specific. For the moment however it will be retained in general terms and represented by the symbol $p(\boldsymbol{x}|\omega_i)$.
  - The desired $p(\omega_i|\boldsymbol{x})$ in (8.1) and the available $p(\boldsymbol{x}|\omega_i)$ - estimated from training data – are related by Bayes' theorem:

$$p(\omega_i|\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)\, p(\omega_i)/p(\boldsymbol{x}) \qquad (8.2)$$

    where $p(\omega_i)$ is the probability that class ω_i occurs in the image and $p(\boldsymbol{x})$ is the probability of finding a pixel from any class at location $\boldsymbol{x}$.
  - It is of interest to note in passing that:

$$p(\boldsymbol{x}) = \sum_{i=1}^{M} p(\boldsymbol{x}|\omega_i)\, p(\omega_i),$$

    although $p(\boldsymbol{x})$ itself is not important in the following.

- The $p(\omega_i)$ are called a priori or prior probabilities, since they are the probabilities with which class membership of a pixel could be guessed before classification.
- By comparison the $p(\omega_i|\boldsymbol{x})$ are posterior probabilities.

The Maximum Likelihood Decision Rule

$p(\omega_i|x) = p(x|\omega_i) \cdot P(\omega_i)/P(x)$ Baye's theorem $\Rightarrow$ $x \notin \omega_i$, if $p(\omega_i|x) > P(\omega_j|x)\big|_{j \neq i}$ $\therefore$ $x \in \omega_i$, if:

- Using (8.2) it can be seen that the classification rule of (8.1) is: $P(x|\omega_i) \cdot P(\omega_i) > P(x|\omega_j)\, P(\omega_j)\big|_{i \neq j}$

Basically: $P(\omega_i|x)$ = Likelihood pixel, x, is in class, ω_i (hard to know).

$$\boldsymbol{x} \in \omega_i \text{ if } p(\boldsymbol{x}|\omega_i)\, p(\omega_i) > p(\boldsymbol{x}|\omega_j)\, p(\omega_j), \text{ for all } j \neq i \qquad (8.3)$$

$P(x|\omega_i)$ = Likelihood class, ω_i, is in any given pixel, x (easier, given app. training data - i.e. we manually pre-classify training data images and since each image is different we can use that data to get the prob. that a given image vector (spectral value) is of a particular type (class) of ground cover (veg., water, etc.))

where $p(\boldsymbol{x})$ has been removed as a common factor.

- The rule of (8.3) is more acceptable than that of (8.1) **since the $p(\boldsymbol{x}|\omega_i)$ are known from training data**, and it is conceivable that the $p(\omega_i)$ are also known or can be **estimated** from the analyst's knowledge of the image.
- Mathematical convenience results if in (8.3) the definition

$$g_i(\boldsymbol{x}) = \ln \{p(\boldsymbol{x}|\omega_i)\, p(\omega_i)\} = \ln p(\boldsymbol{x}|\omega_i) + \ln p(\omega_i) \qquad (8.4)$$

i.e. this must be true, by the nature of log's

And is more convenient computationally

is used, where ln is the natural logarithm, so that (8.3) is restated as

$$\boldsymbol{x} \in \omega_i \text{ if } g_i(\boldsymbol{x}) > g_j(\boldsymbol{x})\big|_{i \neq j} \qquad (8.5)$$

- This is, with one modification to follow, the decision rule used in maximum likelihood classification; the $g_i(\boldsymbol{x})$ are referred to as **discriminant functions**.

Multivariate Normal Class Models

- At this stage it is assumed that the probability distributions for the classes are of the form of multivariate normal models. This is an assumption, rather than a demonstrable property of natural spectral or information classes; however it leads to mathematical simplifications in the following. Moreover it is one distribution for which properties of the multivariate form are well-known.
- In (8.4) therefore, it is now assumed for N bands that (see Appendix E)

$$p(\boldsymbol{x}|\omega_i) = (2\pi)^{-N/2} |\Sigma_i|^{-1/2} \exp\{-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^t \ \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{m}_i)\} \qquad (8.6)$$

where $\boldsymbol{m}_i$ and Σ_i are the mean vector and covariance matrix of the data in class ω_i. The resulting term $-N/2 \ln (2\pi)$ is common to all $g_i(\boldsymbol{x})$ and does not aid discrimination. Consequently it is ignored and the final form of the discriminant function for maximum likelihood classification, based upon the assumption of normal statistics, is:

$$g_i(\boldsymbol{x}) = \ln p(\omega_i) - \tfrac{1}{2}\ln |\Sigma_i| - \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^t \ \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) \qquad (8.7)$$

- Often the analyst has no useful information about the $p(\omega_i)$, in which case a situation of equal prior probabilities is assumed; as a result $\ln p(\omega_i)$ can be removed from (8.7) since it is then the same for all i. In that case the 1/2 common factor can also be removed leaving, as the discriminant function:

$$g_i(\boldsymbol{x}) = -\ln|\Sigma_i| - (\boldsymbol{x} - \boldsymbol{m}_i)^t \ \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) \qquad (8.8)$$

- Implementation of the maximum likelihood decision rule involves using either (8.7) or (8.8) in (8.5). There is a further consideration however concerned with whether any of the available labels or classes is appropriate. This relates to the use of thresholds as discussed later on.

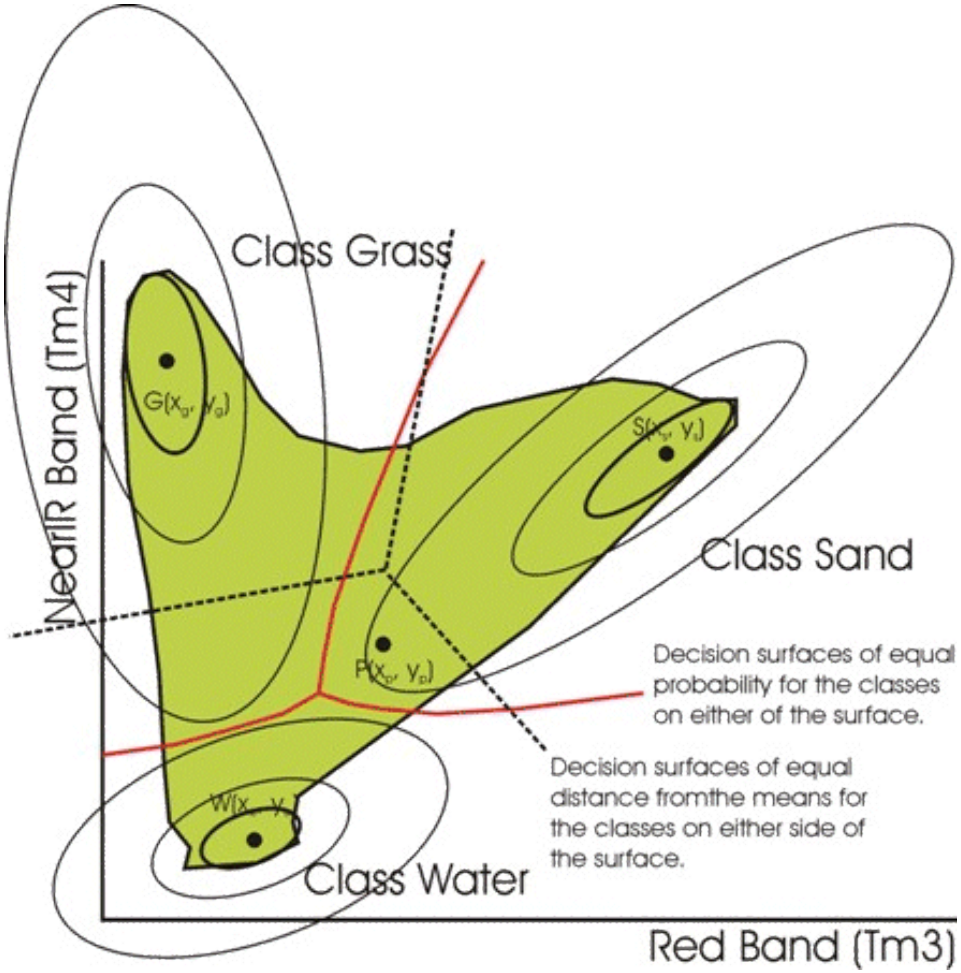
Decision Surfaces

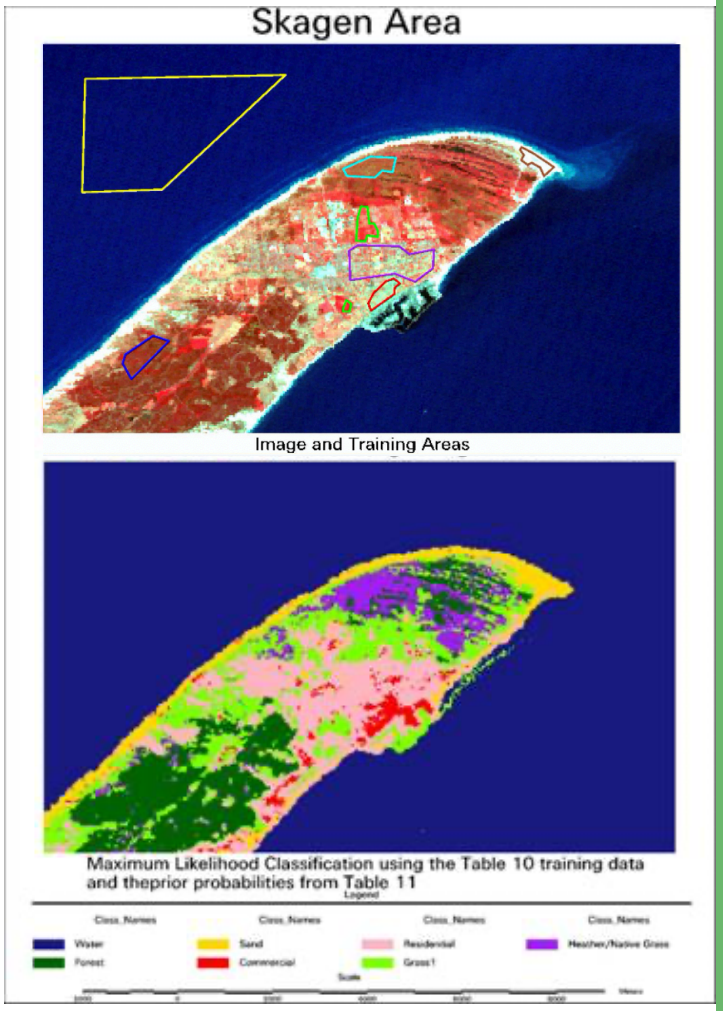
- The determination of the *essential shapes* of the surfaces that separate one class from another, in the multispectral domain is of value for assessing the capabilities of the maximum likelihood decision rule.
- These surfaces, albeit implicit (i.e. solution of an equation), can be devised in the following manner.
  - Spectral classes are defined by those regions in multispectral space where their discriminant functions are the largest.
  - Clearly these regions are separated by surfaces where the discriminant functions for adjoining spectral classes are equal.
  - The implicit *i*-th and *j*-th spectral classes are separated therefore by the surface

$$g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0.$$

  - This is referred to as a ***decision surface*** since, if all the surfaces separating spectral classes are known, decisions about the class membership of a pixel can be made on the basis of its position relative to the complete set of surfaces.
- The construction $(\boldsymbol{x} - \boldsymbol{m}_i)^t \, \Sigma_i^{-1} (\boldsymbol{x} - \boldsymbol{m}_i)$ in (8.7) and (8.8) is a quadratic function of $\boldsymbol{x}$.
  - Consequently the **decision surfaces** implemented by maximum likelihood classification **are quadratic** and thus take the form of parabolas, circles and ellipses.

Decision Surfaces (example)

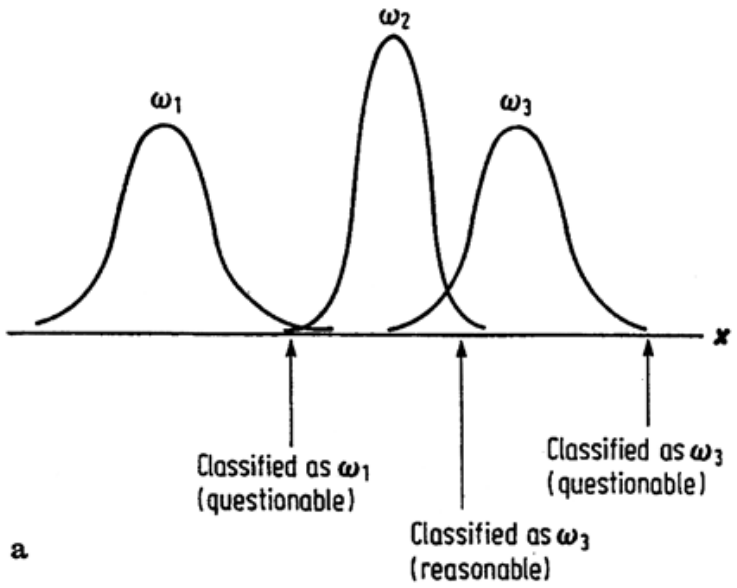


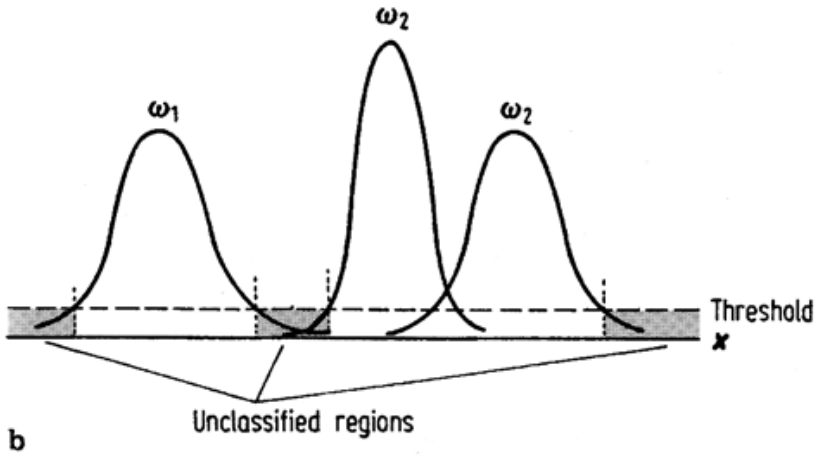


Thresholds

Using Maximum Likelihood Decision Rules

- ~~With the above method~~, pixels at every point in multispectral space will be classified into one of the available classes ω_i, irrespective of how small the actual probabilities of class membership are.
  - This is illustrated for one dimensional data in Figure a.
  - Poor classification can result as indicated. Such situations can arise if spectral classes (between 1 and 2 or beyond 1 and 3) have been overlooked or, if knowing other classes existed, enough training data was not available to estimate the parameters of their distributions with any degree of accuracy (see the following).
  - In situations such as these it is sensible to apply thresholds to the decision process in the manner depicted in Figure b: *pixels which have probabilities for all classes below the threshold are not classified*.





Thresholds

- In practice, thresholds are applied to the discriminant functions and not the probability distributions, since the latter are never actually computed. With the incorporation of a threshold therefore, the decision rule of (8.5) becomes

$$\boldsymbol{x} \in \omega_i \quad \text{if} \quad g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \quad \text{for all} \quad j \neq i \tag{8.9a}$$

$$\text{and} \quad g_i(\boldsymbol{x}) > T_i \tag{8.9b}$$

where T_i is the threshold seen to be significant for spectral class ω_i. It is now necessary to consider how T_i can be estimated. From (8.7) and (8.9b) a classification is acceptable if

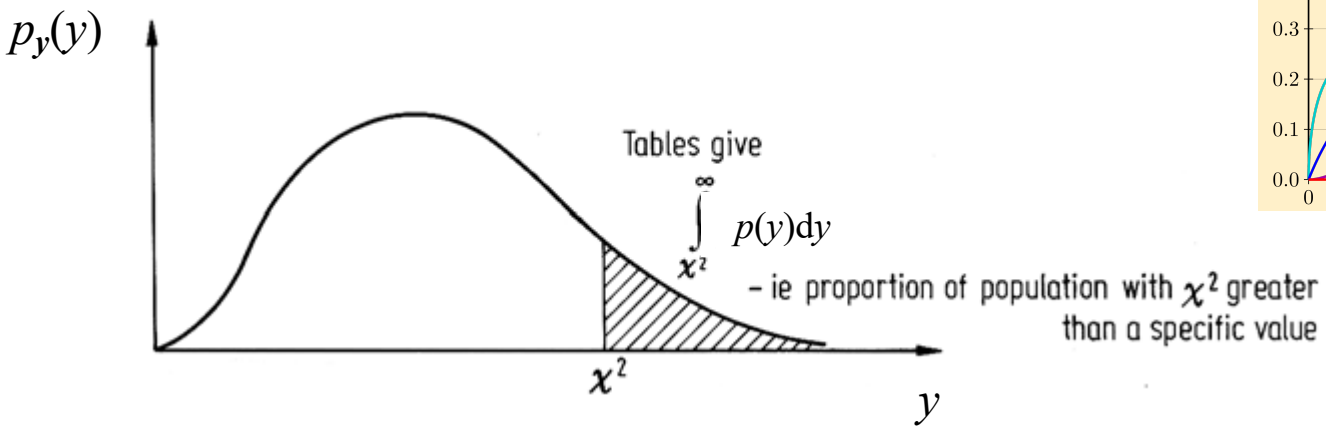
$$\ln\, p(\omega_i) - \tfrac{1}{2} \ln\, |\Sigma_i| - \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^t\ \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) > T_i$$

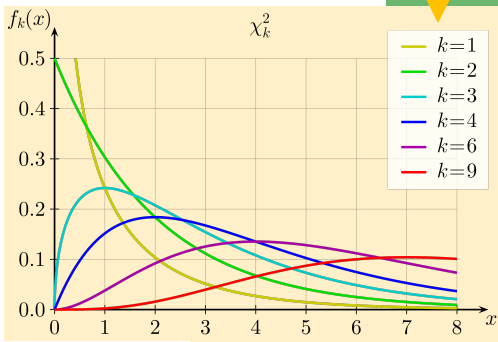
i.e.

$$\underbrace{(\boldsymbol{x} - \boldsymbol{m}_i)^t\ \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{m}_i)}_{\boldsymbol{y}_i} < -2T_i - \ln\, |\Sigma_i| + 2 \ln\, p(\omega_i) \tag{8.10}$$

Thresholds

- The random variable y_i on the left hand side of (8.10) has a χ^2 distribution with N degrees of freedom, if $\boldsymbol{x}$ is (assumed to be) distributed normally (Swain and Davis, 1978).
  - N is the dimensionality of the multispectral space (N=4 in this example referred to Landsat MSS). As a result χ^2 tables can be consulted to determine that value below which a desired percentage of pixels will exist (noting that larger values of that quadratic form correspond to pixels lying further out in the tails of the normal probability distribution).
  - This is depicted in Figure:





https://www.statology.org/chi-square-distribution-table/

- As an example of how this is used
  - consider the need to choose a threshold such that 95% of all pixels in a class will be classified (i.e. such that the 5% least likely pixels for each spectral class will be rejected).
  - χ^2 tables show that 95% of all pixels have $\boldsymbol{y}$ values (in Figure) less than 9.488.
  - Thus, from (8.10) $T_i = -4.744 - \frac{1}{2}\ln|\Sigma_i| + \ln\, p(\omega_i)$ which thus can be calculated from a knowledge of the prior probability and covariance matrix of the i-th spectral class.

	P					
DF	0.995	0.975	0.2	0.1	0.05	0.0
1	.0004	.00016	1.642	2.706	3.841	5.
2	0.01	0.0506	3.219	4.605	5.991	7.
3	0.0717	0.216	4.642	6.251	7.815	9.
4	0.207	0.484	5.989	7.779	9.488	11.
5	0.412	0.831	7.289	9.236	11.07	12.
6	0.676	1.237	8.558	10.645	12.592	14.

Number of Training Pixels Required for Each Class

- **Sufficient training pixels for each spectral class must be available** to allow reasonable estimates to be obtained of the elements of the class conditional mean vector and covariance matrix.
  - For an *N* dimensional multispectral space, the class covariance matrix Σ_i is symmetric of size $N \times N$. It has, therefore, *½N(N + 1)* distinct elements that need to be estimated from the training data.
  - To avoid the matrix being singular (we need its inverse) at least *N(N + 1)* independent samples are needed (to guarantee full rank square matrix, thus invertibility).
    - Fortunately, each *N* dimensional pixel vector in fact contains *N* samples (one in each waveband); thus the minimum **number of independent training pixels required for each spectral class is (N+1)**.
    - Because of the **difficulty in assuring independence of the pixels**, usually many more than this minimum number is selected.
  - Swain and Davis (1978) recommend as a practical minimum that **10*N*** training pixels per spectral class be used, with as many as 100*N* per class if possible.
    - For data with low dimensionality (say up to 5 or 6 bands) those numbers can usually be achieved without problems, but for hyperspectral datasets finding enough training pixels per class can be extremely difficult (this problem will be reconsidered at a later time).

A Simple Illustration

- As an example of the use of maximum likelihood classification, the segment of Landsat multispectral scanner image shown in Figure is chosen.
  - This is a 256×276 pixel array of image data in which four broad ground cover types are evident.
  - These are water, fire burn, vegetation and “developed” land (urban).
  - Suppose we want to produce a thematic map of these four cover types in order to enable the area and extent of the fire burn to be evaluated.
- The first step is to choose training data.
  - For such a broad classification, suitable sets of training pixels for each of the four classes are easily identified visually in the image data.
  - The Figure also shows the locations of four training fields used for this purpose.
  - Sometimes, to obtain a good estimate of class statistics it may be necessary to choose several training fields for the one cover type, located in different regions of the image.

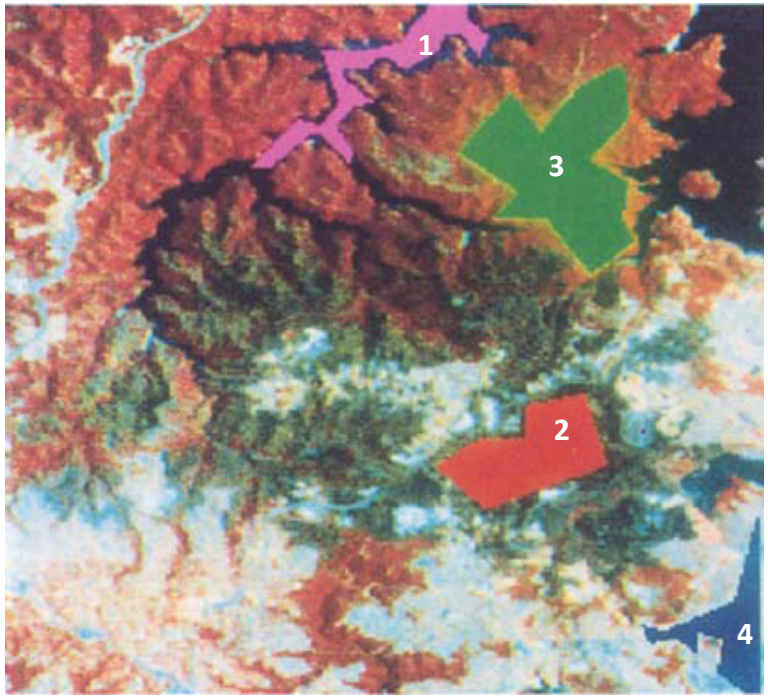


Image segment to be classified, consisting of a mixture of natural vegetation, waterways, urban development and vegetation damaged by fire. Four training regions are identified in solid colour. These are water (1), vegetation (2), fire burn (3) and urban (4 in the bottom right hand corner). Pixels from these were used to generate the signatures in the next Table

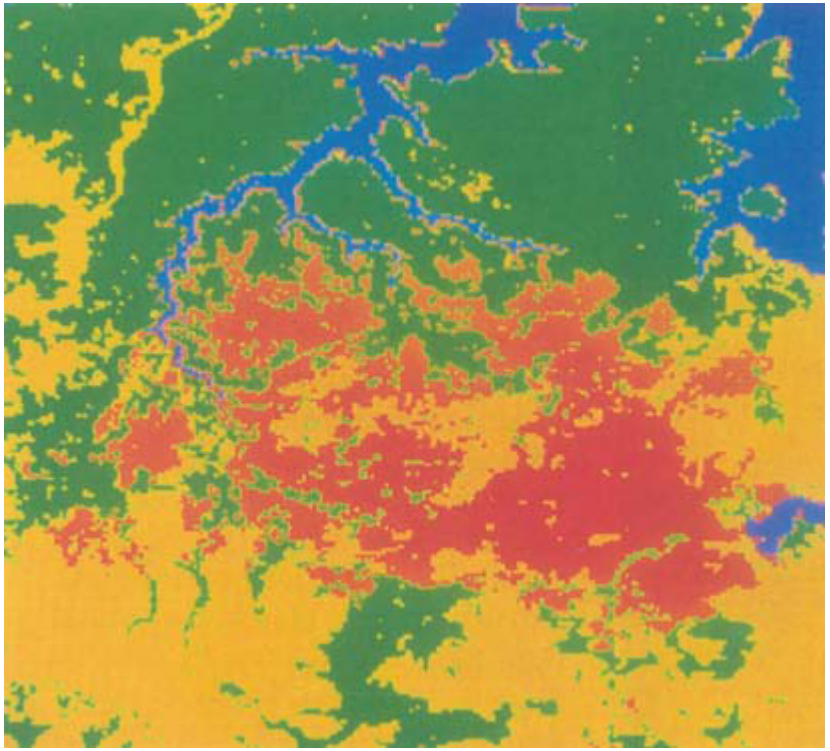
A Simple Illustration

- The *four-band signatures* for each of the four classes, as obtained from the training fields, are given in the Table.
  - The mean vectors can be seen to agree generally with known spectral reflectance characteristics of the cover types.
  - Also the class variances (diagonal elements in the covariance matrices) are small for water as might be expected but on the large side for the developed/urban class, indicative of its heterogeneous nature.
  - Numbers are on a scale of 0 to 255 (8 bit)

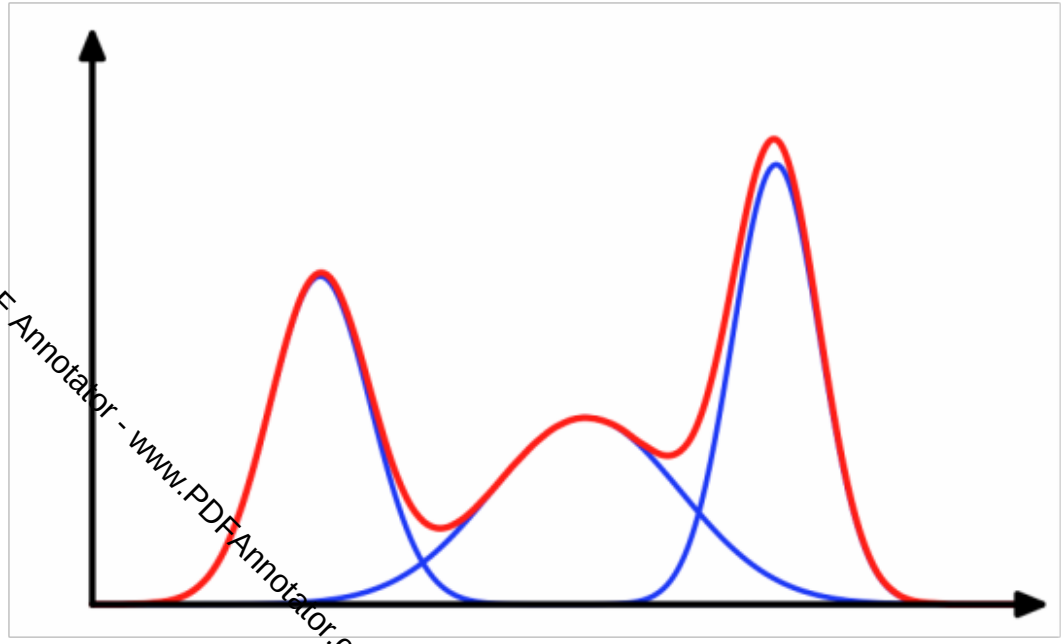
Class	Mean vector	Covariance matrix			
Water	44.27	14.36	9.55	4.49	1.19
	28.82	9.55	10.51	3.71	1.11
	22.77	4.49	3.71	6.95	4.05
	13.89	1.19	1.11	4.05	7.65
Fire burn	42.85	9.38	10.51	12.30	11.00
	35.02	10.51	20.29	22.10	20.62
	35.96	12.30	22.10	32.68	27.78
	29.04	11.00	20.62	27.78	30.23
Vegetation	40.46	5.56	3.91	2.04	1.43
	30.92	3.91	7.46	1.96	0.56
	57.50	2.04	1.96	19.75	19.71
	57.68	1.43	0.56	19.71	29.27
Developed (urban)	63.14	43.58	46.42	7.99	−14.86
	60.44	46.42	60.57	17.38	−9.09
	81.84	7.99	17.38	67.41	67.57
	72.25	−14.86	−9.09	67.57	94.27

A Simple Illustration

- Using these signatures in a maximum likelihood algorithm to classify the four bands of the previous image, the thematic map shown in Figure is obtained.
  - Blue represents water, red is fire damaged vegetation, green is natural vegetation and yellow is urban development
- Table: the four classes by area.
  - Note that there are no unclassified pixels, since no threshold was used in the labeling process.
  - The area estimates are obtained by multiplying the number of pixels per class by the effective area of a pixel.
  - In the case of the Landsat 2 multispectral scanner the pixel size was 0.4424 hectares.



Class	No. of pixels	Area (ha)
Water	4830	2137
Fireburn	14182	6274
Vegetation	28853	12765
Developed (urban)	22791	10083



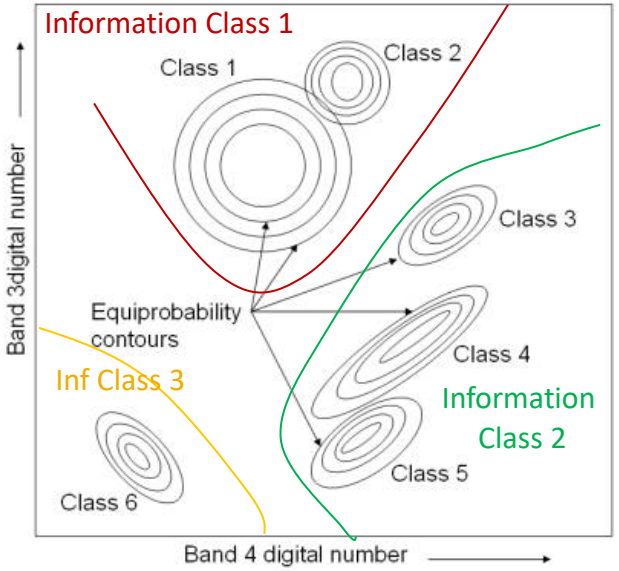
GAUSSIAN MIXTURE MODELS

Should begin here by clarifying difference between a spectral class and information class.

Gaussian Mixture Models

- Mixture modelling, concerns modelling a statistical distribution by a mixture (or weighted sum) of other distributions.
- Until now we derived classification methods associated to information classes corresponding to single spectral classes which in turn can be modelled by multidimensional normal distribution
- However, as we already observed, single information classes can be composed of sets of (adjacent) spectral classes (see figure)

  - Thus, in practice in order to represent the data effectively more than one normally distributed spectral class is required to model properly the distribution of pixel vectors in a given information class.
  - One of the challenges to successful image classification is to find an acceptable set of spectral classes for each information class.
  - Later we will se that clustering algorithms can be used for that purpose, and indeed they can be applied very successfully to that end.
  - Another, more theoretically appealing approach is to try to **learn the mixture of spectral classes for each information class from the available training data**, as in the following.



Gaussian Mixture Models

- If we assume that a given information class is composed of a number of unimodal normally distributed spectral classes, then it is natural to attempt to devise a (information) class model of the form

$$f(\boldsymbol{x}) = \sum_{c=1}^{C} \alpha_c \, p(\boldsymbol{x} | \boldsymbol{m}_c, \Sigma_c)$$

  - where $\boldsymbol{m}_c$ and Σ_c are the mean vector and covariance matrix of the c^{th} spectral class conditional normal distribution;
  - the α_c are weighting parameters (which sum to unity) such that the mixture model expressed by $f(\boldsymbol{x})$ fits the available training data.
  - The total number of spectral class components is C.

- We have to estimate the set of parameters $\{C, \alpha_c, \boldsymbol{m}_c, \Sigma_c\}$ from the training data, and that is a considerable challenge in practice.
  - Kuo and Landgrebe (2002) show how this can be achieved.
  - A good treatment con be found in C.M. Bishop, *Pattern Rec. and Machine Learning*, 2006
  - See also our textbook (J.A.Richards, *Remote Sens. Dig. Imag. An.*, fifth edition) for details.

MINIMUM DISTANCE AND PARALLELEPIPED CLASSIFICATION

Simpifications (degenerations) of the Maximum Likelihood approach

The Case of Limited Training Data

- The *effectiveness* of maximum likelihood classification depends upon reasonably accurate estimation of the mean vector $\boldsymbol{m}$ and the covariance matrix Σ for each spectral class
  - Hence a sufficient number of training pixels for each classes is necessary.
  - In cases where this is not so, inaccurate parameter estimates lead to poor classification.
- When the number of training samples per class is limited it can be more effective to resort to a classifier that does not make use of covariance information but instead depends only upon the mean positions of the spectral classes which can be more accurately estimated than covariances.
- The so-called **minimum distance classifier**, or more precisely, *minimum distance to class-means* classifier, is such an approach:
  - here **training** data is used only to determine (to learn) **class means**;
  - classification is then performed by **placing a pixel in the class of the nearest mean**.
- The minimum distance algorithm is also attractive since **it is a faster technique** than maximum likelihood classification.
  - However, because it does not use covariance data *it is not as flexible* as the latter.
  - Since covariance data is not used in the minimum distance technique class models are symmetric in the spectral domain. *Elongated classes therefore will not be well modeled*:
    - therefore, *several spectral classes (aligned in space) may need to be used to model elongated classes* with this algorithm, where just one might be suitable for maximum likelihood classification.

Minimum distance classifier: the Discriminant Function

- The *discriminant function* for the minimum distance classifier is developed as follows.

Suppose $\boldsymbol{m}_i, i = 1, \ldots M$ are the means of the M classes determined from training data, and $\boldsymbol{x}$ is the position of the pixel to be classified. Compute the set of squared Euclidean distances of the unknown pixel to each of the class means, defined in vector form as

$$\begin{aligned} \mathrm{d}(\boldsymbol{x}, \boldsymbol{m}_i)^2 &= (\boldsymbol{x} - \boldsymbol{m}_i)^t (\boldsymbol{x} - \boldsymbol{m}_i) \\ &= (\boldsymbol{x} - \boldsymbol{m}_i) \cdot (\boldsymbol{x} - \boldsymbol{m}_i), i = 1, \ldots M \end{aligned}$$

Expanding the product gives

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

Classification is performed on the basis of

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

Note that $\boldsymbol{x} \cdot \boldsymbol{x}$ is common to all $\mathrm{d}(\boldsymbol{x}, \boldsymbol{m}_j)^2$ and thus can be removed. Moreover, rather than classifying according to the smallest of the remaining expressions, the signs can be reversed and classification performed on the basis of

$$\boldsymbol{x} \in \omega_i \quad \text{if} \quad g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \quad \text{for all} \quad j \neq i \tag{8.11a}$$

where

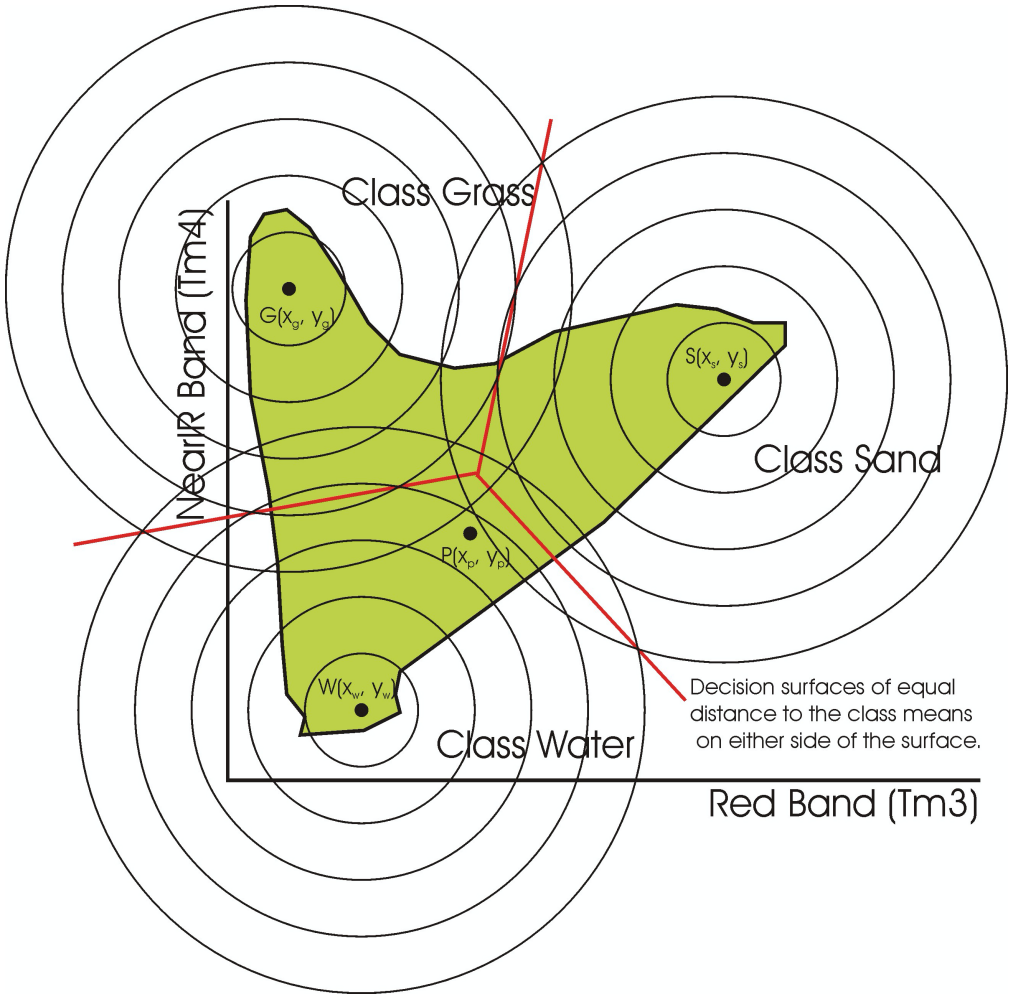
$$g_i(\boldsymbol{x}) = 2\boldsymbol{m}_i \cdot \boldsymbol{x} - \boldsymbol{m}_i \cdot \boldsymbol{m}_i, \quad \text{etc.} \tag{8.11b}$$

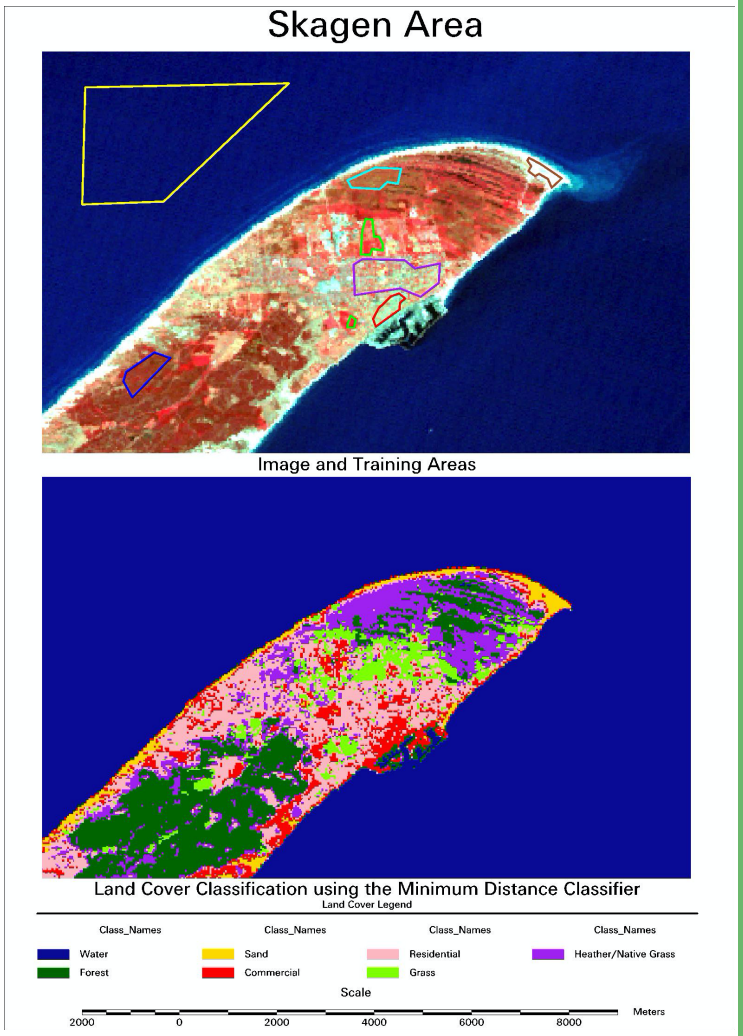
- Equation (8.11b) defines the discriminant function for the minimum distance classifier.

Remarks on Minimum Distance Classification

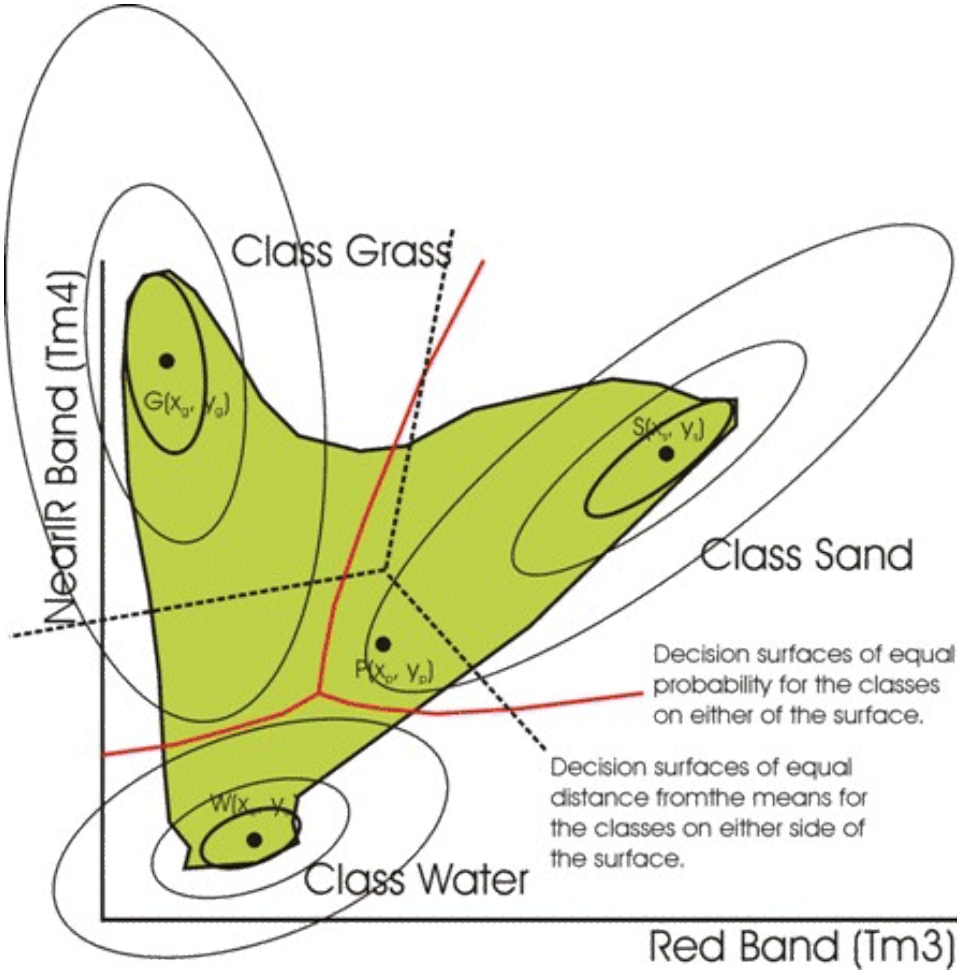
- **Decision Surfaces:**
  - the surface between the *i*th and *j*th spectral classes is given by $g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0$
  - Substituting from (8.11b) gives $2(\boldsymbol{m}_i - \boldsymbol{m}_j) \cdot \boldsymbol{x} - (\boldsymbol{m}_i \cdot \boldsymbol{m}_i - \boldsymbol{m}_j \cdot \boldsymbol{m}_j) = 0$
  - This defines a linear surface – i.e. an hyperplane in more than three dimensions. In contrast therefore to maximum likelihood classification in which the decision surfaces are quadratic and therefore more flexible, the decision surfaces for minimum distance classification are linear and more restricted.
- The higher order decision surface possible with maximum likelihood classification renders it more powerful for partitioning multispectral space than the linear surfaces for the minimum distance approach.
  - Nevertheless, as noted earlier, minimum distance classification is of value when the number of training samples is limited and, in such a case, can lead to better accuracies than the maximum likelihood procedure.
  - However, when class covariance is dominated by systematic (hyperspherical) ***noise*** rather than by natural spectral spreads of the individual spectral classes, there is no advantage in maximum likelihood procedures.
- **Thresholds:**
  - Thresholds can be applied to minimum distance classification by ensuring that not only is a pixel closest to a candidate class but also that it is within a prescribed distance of that class. Such a technique is used regularly.
  - Often the distance threshold is specified according to a number of standard deviations from a class mean.

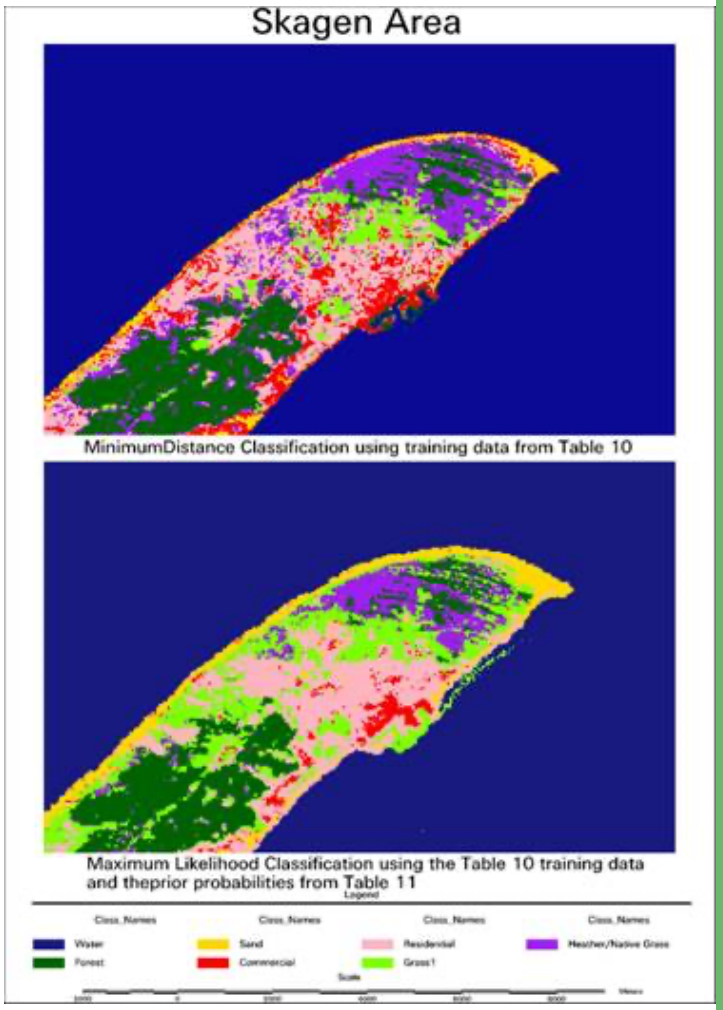
Decision Surfaces (example)





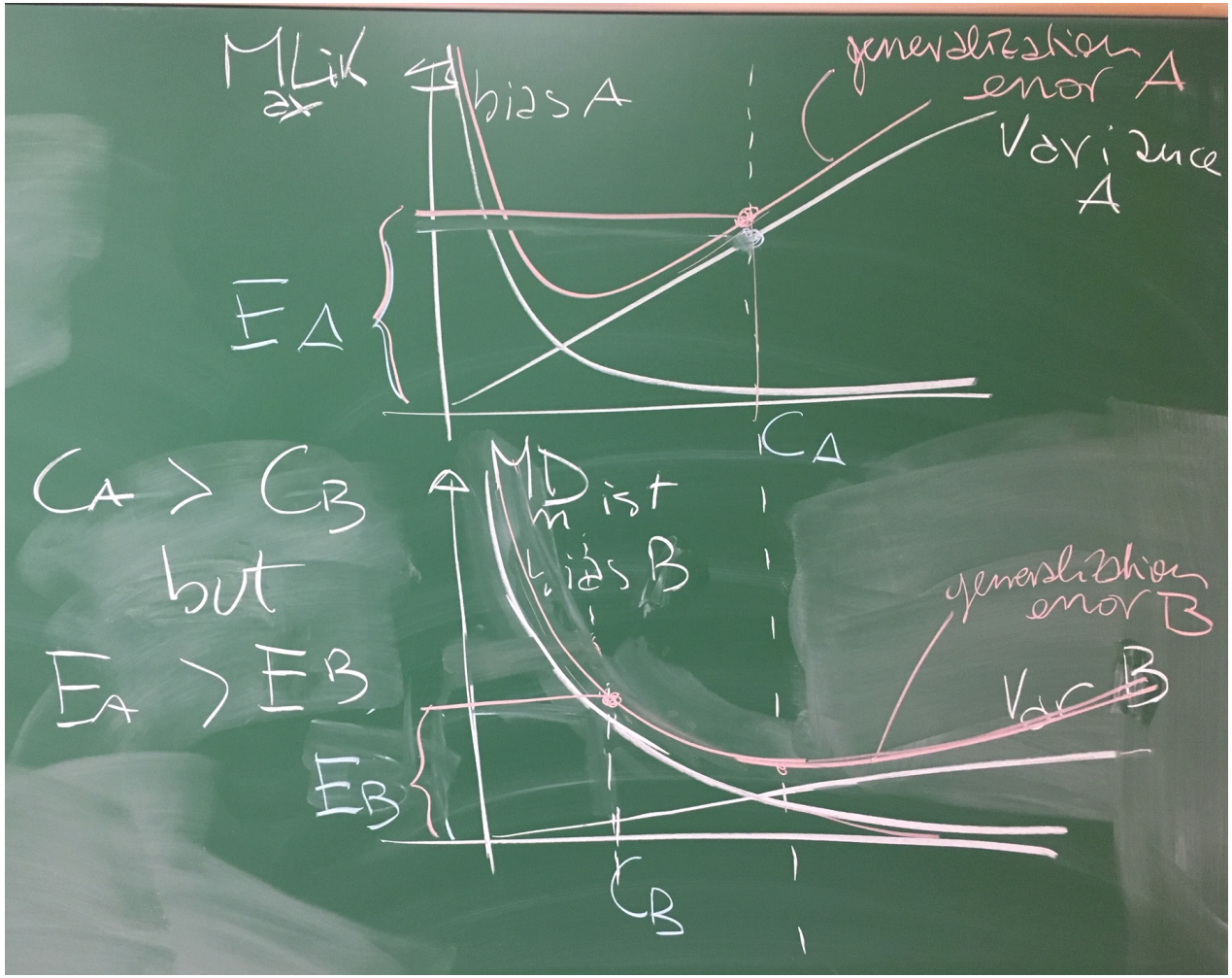
Decision Surfaces (example)





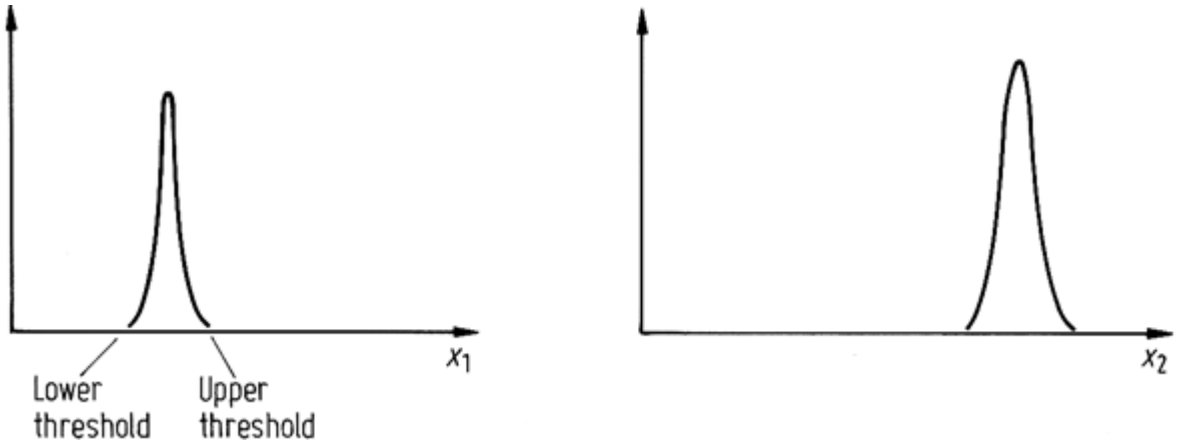
Remarks on Minimum Distance Classification

- Considerations about Bias and Variance in the selection of the right model **given a limited size of the training set** (with MinDist we can work at a lower variance, for ML capacity is too high for the samples we have, both are far from the optimum but MinDist can operate at lower global error with a better model complexity/capacity vs dataset dimension tradeoff)



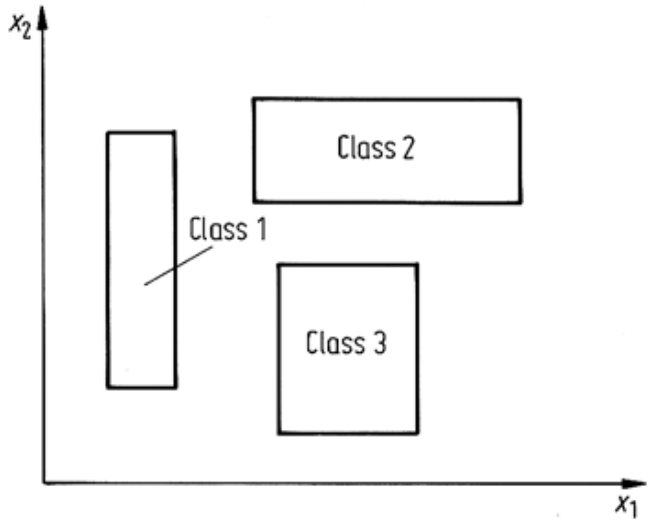
Parallelepiped Classification

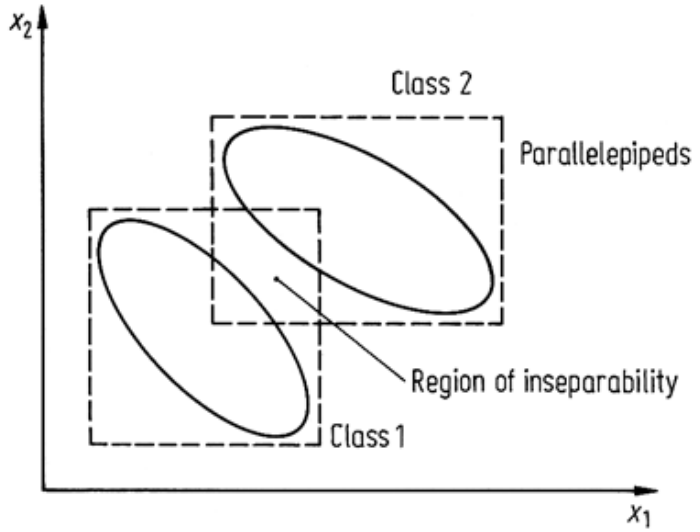
- The **parallelepiped classifier** is a *very simple* supervised classifier that is, in principle, *trained by inspecting histograms* of the individual spectral components of the available training data.
  - Suppose, for example, that the histograms of one particular spectral class for two-dimensional data are as shown in Figure.
  - Then *the upper and lower significant bounds on the histograms are identified* and used to describe *the brightness value range* for each band for that class.
  - Together, the range in all bands describes a *multidimensional box or parallelepiped*.
  - If, on classification, pixels are found to lie in such a parallelepiped they are labeled as belonging to that class.



Parallelepiped Classification

- A two-dimensional pattern space might therefore be segmented as shown in Figure (left).
- While the parallelepiped method is, in principle, a particularly simple classifier to train and use, it has *several drawbacks*.
  - One is that there can be *considerable gaps* between the parallelepipeds; pixels in those regions *will not be classified*.
    - By comparison the minimum distance and maximum likelihood classifiers will label all pixels in an image, unless thresholding methods are used.
  - Another limitation is that *prior probabilities of class membership are not taken account of*; nor are they for minimum distance classification.
  - Finally, for correlated data *there can be overlap of the parallelepipeds* since their sides are parallel to the spectral axes.
    - Consequently, there is *some data that cannot be separated*, as illustrated in Figure (right).





Classification Time Comparison of the Classifiers

- Of the three classifiers commonly used with remote sensing image data
  - the **parallelepiped** procedure is the fastest in classification since only comparisons of the spectral components of a pixel with the spectral dimensions of the parallelepipeds are required.
  - For the **minimum distance** classifier, the discriminant function in (8.11b) requires evaluation for each pixel. In practice $2\boldsymbol{m}_i$ and $\boldsymbol{m}_i \cdot \boldsymbol{m}_i$ would be calculated beforehand, leaving N multiplications and N additions to check the potential membership of a pixel to one class, where N is the number of components in $\boldsymbol{x}$.
  - By comparison, evaluation of the discriminant function for **maximum likelihood** classification in (8.7) requires $N^2 + N$ multiplications and $N^2 + 2N + 1$ additions, to check one pixel against one class, given that

$$-\frac{1}{2} \ln |\Sigma_i| + \ln\, p(\omega_i)$$

    would have been calculated beforehand. Ignoring additions by comparison to multiplications, the maximum likelihood classifier takes $N + 1$ times as long as the minimum distance classifier to perform a classification.
- It is also significant to note that classification time, and thus cost, increases quadratically with number of spectral components for the maximum likelihood classifier but only linearly for minimum distance and parallelepiped classification.
  - This has particular relevance to feature reduction (as we will see).

CONTEXT CLASSIFICATION

The Concept of Spatial Context

- The classifiers treated so far are often referred to as ***point or pixel-specific classifiers***
  - in that they label a pixel on the basis of its spectral properties alone, with no account taken of how any neighbouring pixels are labelled.
- Yet, in any real image, **adjacent pixels are related or correlated**,
  1. because **imaging sensors** acquire significant portions of energy from adjacent pixels (*point spread function* effect)
  2. because **cover types** usually occur on a region that is large compared with the pixel size (this is especially true for satellite Remote Sensing imaging).
     - In an agricultural area, for example, if a particular image pixel represents wheat it is highly likely that its neighbouring pixels will also be wheat.
- This knowledge of neighbourhood relationships is a *rich source of information* that is not exploited in traditional (pixel-specific) classifiers (the ones already seen and the ones we will see, except Convolutional Neural Networks).
  - Here we consider the importance of **spatial context** and see the benefit of taking it into account when making classification decisions:
    - Exploitation of the spatial information, e.g. by **statistical modeling** of the neighbour of pixels (this is maybe an option we have when no other criteria or prior knowledge are available, e.g. medical imaging atlases or other kind of higher level statistical models)
    - Thematic/Classification map improvement, by helping to remove scattered pixel labelling errors that might result from noisy data, or unusual classifier performance

The Concept of Spatial Context

- Classification methods that ponder the labelling of neighbours when seeking to determine the most appropriate class for a pixel are said to be **context sensitive**, or simply **context classifiers**.
  - They attempt to develop a thematic map that is consistent both spectrally and spatially.
- In Remote Sensing the degree to which adjacent pixels are strongly correlated will depend on

1. the scale of natural and cultural regions on the earth's surface.
   - Adjacent pixels over an agricultural region will be strongly correlated, whereas for the same sensor, adjacent pixels over a busier, urban region would not show strong correlation.
2. the spatial resolution of the sensor
   - For a given area, neighbouring Landsat MSS pixels, being larger (spatial resolution 60 m), may not demonstrate as much correlation as adjacent SPOT HRV pixels (20 m).
   - In general terms, context classification techniques usually warrant consideration when processing *higher resolution imagery*.

- More in general the presence of **acquisition noise** can be a source of spatial inconsistency in the classification maps which can be partially recovered by context sensitive classifiers

- **Three possible approaches**:
  - image **pre**-processing (denoising, spatial channel, super-pixel oversegmentation,...)
  - **post** processing (classification map smoothing)
  - "**contextual**": embedded in the (statistical) model used for classification

Context Classification by Image Pre-processing

- Perhaps the simplest method for exploiting spatial context is to **process the image data before classification** in order to *modify or enhance its spatial properties*.
  - A median filter, for example, will help in reducing *salt and pepper noise* that would lead to inconsistent class labels.
  - The application of simple averaging filters (possibly with edge preserving thresholds) can be used to *impose a degree of homogeneity* among the brightness values of adjacent pixels thereby increasing the chance that neighbouring pixels may be given the same label.
- An alternative is to **generate a separate channel of data** that associates spatial properties with pixels.
  - For example, a texture channel could be added, and *classification carried out* (using a suitable algorithm such as the minimum distance rule) *on the combined multispectral and texture* channels.
    - Along this line, Gong and Howarth (1990) have set up a *structural information* channel to bias a classification according to the density of high spatial frequency data in order to improve the classification of image data containing urban segments.
    - The reasoning behind the approach is that urban regions are characterised by high spatial frequency detail whereas, conversely, the high frequency detail present in non-urban regions is low.
  - The additional channel reflects the underlying structural hypothesis and accordingly influences the classification which would otherwise be carried out on the basis of spectral data alone.

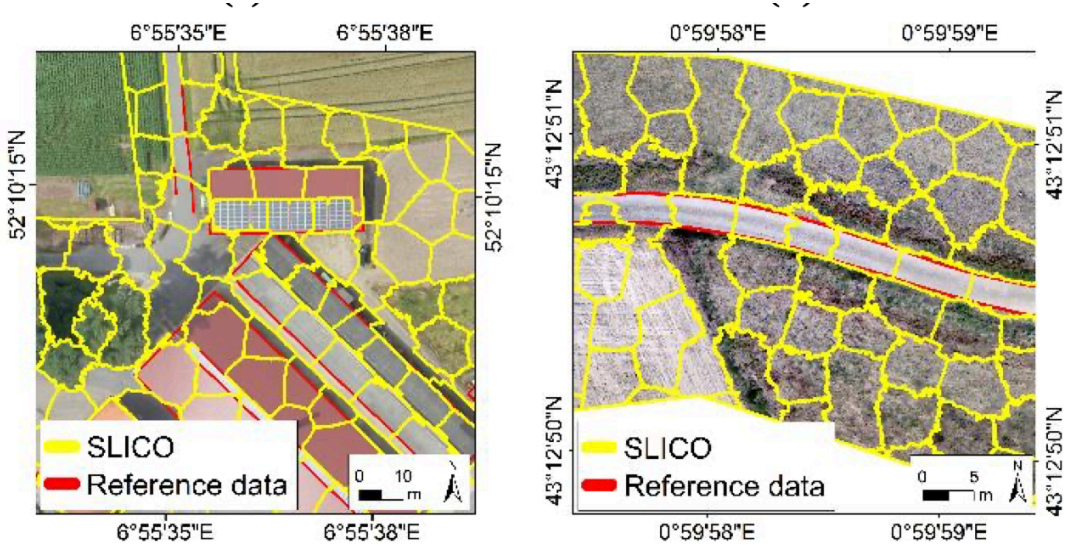
Context Classification by Image Pre-processing

- One of the more useful spatial pre-processing techniques is that used in the **ECHO** classification methodology.
  - In ECHO (Extraction and Classification of Homogeneous *Objects*) regions of similar spectral properties are "grown" before classification is performed.
  - Several **region growing techniques** are available, the simplest of which is to aggregate pixels into small regions by comparing their brightnesses in each channel and then aggregate the small regions into bigger regions in a similar manner.
  - When this is done, ECHO classifies the regions as single *objects* and only resorts to standard maximum likelihood classification when it has to treat individual pixels that could not be put into regions.
  - Here ***objects*** are homogeneous regions according to some criteria (used to guide region geowing) and should not be confused with higher-level semantics-based analysis.

    - Details of ECHO will be found in Kettig and Landgrebe (1976); it is also available in the Multispec image analysis software (https://engineering.purdue.edu/~biehl/MultiSpec/)

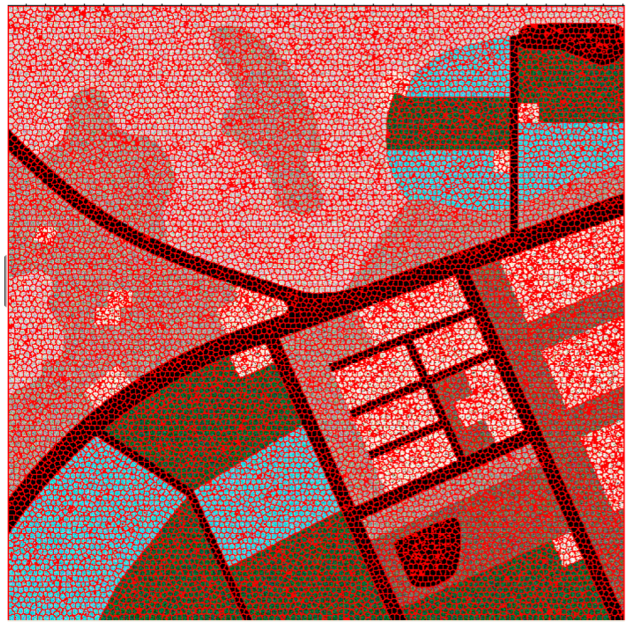
Context Classification by Image Pre-processing

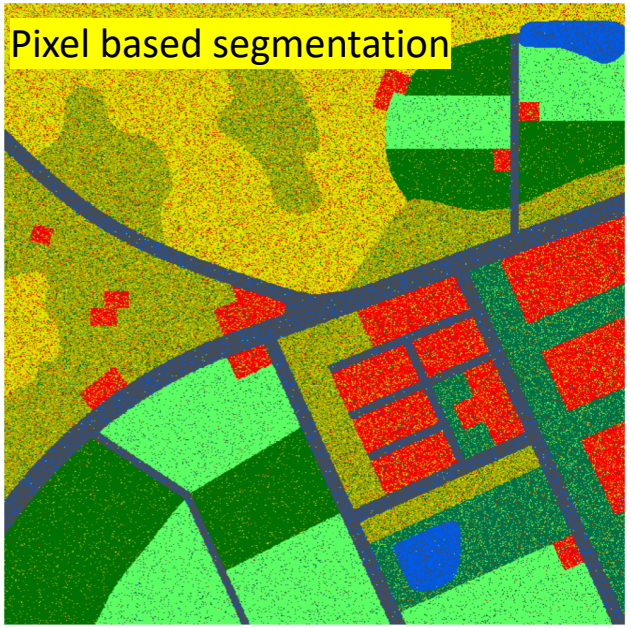
- A more general approach is to **oversegment** the image into what are known as **superpixels** and then classify the superpixels individually.
  - While initially used as a structured (contextual) prediction method, more recently (over)segmentation-based methods have been used for reducing the complexity of image labeling problems. Contextual methods can then be used to exploit dependencies in neighbouring superpixel labels.
  - Labeling a few hundred superpixels per image can be a lot less computationally demanding than labeling thousands of pixels, allowing for more complex models to be used.
  - The main drawback of segmentation-based approaches to context classification is that they are usually unable to recover from incorrect segmentations, which can be particularly problematic in the aerial image setting where occlusion of objects by trees or buildings is common.

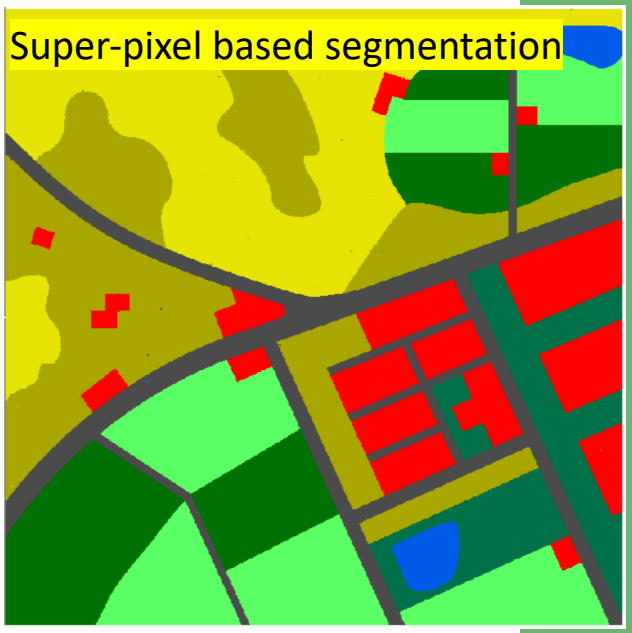


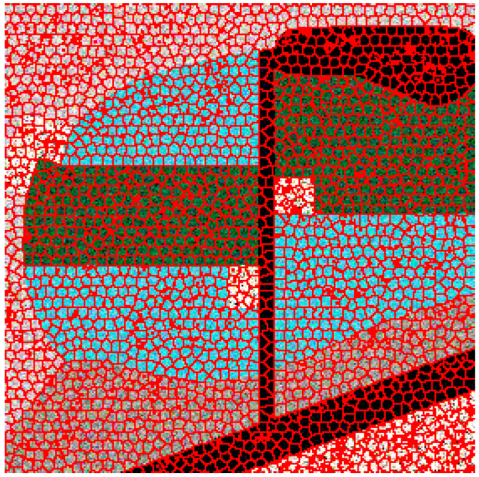
Context Classification by Image Pre-processing

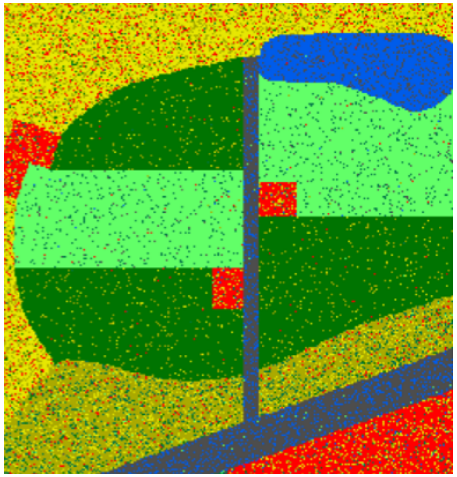
https://doi.org/10.3390/rs8080619

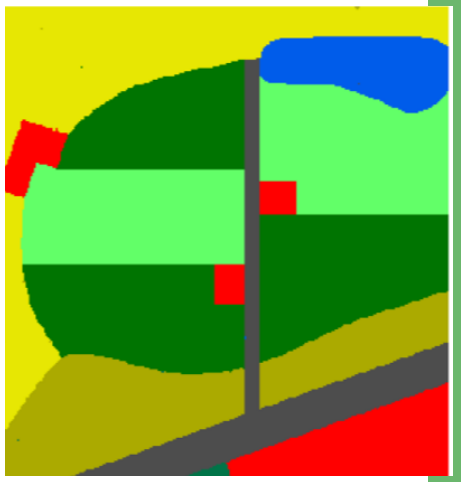








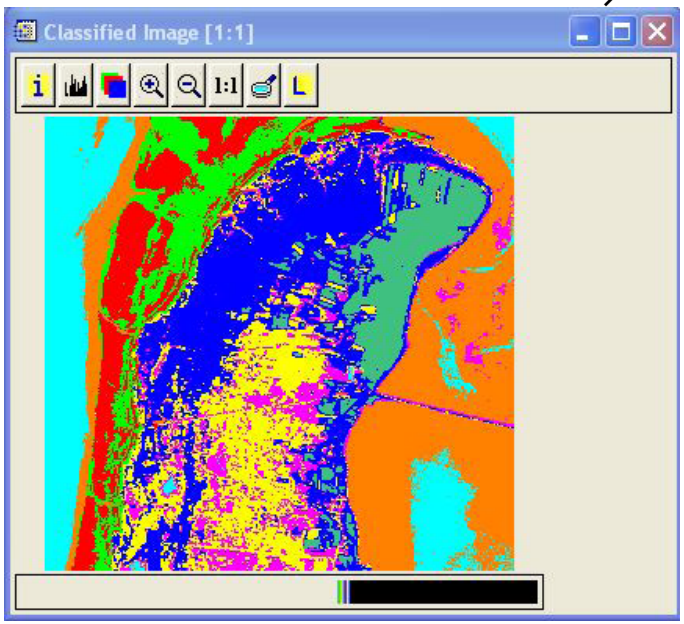


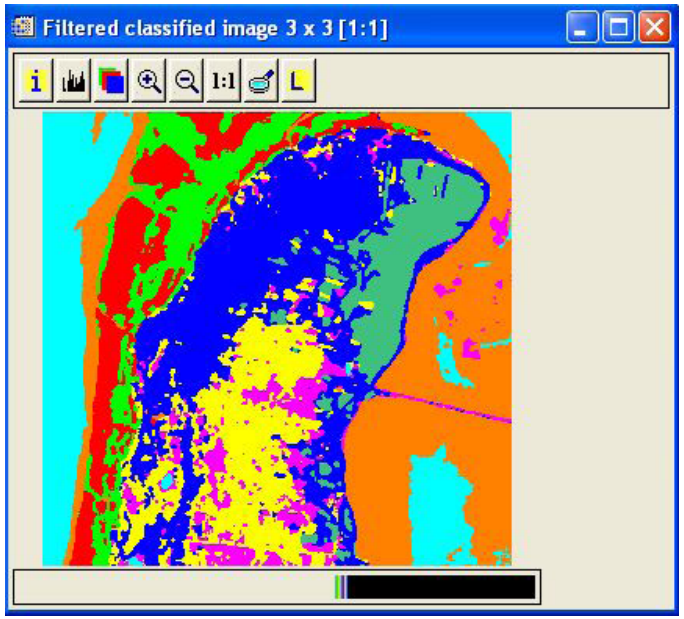


Crop1 | Crop2 | Road | Bush | Forest | Urban | Water | Grass

Post Classification Filtering

- Once a thematic map has been generated using a simple point classifier some degree of spatial context can be developed by logically filtering the map.
  - For example, if the map is examined in 3 × 3 windows, a label at the centre of the window might be changed to the label most represented in the window.
  - Clearly *this must be done carefully*, with the user having some control over the *minimum size region* of a given cover type that is acceptable in the filtered image product (Harris, 1985).
  - Post classification filtering by this approach has been treated by Townsend (1986)
  - Example taken from http://leoworks.terrasigna.com





Probabilistic Label Relaxation

- **Label relaxation** process:
  - it has *little theoretical foundation* (despite its formalized appearance),
  - and is *more complex* than the methods outlined before,
  - However, it does allow the spatial properties of a region to be carried **within** the classification process in a ***logically consistent way***

- The process commences by assuming that a parametric (statistical) classification, based on spectral data alone, has already been carried out.

- There is available therefore, for each pixel, a set of probabilities $p_m(\omega_i)$, that describe the *chance that the pixel* m *belongs to each of the possible ground cover classes* ω_i under consideration:
  - i.e. **posterior probabilities** $p(\omega_i|\boldsymbol{m})$ in the case of max likelihood classification
  - or some other assignment in the case of a different classifier

Probabilistic Label Relaxation

The basic algorithm

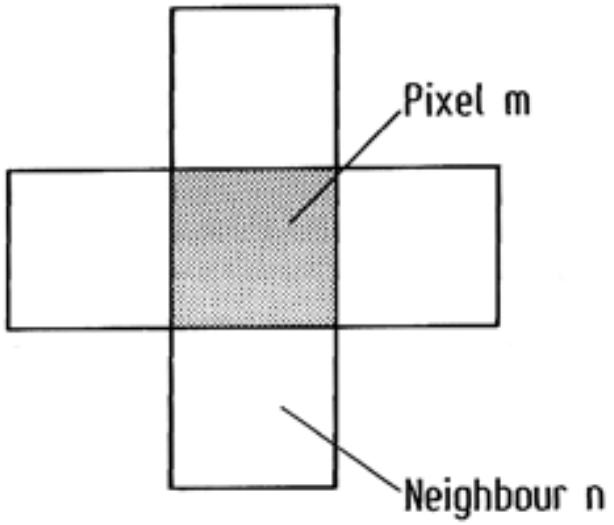
- Let the set of probabilities for a pixel (m) currently of interest be represented by

$$p_m(\omega_i) \qquad i = 1, \ldots K \tag{8.14}$$

where K is the total number of classes; $p_m(\omega_i)$ should be read as "the probability that ω_i is the correct class for pixel m." Note that the full set of $p_m(\omega_i)$ must sum to unity for a given pixel – viz.

$$\sum_i p_m(\omega_i) = 1.$$

- Suppose now that a neighbourhood is defined surrounding pixel m. This can be of any size and, in principle, should be large enough to ensure that all the pixels considered to have any spatial correlation with m are included.
- For high resolution imagery this is not practical and simple neighbourhoods such as that shown in Figure are often adopted.



Probabilistic Label Relaxation

The basic algorithm

- Now assume that a *neighbourhood function* $Q_m(\omega_i)$ can be found (by means to be described below) which allows the pixels in the prescribed neighbourhood to influence the possible classification of pixel m. This influence is exerted by multiplying the label probabilities in (8.14) by the $Q_m(\omega_i)$. However, so that the new set of label probabilities sum to one, these new values are divided by their sum:

$$p'_m(\omega_i) = \frac{p_m(\omega_i)Q_m(\omega_i)}{\sum_i p_m(\omega_i)Q_m(\omega_i)} \qquad (8.15)$$

Such a modification is made to the set of label probabilities for all pixels by moving over the image from its top left hand to bottom right hand corners. In the following it will be seen that the neighbourhood function $Q_m(\omega_i)$ depends on the label probabilities of the neighbouring pixels, so that if all the pixel probabilities are modified in the manner just described then the neighbours for any given pixel have also been altered. Consequently, (8.15) should be applied again to give newer estimates still of the label probabilities. Indeed, (8.15) is applied as many times as necessary to ensure that the $p'_m(\omega_i)$ have stabilised – i.e. that they do not change with further iteration.

Probabilistic Label Relaxation

The basic algorithm

- It is assumed that the $p'_m(\omega_i)$ then represent the correct set of label probabilities for the pixel, having taken account both of spectral data (in the initial determination of label probabilities) and spatial context (via the neighbourhood functions). Since the process is iterative, (8.15) is usually written as an explicit iteration formula:

$$p_m^{k+1}(\omega_i) = \frac{p_m^k(\omega_i)\,Q_m^k(\omega_i)}{\sum_i p_m^k(\omega_i)\,Q_m^k(\omega_i)} \tag{8.16}$$

where k is the iteration counter. Depending on the size of the image and its spatial complexity, the number of iterations required to stabilise the label probabilities may be quite large. However, most change in the label probabilities occurs in the first few iterations and there is good reason to believe that proceeding beyond say 5 to 10 iterations may not be necessary in most cases

Probabilistic Label Relaxation

The Neighbourhood Function

- Consider just one of the neighbours of pixel m – call it pixel n.

- Suppose there is available a ***measure of compatibility*** of the current labelling of pixel m and its neighbouring pixel n.
  - For example let $\boldsymbol{r_{mn}(\omega_i, \omega_j)}$ describe numerically how compatible it is to have pixel m classified as ω_i and neighbouring pixel n classified as ω_j.
    - It would be expected, for example, that this measure will be high if the adjoining pixels are both labelled wheat in an agricultural region, but low if one of the neighbours was classified as snow.

- There are *several ways* these compatibility coefficients, as they are called, can be defined. An intuitively appealing definition is based on *conditional probabilities*.
  - The compatibility measure $\boldsymbol{p_{mn}(\omega_i \mid \omega_j)}$ is the probability that ω_i is the correct label for pixel m **if** ω_j is the correct label on pixel n.
  - A *small piece of evidence* in favour of ω_i being correct for pixel m is $p_{mn}(\omega_i \mid \omega_j)\, p_n(\omega_j)$ – i.e. the probability that ω_i is correct for pixel m if ω_j is correct for pixel n multiplied by the probability that ω_j is correct for pixel n. This is also the joint probability $p_{mn}(\omega_i , \omega_j)$.
  - Since probabilities for all possible labels on pixel n are available (even though some might be very small) the ***total evidence*** from pixel n in favour of ω_i being the correct class for pixel m will be the sum of the contributions from all pixel n's labelling possibilities, viz.

$$\sum_j p_{mn}(\omega_i \mid \omega_j)\, p_n(\omega_j)$$

Probabilistic Label Relaxation

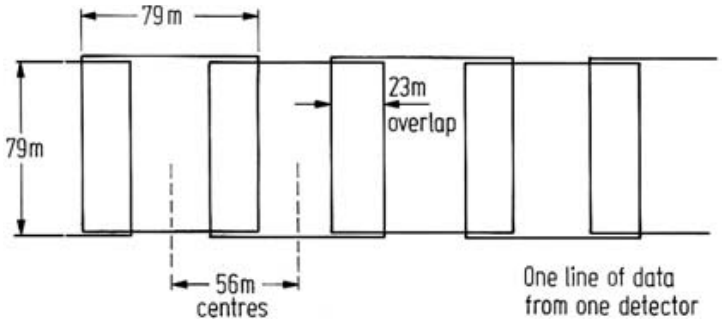
The Neighbourhood Function

- Consider now the full neighbourhood of the pixel m. In a like manner all the neighbours contribute evidence in favour of labelling pixel m as coming from class ω_i. All these contributions are simply added , via the use of *neighbour weights* d_n that recognise that some neighbours may be more influential than others (as for example, pixels along a scan line in MSS data compared with those running down an image, owing to the oversampling that occurs along rows – see Figure). Thus, at the kth iteration, the total neighbourhood support for pixel m being classified as ω_i is:

$$Q_m^k(\omega_i) = \sum_n d_n \sum_j p_{mn}(\omega_i|\omega_j)\, p_n^k(\omega_j) \qquad (8.17)$$

This is the definition of the neighbourhood function. In (8.16) and (8.17) it is common to include pixel m in its own neighbourhood so that the modification process is not entirely dominated by the neigbours, particularly if the number of iterations is so large as to take the process quite a long way from its starting point.

Unless there is good reason to do otherwise the neighbour weights are generally chosen all to be the same.



The relationship between instantaneous field of view and pixel overlap for Landsat MSS pixels

Probabilistic Label Relaxation

Determining the Compatibility Coefficients

- Several methods are possible for determining values for the compatibility coefficients $p_{mn}(\omega_i \mid \omega_j)$
  - One is to have available a spatial model for the region under consideration, derived from some other data source.
    - In an agricultural region, for example, some general idea of *field sizes along with a knowledge of the pixel size* of the sensor being used should make it possible to estimate how often one particular class occurs following a given class on an adjacent pixel.
  - Another approach is to compute values for the compatibility coefficients from ground truth pixels, although the ground truth needs to be in the form of training regions that contain heterogeneous and spatially representative cover types.

The Final Step – Stopping the Process

- While the relaxation process operates on label probabilities, the user is interested in the actual labels themselves.
  - At the completion of relaxation, or at any intervening stage, each of the pixels can be classified according to the highest label probability.
  - Thought has to be given as to how and when the iterations should be terminated

Probabilistic Label Relaxation

The Final Step – Stopping the Process

- As suggested earlier, the process can be allowed to go to a *natural completion* at which further iteration leads to no changes in the label probabilities for all pixels.
- This however presents two *difficulties*.
  - First, *up to several hundred iterations* may be involved leading to a costly post classification step.
  - Secondly, it is observed *in practice that the relaxation process improves the classification results in the first few iterations*, by the embedding of spatial information, *often to deteriorate later in the process* (Richards, Landgrebe and Swain, 1981).
    - Indeed, if the process is not terminated, the thematic map, after a large number of iterations of relaxation, can be worse than before the technique was applied.
- To avoid these difficulties, a stopping rule or other controlling mechanism is needed.
  - As seen in the following example, stopping after just a few iterations may allow most of the benefit to be drawn from the process.
  - Alternatively, the labelling errors remaining at each iteration can be checked against ground truth, if available, and the iterations terminated when the labelling error is seen (remember the little theoretical foundation) to be minimised (Gong and Howarth, 1989).

Probabilistic Label Relaxation

The Final Step – Propagation control

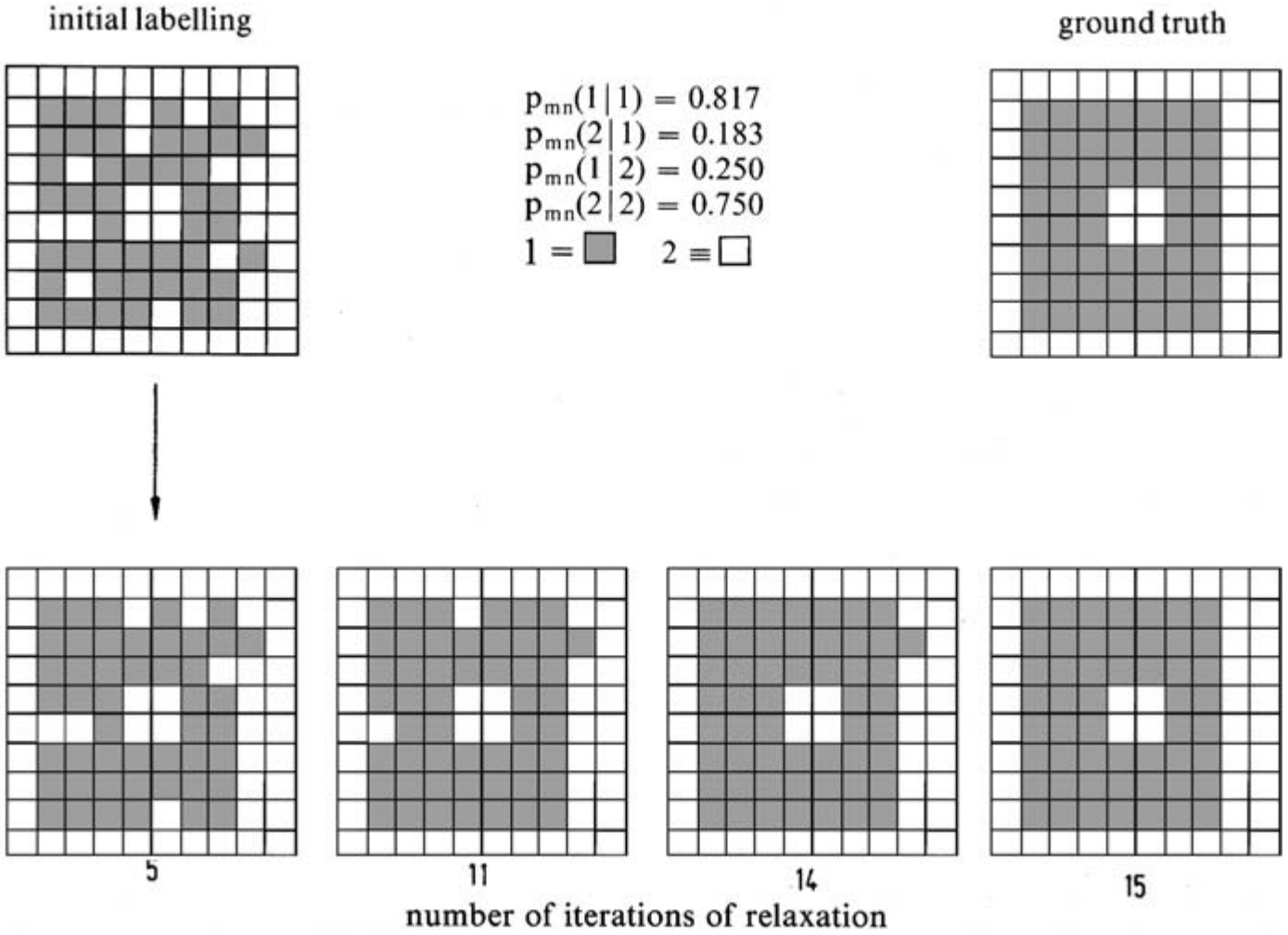
- Another approach is to control the propagation of contextual information as iteration proceeds (Lee, 1984).
  - A little thought will reveal that, in the first iteration, only the immediate neighbours of a pixel have an influence on its labelling.
  - In the second iteration the neighbours two away will now have an influence via the intermediary of the intervening pixels.
  - Similarly, as iterations proceed, information from neighbours further away is propagated into the pixel of interest to modify its label probabilities.

- If the user has a view of the separation between neighbours at which the spatial correlation has dropped to negligible levels, then the *appropriate number of iterations should be able to be identified at which to terminate the process* without unduly sacrificing any further improvement in labelling accuracy.
  - Noting also that the nearest neighbours should be most influential, with those further out being less important, a useful variation is to reduce the values of the neighbour weights d_n as iteration proceeds so that after say 5 to 10 iterations they have been brought to zero.
  - Further iterations will then have no effect, and degradation in labelling accuracy cannot occur (Lee and Richards, 1989).

Probabilistic Label Relaxation

Example 1

- Figure illustrates a simple application of relaxation labelling, in which a hypothetical image of 100 pixels has been classified into just two classes – grey and white.



Probabilistic Label Relaxation

Example 1 (contd)

- The ground truth for the region is shown, along with the *thematic map* (initial labelling) assumed to have been generated from a point classifier such as the maximum likelihood rule.
- Also shown are the *compatibility coefficients*, expressed as conditional probabilities, computed from the ground truth map.
  - Label probabilities were assumed to be 0.9 for the favoured label in the initial labelling and 0.1 for the less likely label.
  - The initial labelling, by comparison with the ground truth, can be seen to have an accuracy of 82% (there are 12 pixels in error).
- The labelling (selected on the basis of the largest current label probability) at significant *stages during iteration* is shown, illustrating the *reduction in classification error* resulting from the incorporation of spatial information into the process.
  - After 15 iterations all initial labelling errors have been removed, leading to a thematic map 100% in agreement with the ground truth.
- In this case, the relaxation process was allowed to proceed to completion and there have been *no ill effects*.
- As pointed out previously, however, this is the exception and stopping rules may have to be applied in most cases.

Probabilistic Label Relaxation

Example 2

- As a second example, 82×100 pixels of an agricultural image have been chosen (in Figure they are represented in false color).

Probabilistic Label Relaxation

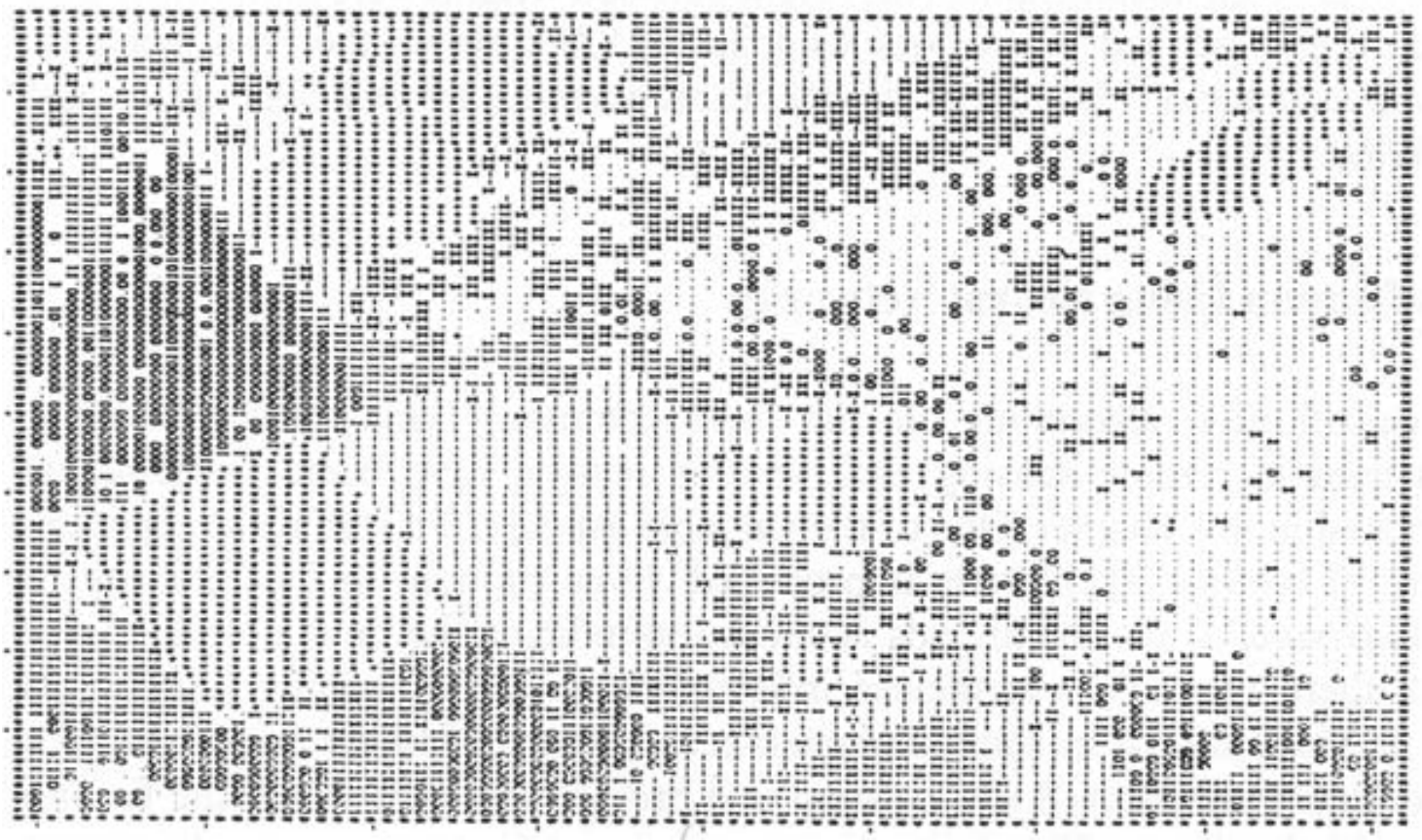
Example 2 (contd)

- This Figure shows the classification ground truth

Probabilistic Label Relaxation

Example 2 (contd)

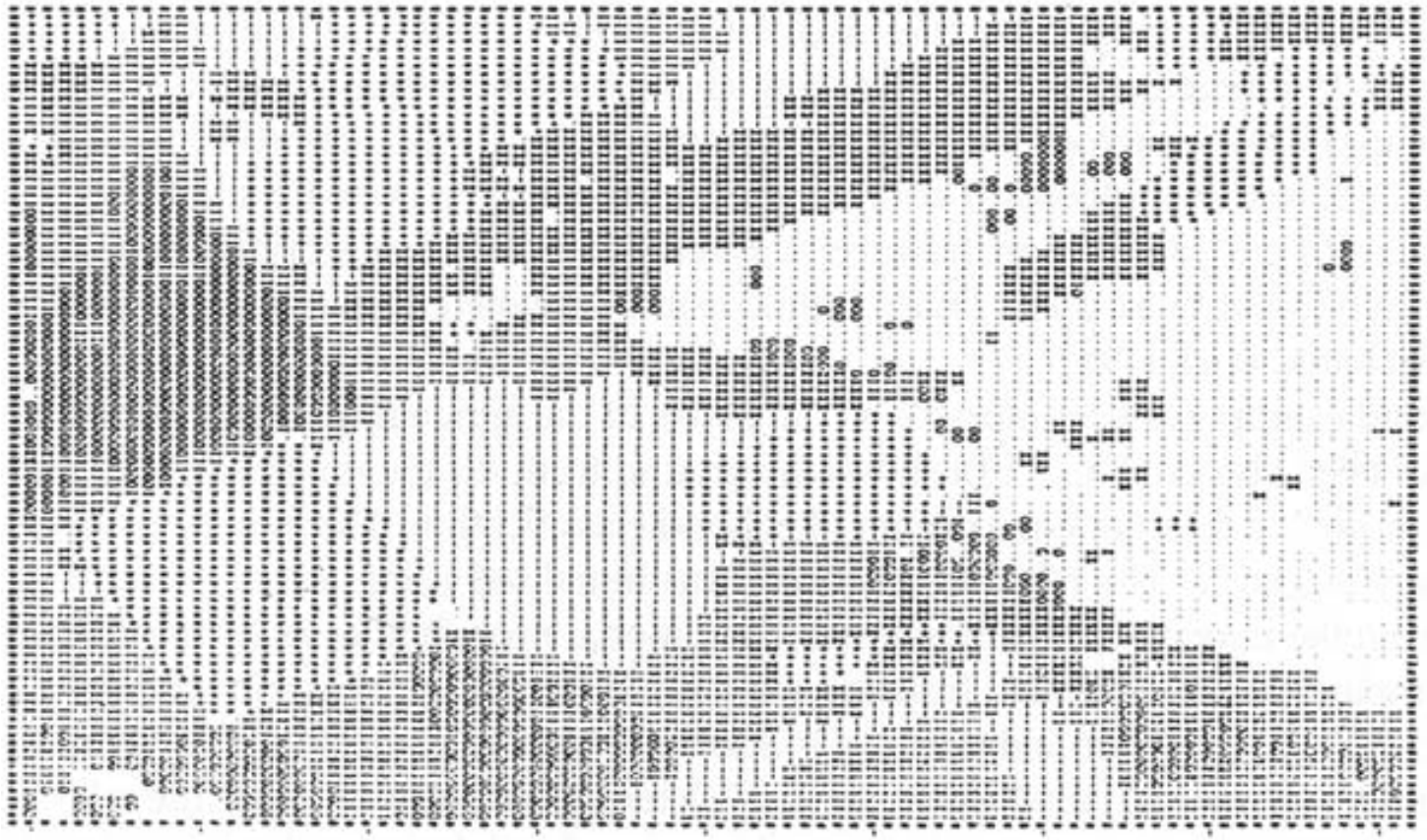
- This Figure shows the result of a maximum likelihood classification.
  - The initial classification accuracy is 65.6%. The relaxation process was initialised using actual probability estimates from the maximum likelihood rule.



Probabilistic Label Relaxation

Example 2 (contd)

- This Figure shows the final labelling, which has an accuracy of 72.2%.
  - to control the propagation of context information and thereby obviate any deleterious effect of allowing the relaxation process to proceed unconstrained, the neighbourhood weights were diminished with iteration count (as specified next)



Probabilistic Label Relaxation

Example 2 (contd)

- In this example the neighbourhood weights were diminished with iteration count according to

$$d_n(k) = d_n(1)e^{-\alpha(k-1)}$$

in which α controls how the neighbour weights change with iteration.

  - If α = 0 there is no reduction and normal relaxation applies. For α large the weights drop quickly with iteration. The central pixel was not included in the neighbourhood definition.

- **Parameter tuning** The Table shows how the relaxation performance depends on α . Irrespective of the value chosen the optimal result is achieved after about 4 iterations, giving an accuracy of 72.2%. The table also shows the result achieved if relaxation is left to run for more iterations (final result).

	Optimal result		Final result	
α	Accuracy	At iteration	Accuracy	At iteration
0.0	72.2	4	70.6	32
1.0	72.2	4	71.4	17
1.8	72.2	4	72.1	10
2.0	72.2	4	72.2	9
2.2	72.2	4	72.2	8
2.5	72.2	5	72.2	7
3.0	72.2	4	72.2	6

  - Without diminishing the neighbour weights or without diminishing them sufficiently, the final result is worse than the initial classification error. However, for values of a in the vicinity of 2 the result is fixed at 72.2% from iteration 4 onwards.

Handling Spatial Context by Markov Random Fields

The effect of spatial context can also be incorporated into a classification using the concept of the Markov Random Field (MRF). It is useful in developing the Markov Random Field approach to commence by considering the whole image, rather than just a local neighbourhood. We will restrict our attention to a neighbourhood once we have established some fundamental concepts.

Suppose there is a total of M pixels in the image to be classified, with measurement vectors $\boldsymbol{x}_1, \ldots \boldsymbol{x}_M$. Alternatively, the measurement vectors can be expressed $\{\boldsymbol{x}_m : m = 1, \ldots M\}$, in which $m \equiv (i, j)$ in our usual way of indexing the pixels in an image. We can describe the full set of measurement vectors by $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots \boldsymbol{x}_M\}$. Further, suppose the class labels on each of the M pixels can be represented by the set $\Omega = \{\omega_{c1}, \ldots \omega_{cM}\}$; we could refer to that as the *scene labelling*, because it looks at the classification of every pixel in the scene. Each ω_{cm} can be one of $c = 1, \ldots C$ available classes. By classification what we want to find, of course, is the scene labelling (or our best estimate) that matches the ground truth – i.e. the actual classes of the pixels on the earth's surface. Let the actual labels on the ground be represented by Ω^*.

There will be a probability distribution $p(\Omega)$ associated with the labelling Ω of the whole scene which describes the likelihood of finding that distribution of labels over the image. Ω is sometimes referred to as a *random field*.

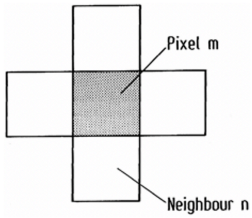
Handling Spatial Context by Markov Random Fields

In principle, what we would like to do is find the scene labelling $\hat{\Omega}$ – that is the classification of all pixels – that maximises the global posterior probability $p(\Omega|\boldsymbol{X})$, the probability that Ω is the correct overall scene labelling given that the full set of measurement vectors for the scene is $\boldsymbol{X}$. By using Bayes' theorem we can express this as

$$\hat{\Omega} = \arg\max_{\Omega}\{p(\boldsymbol{X}|\Omega)p(\Omega)\} \tag{8.18}$$

in which the argmax function says that we choose the value of Ω that maximises its argument. The distribution $p(\Omega)$ is the prior probability of the scene labelling.

What we need to do now essentially is to perform the maximisation in (8.18), recognising however that the pixels are contextually dependent ie. there is some spatial correlation among them because adjacent pixels are likely to come from the same class. To render the problem tractable we consider the posterior probability just at the individual pixel level, so that our objective, for pixel m, is to find the class c that maximises $p(\omega_{cm}|\boldsymbol{x}_m, \omega_{\partial m})$ where $\omega_{\partial m}$ is the labelling on the pixels in a neighbourhood about pixel m. A possible neighbourhood is that shown in Fig. although often the immediately diagonal neighbours about m can also be included.



Handling Spatial Context by Markov Random Fields

Now we note

$$\begin{aligned} p(\omega_{cm}|\boldsymbol{x}_m, \omega_{\partial m}) &= p(\boldsymbol{x}_m, \omega_{\partial m}, \omega_{cm})/p(\boldsymbol{x}_m, \omega_{\partial m}) \\ &= p(\boldsymbol{x}_m|\omega_{\partial m}, \omega_{cm})\, p(\omega_{\partial m}, \omega_{cm})/p(\boldsymbol{x}_m, \omega_{\partial m}) \\ &= p(\boldsymbol{x}_m|\omega_{\partial m}, \omega_{cm})\, p(\omega_{cm}|\omega_{\partial m})\, p(\omega_{\partial m})/p(\boldsymbol{x}_m, \omega_{\partial m}) \end{aligned}$$

The first term on the right hand side is similar to the class conditional distribution function, but conditional also on the neighbourhood labelling. It is reasonable to assume that the class conditional density is independent of the neighbourhood labelling so that $p(\boldsymbol{x}_m|\omega_{\partial m}, \omega_{cm}) = p(\boldsymbol{x}_m|\omega_{cm})$. Note also that the measurement vector $\boldsymbol{x}_m$ and the neighbourhood labelling are independent of each other so that $p(\boldsymbol{x}_m, \omega_{\partial m}) = p(\boldsymbol{x}_m)\, p(\omega_{\partial m})$, so that the last expression becomes

$$\begin{aligned} p(\omega_{cm}|\boldsymbol{x}_m, \omega_{\partial m}) &= p(\boldsymbol{x}_m|\omega_{cm})\, p(\omega_{cm}|\omega_{\partial m})\, p(\omega_{\partial m})/p(\boldsymbol{x}_m)\, p(\omega_{\partial m}) \\ &= p(\boldsymbol{x}_m|\omega_{cm})\, p(\omega_{cm}|\omega_{\partial m})/p(\boldsymbol{x}_m) \end{aligned}$$

Since $1/p(\boldsymbol{x}_m)$ does not contribute to the decision concerning the correct label for pixel m it can be removed from the last expression, leaving

$$p(\omega_{cm}|\boldsymbol{x}_m, \omega_{\partial m}) \propto p(\boldsymbol{x}_m|\omega_{cm})\, p(\omega_{cm}|\omega_{\partial m}) \tag{8.19}$$

Handling Spatial Context by Markov Random Fields

Now consider the probability $p(\omega_{cm}|\omega_{\partial m})$. Essentially it is the probability that the correct class for pixel m is c given the classes currently on the neighbours of pixel m. In many ways it is analogous to the neighbourhood function for probabilistic relaxation in (8.17). It is also a conditional prior probability – i.e. a prior probability for the class on pixel m conditional on its neighbourhood. Because of this conditionality, the random fields of labels we are considering are now referred to as *Markov* Random Fields (*MRF*).

The question is how do we now find a value for $p(\omega_{cm}|\omega_{\partial m})$? It is a property of MRFs that we can express the conditional prior distribution in the form of a Gibbs distribution

$$p(\omega_{cm}|\omega_{\partial m}) = \frac{1}{Z}\exp\{-U(\omega_{cm})\} \tag{8.20a}$$

in which (based on the so-called Ising model)

$$U(\omega_{cm}) = \sum_{\partial m} \beta[1 - \delta(\omega_{cm}, \omega_{\partial m})] \tag{8.20b}$$

where $\delta(\omega_{cm}, \omega_{\partial m})$ is the Kroneker delta, which is unity if the arguments are equal and zero otherwise; $\beta > 0$ is a parameter with value fixed by the user when applying the MRF technique to control the influence of the neighbours.

Handling Spatial Context by Markov Random Fields

Equation (8.20) is now substituted into (8.19) to generate a posterior probability that depends on the class conditional probability found from the available spectral measurements (the first term on the right hand side) and the effect of the spatial neighbourhood. However, as with (8.4), it is convenient to take the logarithm of (8.19) to yield (with the choice of $Z = 1$), an MRF-based discriminant function for the class on pixel m assuming a multivariate normal class conditional density function:

$$g_{cm}(\boldsymbol{x}_m) = -\frac{1}{2}\ln|\Sigma_c| - \frac{1}{2}(\boldsymbol{x}_m - \boldsymbol{m}_c)\Sigma_c^{-1}(\boldsymbol{x}_m - \boldsymbol{m}_c)^t - \sum_{\partial m}\beta[1 - \delta(\omega_{cm}, \omega_{\partial m})] \,.$$

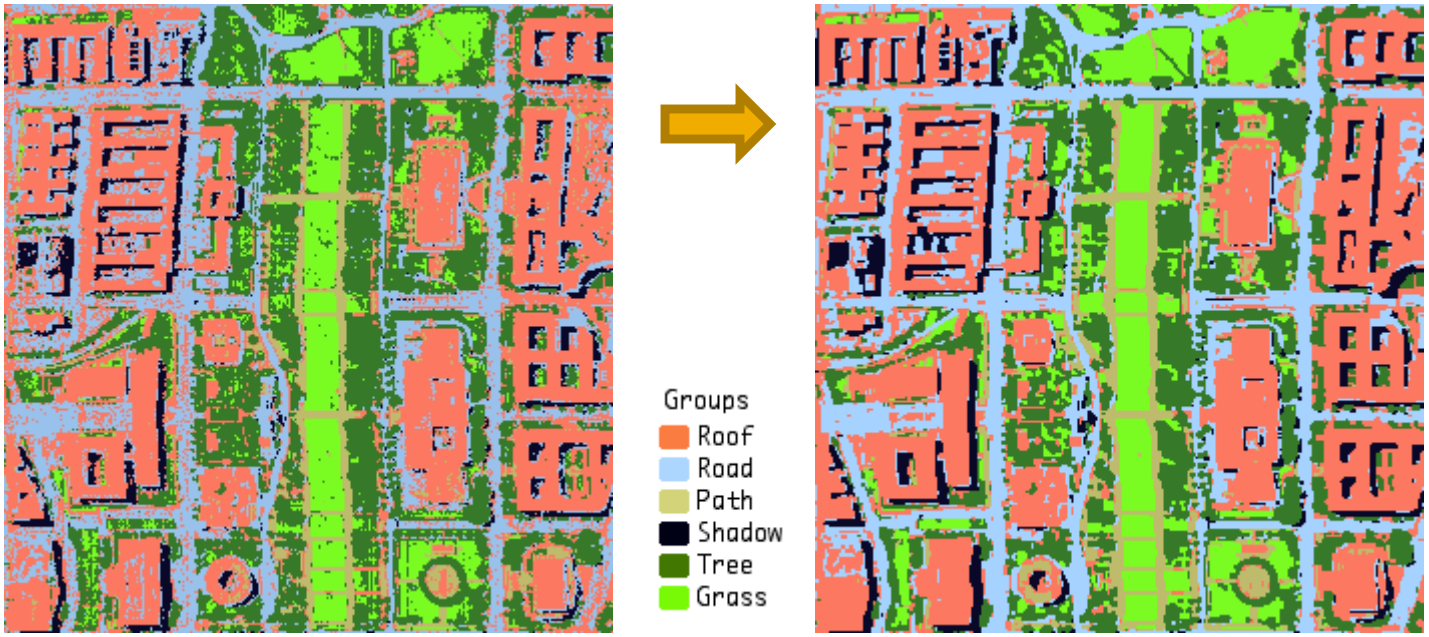
Recall that classification is carried out on the basis of finding the class for the pixel that maximises the discriminant function. Noting the negative signs above, the most appropriate class for pixel m can be found by minimising the expression

$$d_{cm}(\boldsymbol{x}_m) = \frac{1}{2}\ln|\Sigma_c| + \frac{1}{2}(\boldsymbol{x}_m - \boldsymbol{m}_c)\Sigma_c^{-1}(\boldsymbol{x}_m - \boldsymbol{m}_c)^t + \sum_{\partial m}\beta[1 - \delta(\omega_{cm}, \omega_{\partial m})] \tag{8.21}$$

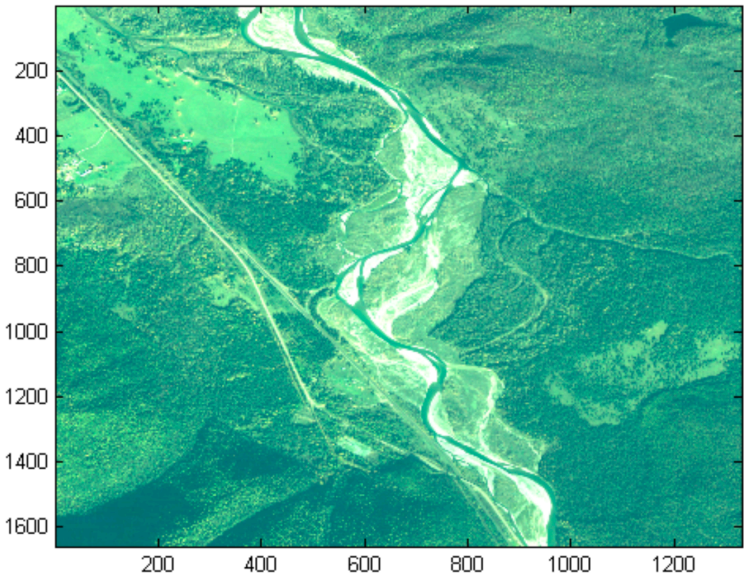
Handling Spatial Context by Markov Random Fields

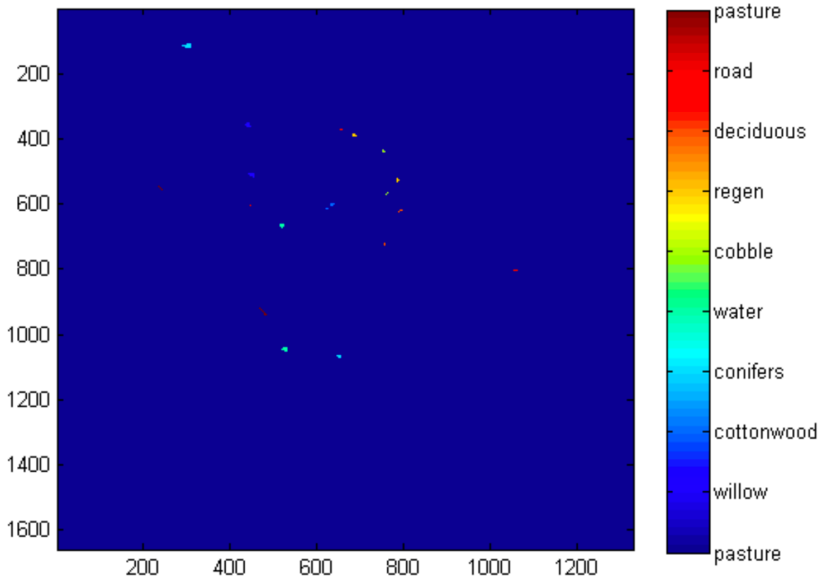
To use (8.21) there needs to be an allocation of classes over the scene before the last term can be computed. Accordingly, an initial classification would be performed, say with the maximum likelihood classifier of Sect. 8.2.3. Equation (8.21) would then be used to modify the labels attached to the individual pixels to incorporate the effect of context. However, in so doing some (or initially many) of the labels on the pixels will be modified. The process should then be run again, and indeed as many times presumably until there are no further changes.

Example from: Q. Jackson and D. Landgrebe, Adaptive Bayesian Contextual Classification Based on Markov Random Fields, IEEE Tran. Geoscience and Remote Sensing (2002)

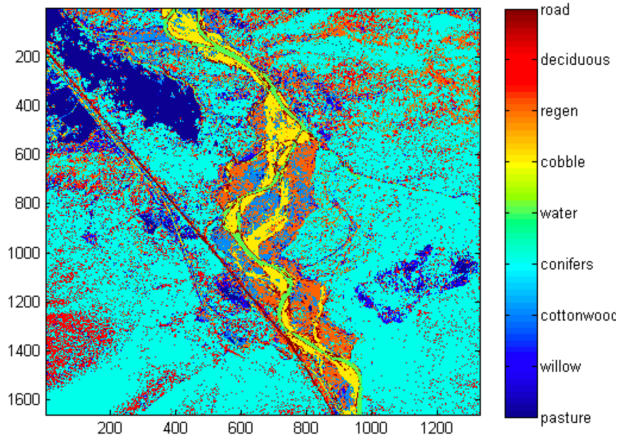


Comparative example

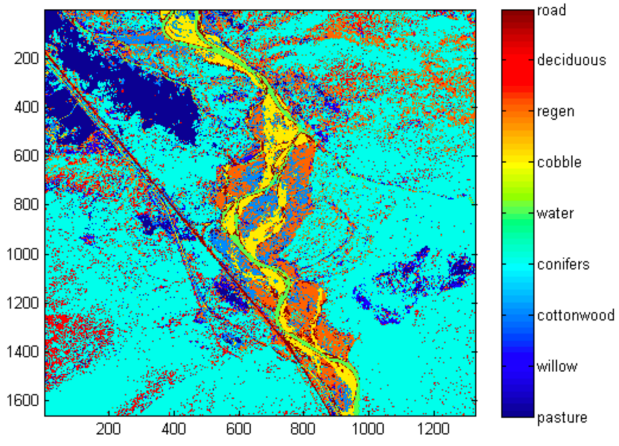




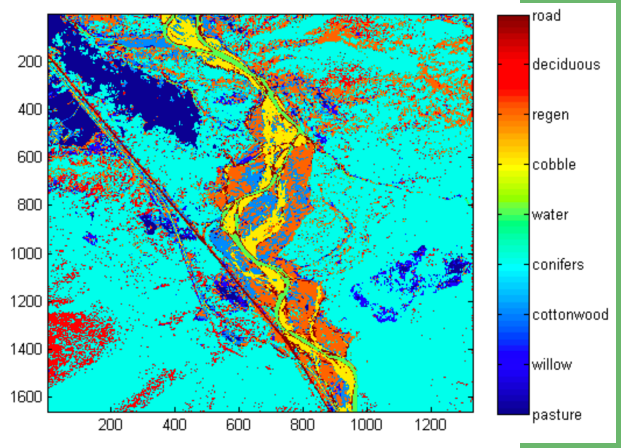
Max likelihood classification



Probability label relaxation

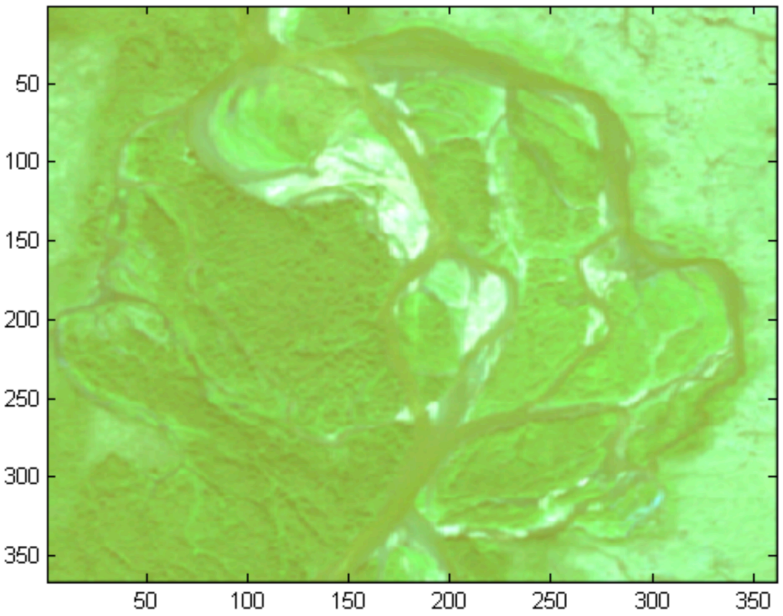


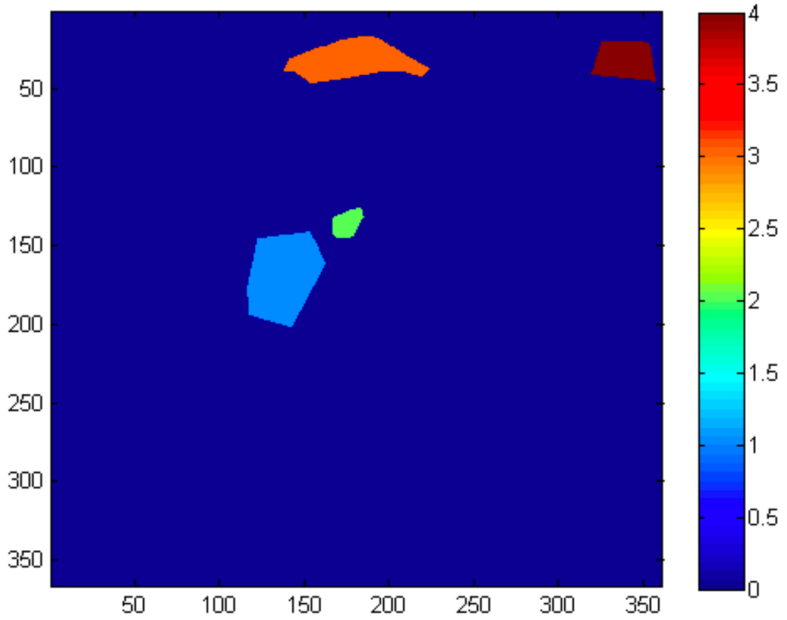
Markov random fields



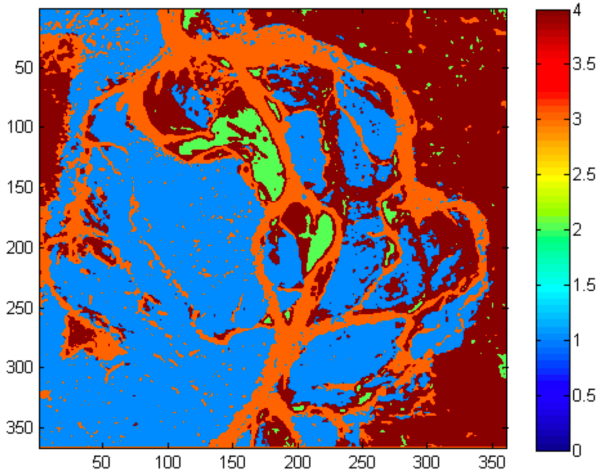
http://hs.umt.edu/math/research/technical-reports/documents/2010/05_Supervised_ClassificationJSS.pdf

Comparative example

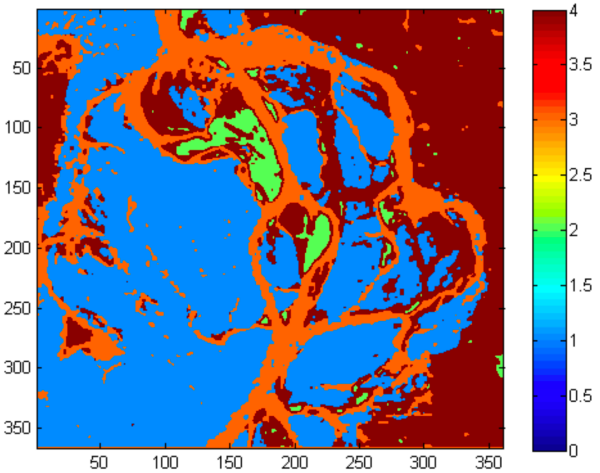




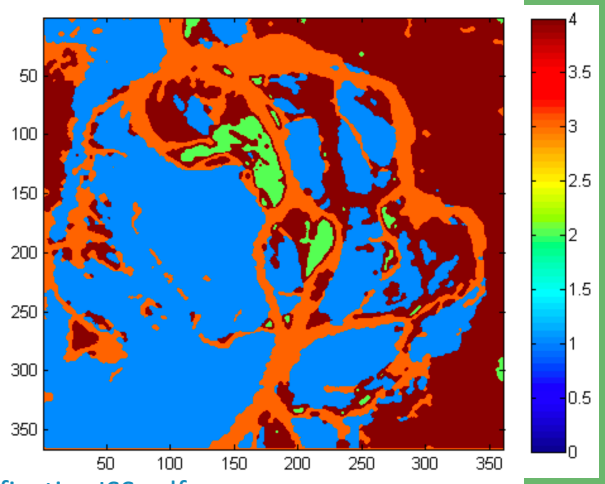
Max likelihood classification



Probability label relaxation



Markov random fields



http://hs.umt.edu/math/research/technical-reports/documents/2010/05_SupervisedClassificationJSS.pdf