Machine Learning Lesson 2 Supervised learning

SILVERIO GARCÍA CORTÉS

ASSOCIATE PROF. UNIVERSITY OF OVIEDO

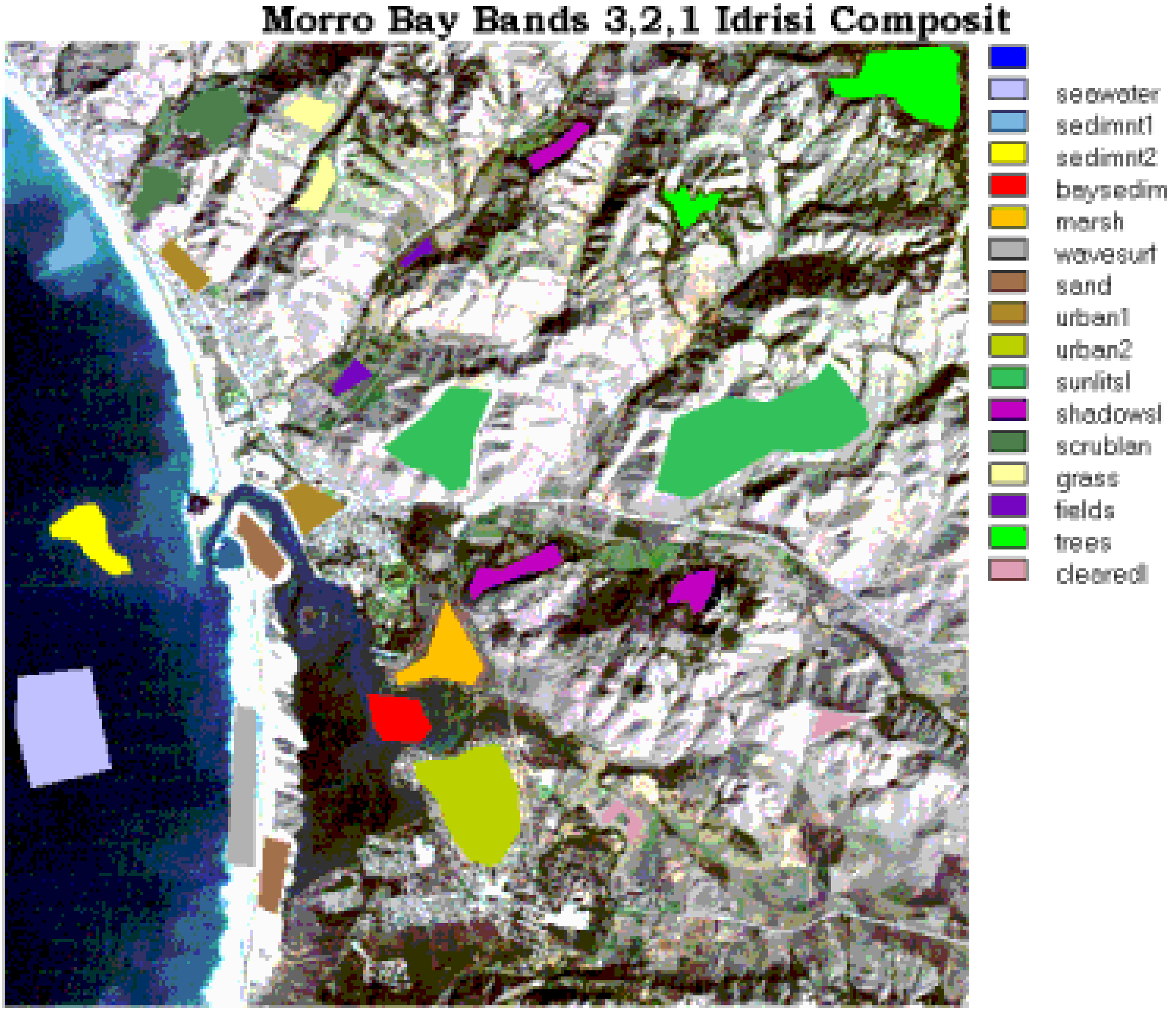
MAY 23, NAPLES

PREPARED FOR UNIV. FEDERICO II

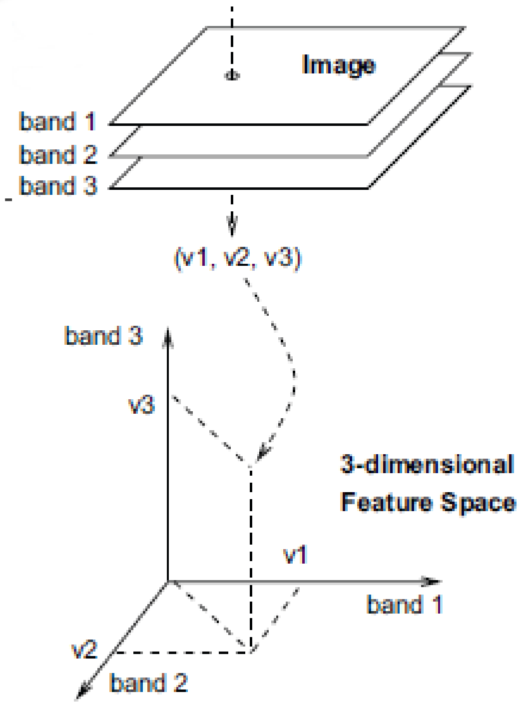
Classification: supervised learning. Training areas

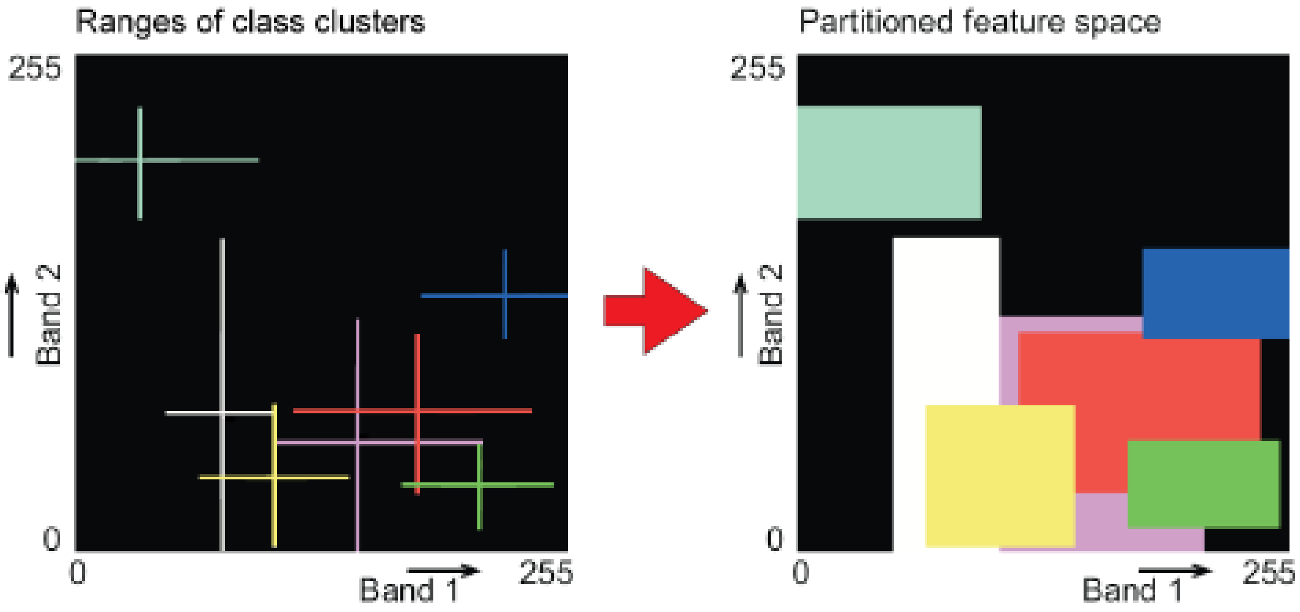
Supervised Classification

- The user defines training pixel areas (polygons) which for sure belong to the desired classes (e. g vegetation, water, …)
- Training regions must be representative of class variability and size :
  - (nº pixels= 30 x Nº bandas)
- There must not exist spatial nor spectral overlap between training áreas from different classes. It is neccesary to check the spectral signatures.
- Thus a previous knowledge of the área is needed before marking training polygons
- Some of the training regions must be preserved for classifier validation.



Some classifiers :box classifier, Parallelepiped classifier



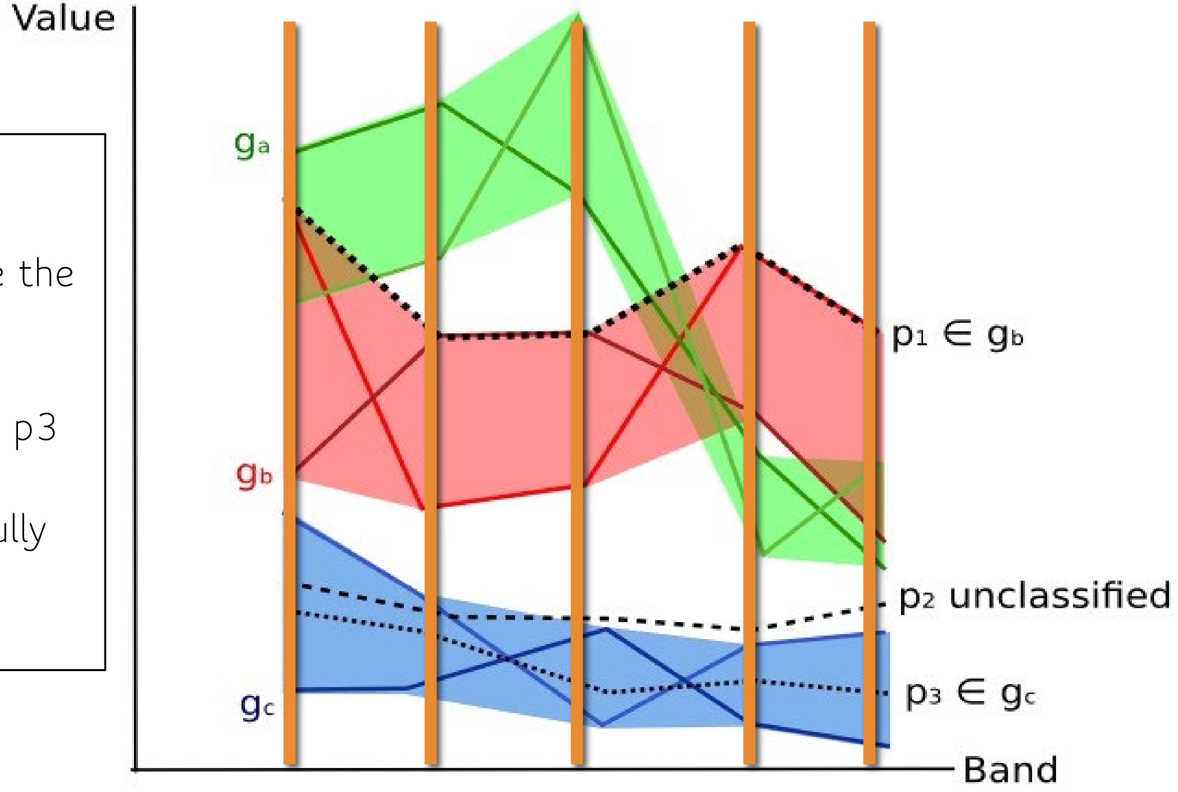


Feature space partitioning — Box classifier

- It is not very much used today. It is very simple and results are not very good..
- Upper and lower limits are defined for all classes in all bands. Reflectance absolute values can be used or mean and standar deviation.
- Some “boxes” or hyperdimensional parallelepiped areas in the feature space appear as result
- During the clasification (prediction phase) each pixel is assigned to the class corresponding to the containing box in the feature space.
- Pixels (hyperspectral vectors) which does not belong to any box are labelled as unknown class (or reject class)

Spectral signature(Landcover signature. SCP)

- Coloured regions ga, gb and gc are the bands bounding the spectral regions to which belong the spectral signatures
- The lines indicate the pixel signatures of the ROIs that define the upper and lower limits of each class
- A pixel belongs to a class if its spectral signature is fully contained in the.spectral signature region of a class (p1*ϵ* gb, p3 *ϵ* gc)
- P2 will be an unclassified pixel because its signature is not fully contained in gc



- Upper and lower limits around the spectral signature are defined. For this, the ROI statistics are used. (user definition is also posible in SCP)
- One pixel is classified as belonging to a specific class if its spectral signature completely belongs to the class área for all bands.
- Classes will be correctly assigned if the spectral regions does not overlap at least in one band. For example red and green classes partially overlap but even then the pixels can be correctly assigned using the condition of full pertenence to a región.

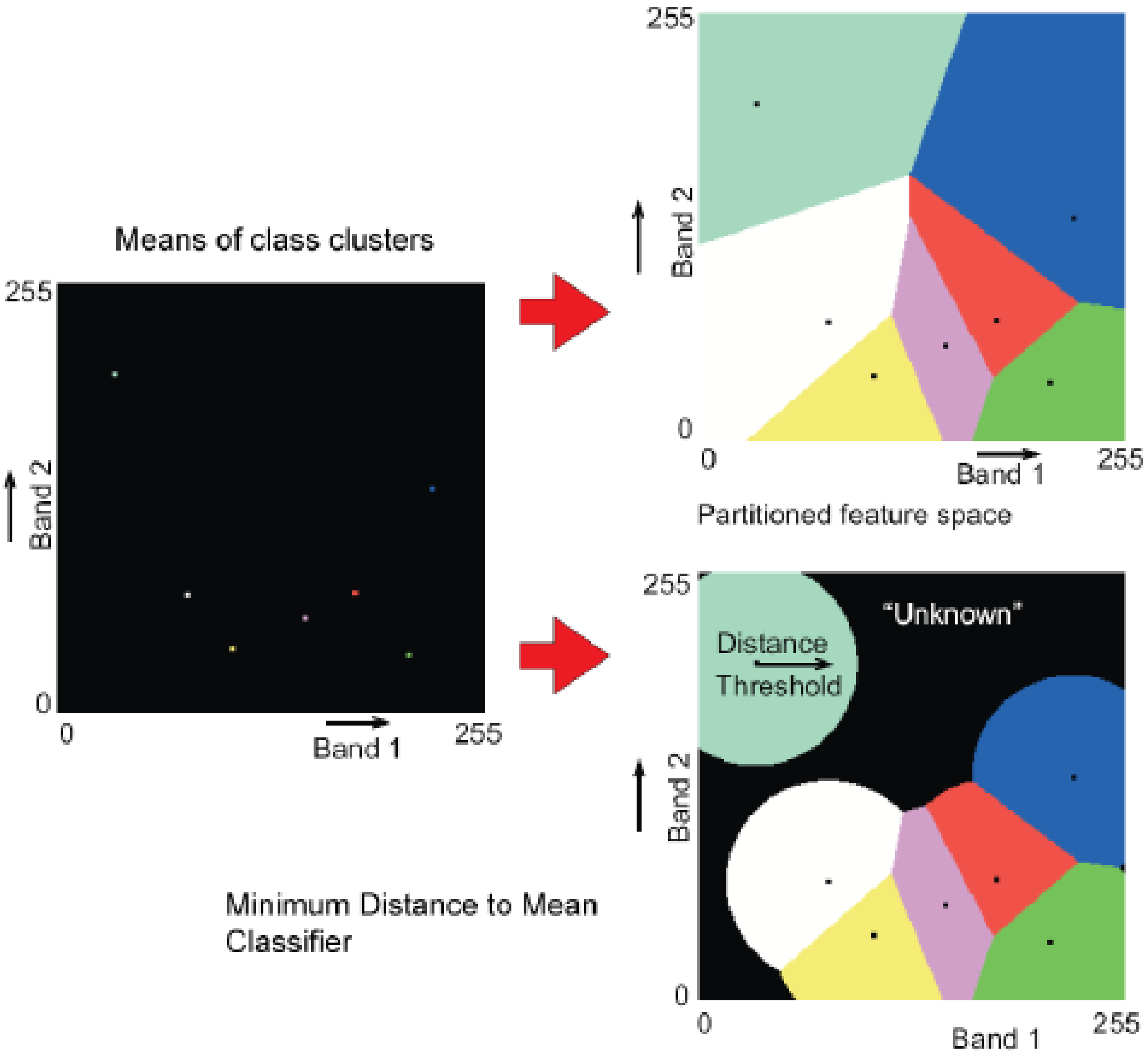
MDM: Minimum Distance to mean classifier

MDM

- Each pixel is classified depending of its distance to the class centers
- Class centers are defined during the training phase

- Problems:
  - Very far pixels can be assigned to a class
  - Dispersion of classes is not taking into account

- $L_2(\vec{x}_1,\vec{x}_2) = [\sum_i |x_{1i} - x_{ci}|^2]^{1/2}$



Minimum Distance to Mean Classifier

CLASIFICADOR DE MÁXIMA PROBABILIDAD
MÁXIMUM LIKELIHOOD CLASSIFIER (ML)

Bayes decision Function

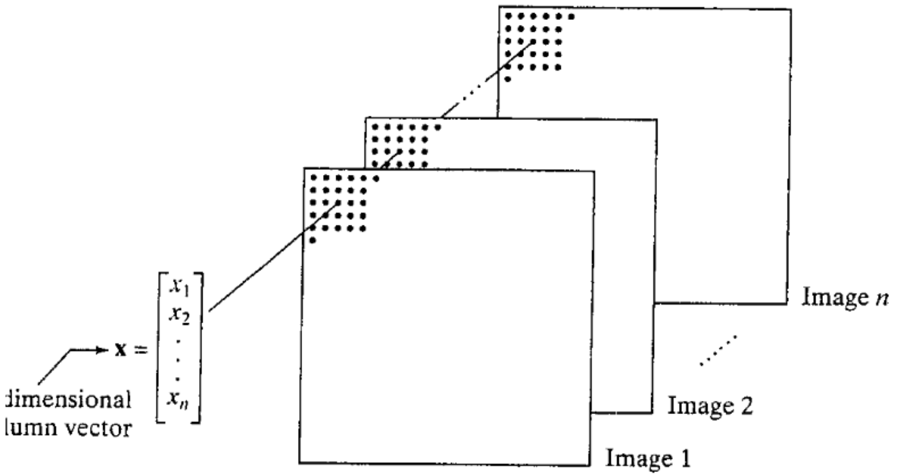
$$d_j(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j) \quad j = 1, 2, \ldots, W$$

Gaussian Probability

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}[(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)]}$$

Final decision Function

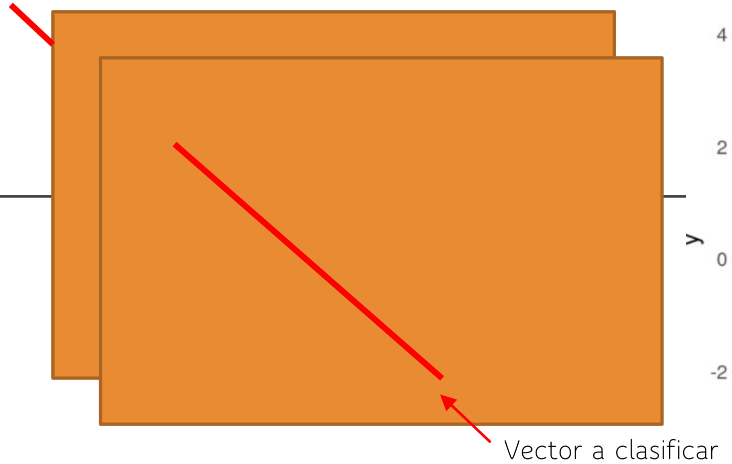
$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}[(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)]$$



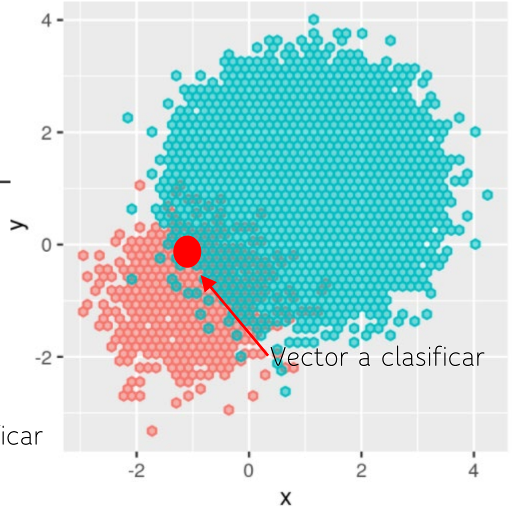
- It takes into account not only the centres of the classes but also the shape. The size and orientation of the clusters by calculating the statistical distance from the means and the covariance matrix. This statistical distance is a probability

- A pixel is assigned to a class if the probability of belonging to that class is the maximum among the probabilities of belonging to all possible classes.
- Gaussian distribution is assumed in each class.
- It is necessary to have a sufficient number of pixels in the training regions to allow the calculation of the covariance nuance.

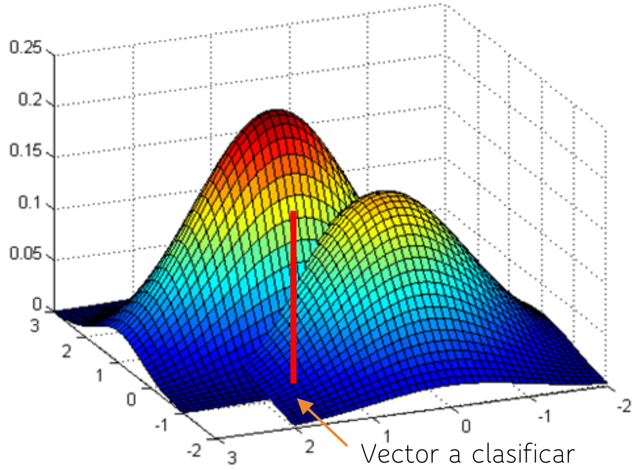
2 band scene

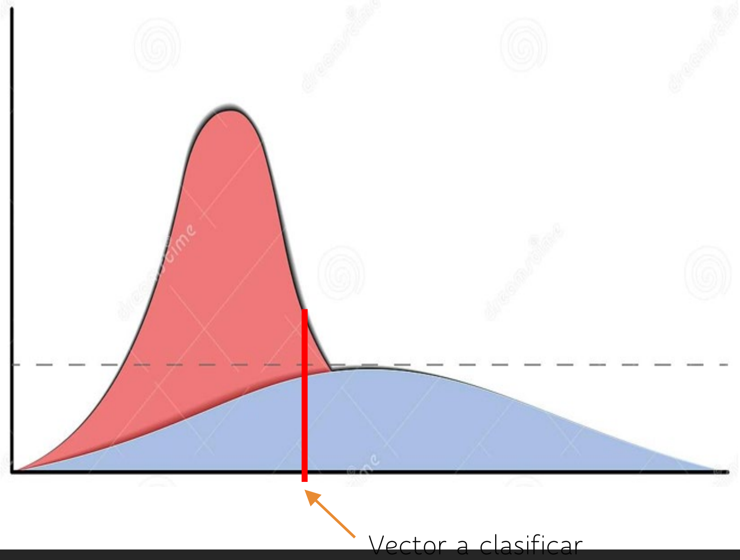


Feature space
2 classes



Probability Density
2 band Vector





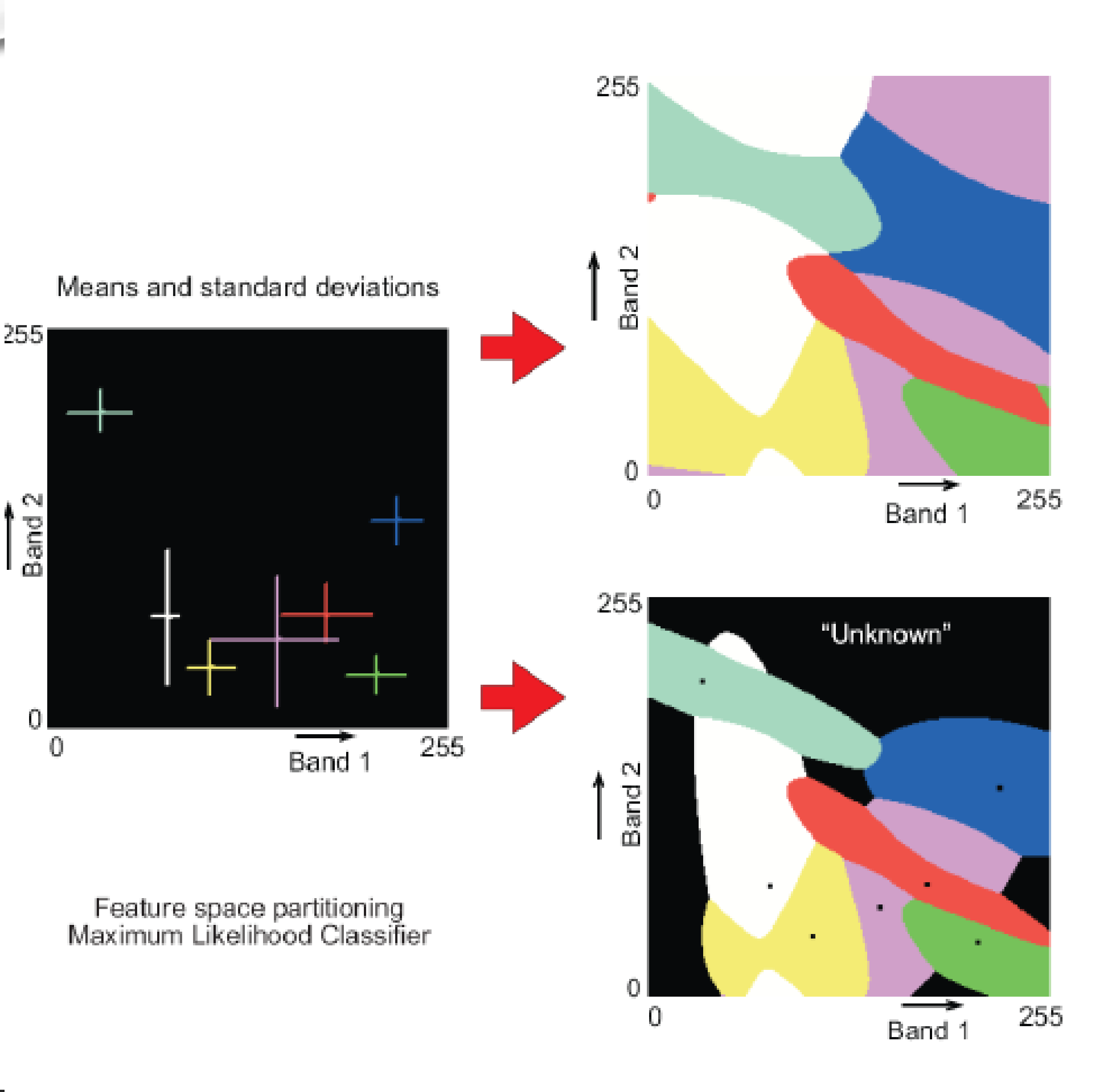
$\bar{x} = [ND_1, ND_2]$ $\quad$ $Classes: (\omega_1, \omega_2)$

$$P(\bar{x}/\omega_k) = \frac{P(\bar{x} \cap \omega_k)}{P(\omega_k)} \qquad P(\bar{x} \cap \omega_k) = P(\bar{x}/\omega_k) \cdot P(\omega_k)$$

$$P(\bar{x}/\omega_k) = \frac{1}{2\pi\sqrt{|\Sigma|}} \cdot e^{-1/2[(x-\mu)^T \cdot \Sigma^{-1} \cdot (x-\mu)]}$$

$$Predicted\ Class: \max(P(\bar{x} \cap \omega_k))$$

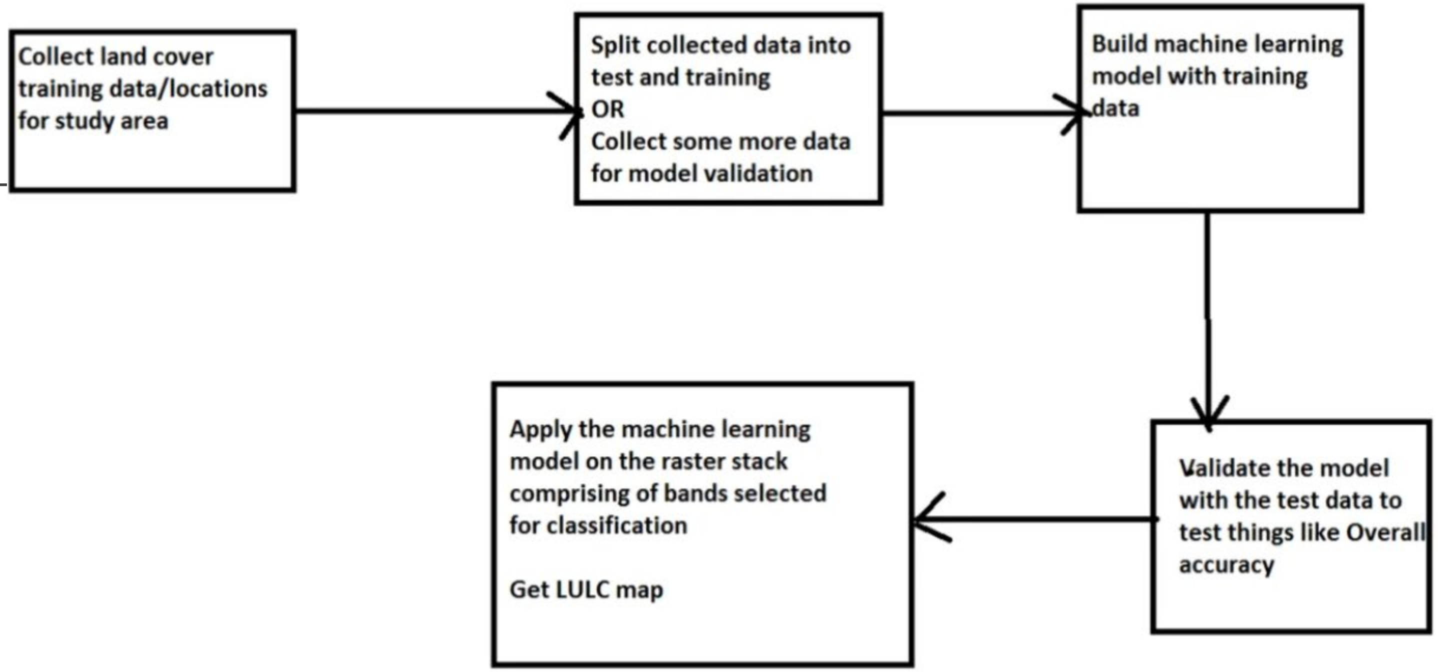
MAXIMUM LIKELIHOOD CLASSIFICATION (CONT.)



~~(right top) Classified feature space with the~~ decision functions without a threshold for the class "unknown".

(right bottom) Feature space classified with decision functions with such a threshold

MACHINE LEARNING IN REMOTE SENSING

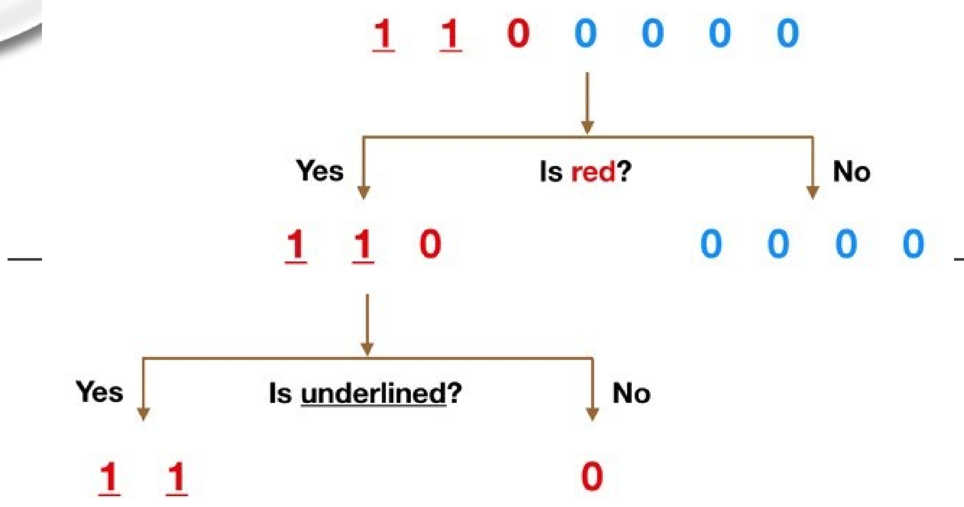


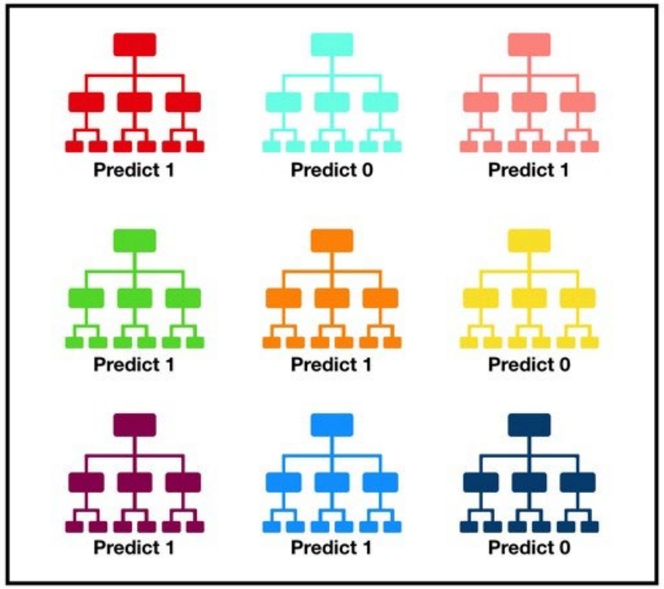
Division between training and validation sets

A classifier is trained with the training set and validated with the validation set.

Prediction. The classifier is applied on the rest of the data. These may include layers of calculated indices, texture maps and others.

MACHINE LEARNING: RANDOM FOREST



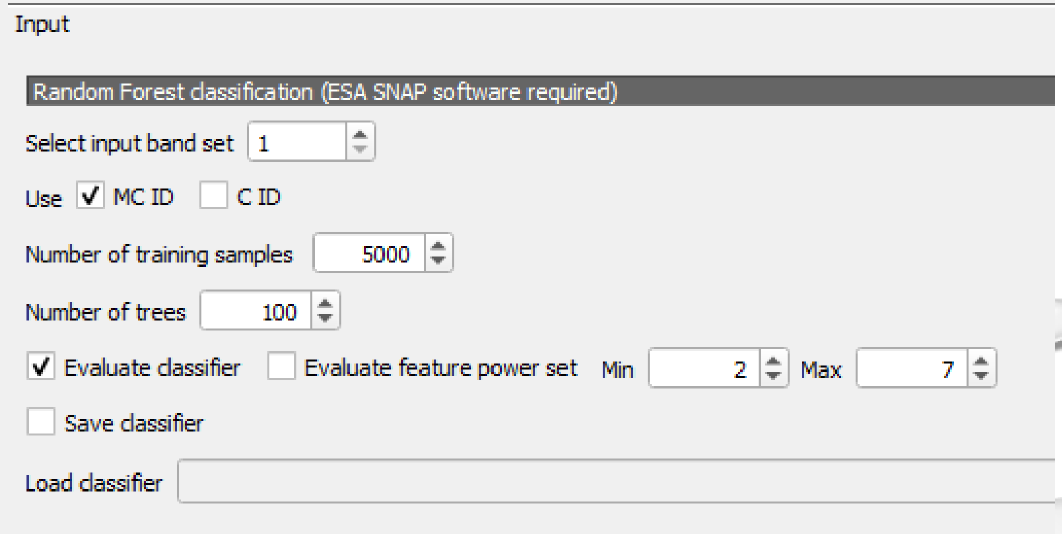


Tally: Six 1s and Three 0s
Prediction: 1

The "random forest" is composed of "decision trees".

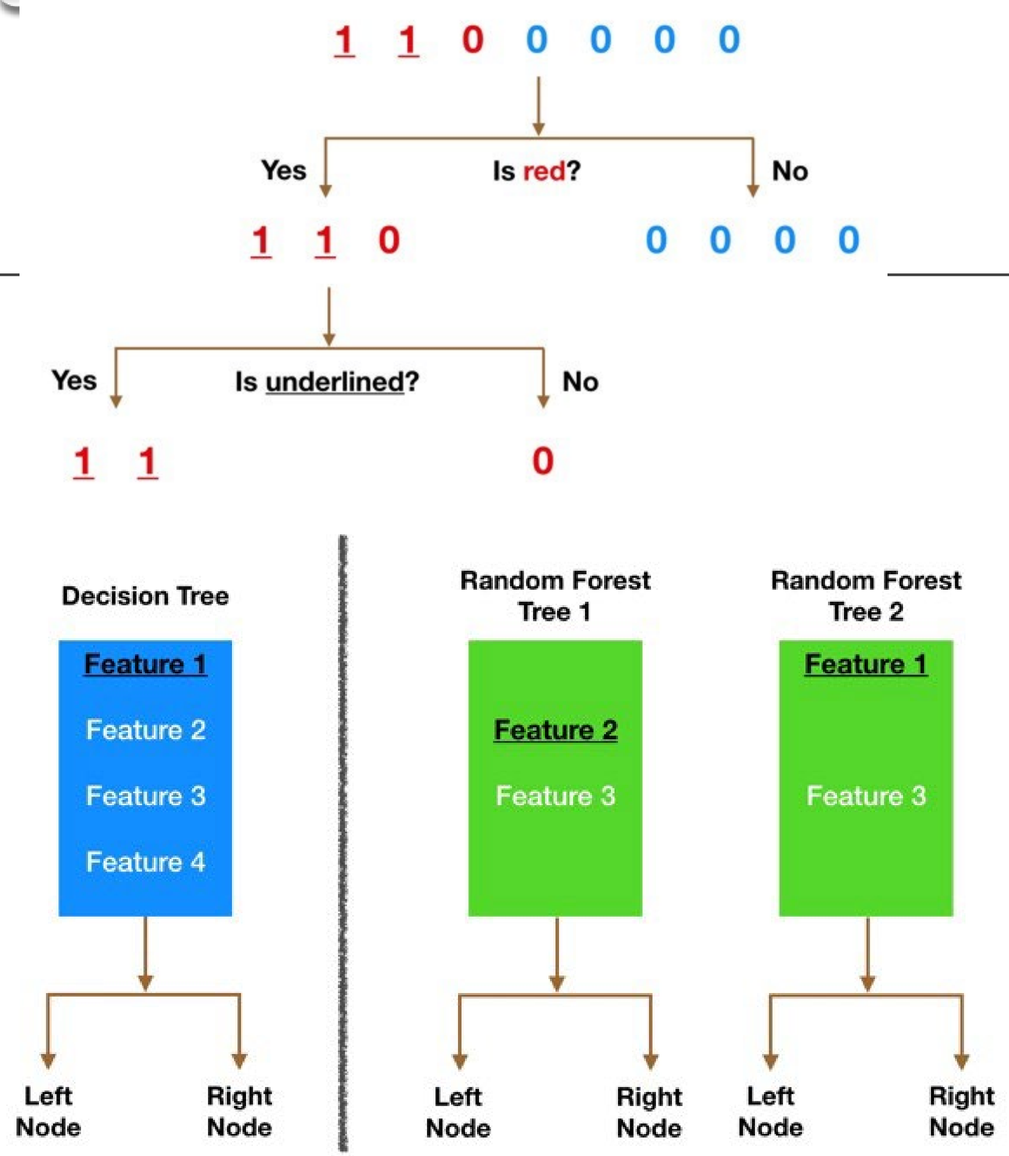
A large set of models (classifiers, trees) operating as an ensemble improve the performance of any individual component.

In the forest, several decision trees are trained as classifiers and their results are combined in a voting process.



Understanding Random Forest

MACHINE LEARNING: RANDOM FOREST (II)



[Understanding Random Forest](#)

Each tree is fed into the training with the set of samples with replacement from the training set.

At each node of the tree the algorithm only looks at a subset of the available descriptors (a much smaller number than the total).Each tree supplies a class which is called the vote of that tree. The most voted class in the whole forest will constitute the prediction of the forest.

In this way each tree is trained on different datasets thanks to the replacement and with different descriptor features (bands) to perform the classification.

The user controls only the number of training samples and the number of trees in the forest. The final result of the classification will be the one with the highest number of votes for each sample.

The classification result returns:

+ The importance of each descriptor (band).

+ The precision and accuracy in each class

CLASSIFIER VALIDATION

Result Validation

- An error matrix (also called confusión matrix can be used)
- Predicted classes and true classes are compared for each pixel in the validation set
- The validation set must also be selected from the original dataset and must be representative enough of each class variability (in size and dispersion, and spatial distribution in the image)

	A	B	C	D	Total	Error of Com-mission (%)	User Ac-curacy (%)
a	35	14	11	1	61	43	57
b	4	11	3	0	18	39	61
c	12	9	38	4	63	40	60
d	2	5	12	2	21	90	10
Total	53	39	64	7	163		
Error of Omission %	34	72	41	71			
Producer Accuracy %	66	28	59	29			

- In the case of the table in the image, 163 samples have been collected. A,B,C and D are ground truth classes and a,b,c,d are the classes obtained in classification.
- 53 pixels actually belonged to class A, 4 were classified as b, 12 as c and 2 as d.
- In the classification 61 pixels were classified as a.
- Elements of the main diagonal allow to calculate the overall accuracy: (35+11+38+2)/163=53%.
- With the elements of the rows the error of commission of a class (pixels of other classes classified as belonging to this class) can be calculated:Class a:(14+11+1)/61=26/61=43%.
- With the elements of the cols, the error of omission of a class can be calculated (pixels of this class classified as belonging to another class): Class a:(4+12+2)/53=18/53=34%.
- Commission error=User risk=Type II error;
- Omission error=Producer risk=Type I error

Other validation metrics: Precision and Recall (completeness)
Prec=TP/(TP+FP) ; Rec=TP/(TP+FN); F1=2*(Prec x Rec)/(Prec+Rec)