Remote Sensing & Photogrammetry W4

Beata Hejmanowska
Building C4, room 212,
phone: +4812 617 22 72
605 061 510
galia@agh.edu.pl

General procedures in image classification

- **Conventional multispectral classification techniques** perform class assignments based only on the spectral signatures of a classification unit.
- **Contextual classification** refers to the use of spatial, temporal, and other related information, in addition to the spectral information of a classification unit in the classification of an image. Usually, it is the pixel that is used as the classification unit.

General image classification procedures include

(1) Design image classification scheme: they are usually information classes such as urban, agriculture, forest areas, etc. Conduct field studies and collect ground infomation and other ancillary data of the study area.

(2) Preprocessing of the image, including radiometric, atmospheric, geometric and topographic corrections, image enhancement, and initial image clustering.

(3) Select representative areas on the image and analyze the initial clustering results or generate training signatures.

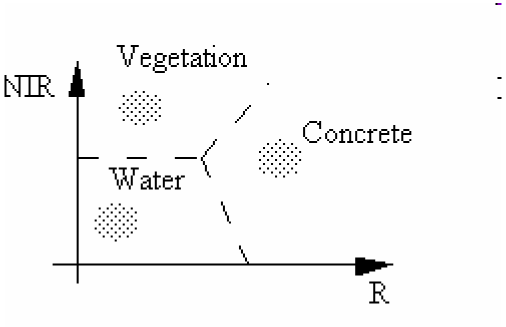
(4) Image classification

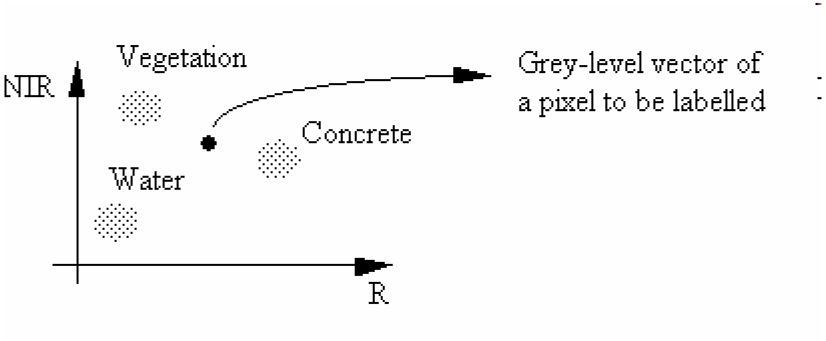
- Supervised mode: using training signature
- unsupervised mode: image clustering and cluster grouping

(5) Post-processing: complete geometric correction & filtering and classification decorating.

(6) Accuracy assessment: compare classification results with field studies.

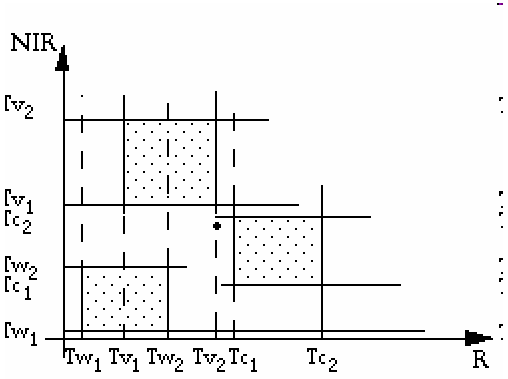
Supervised classification





There are two obvious ways of classifying this pixel

(1) Multidimensional thresholding
As in the above diagram, we define two threshold values along each axis for each class. A grey-level vector is classified into a class only if it falls between the thresholds of that class along each axis.
The advantage of this algorithm is its simplicity. The drawback is the difficulty of including all possible grey-level vectors into the specified class thresholds. It is also difficult to properly adjust the class thresholds.
(2) Minimum-Distance Classification



Minimum-Distance Classification

The closer the two points, the more likely they are in the same class.
We can use various types of distance as similarity measures to develop a classifier, i.e. minimum-distance classifier.

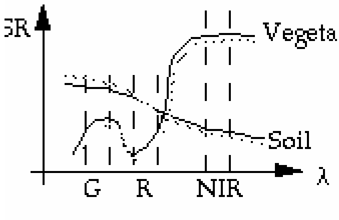


Fig. 1

⇒

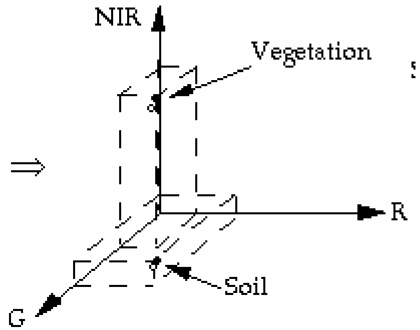
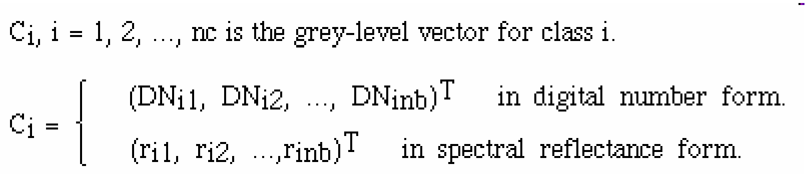


Fig. 2

In a minimum-distance classifier, suppose we have nc known class centers
C = {C1, C2, ..., Cnc}, Ci, i = 1, 2, ..., nc is the grey-level vector for class i.



C_i, i = 1, 2, ..., nc is the grey-level vector for class i.

$$C_i = \begin{cases} (DN_{i1},\ DN_{i2},\ \ldots,\ DN_{inb})^T & \text{in digital number form.} \\ (r_{i1},\ r_{i2},\ \ldots, r_{inb})^T & \text{in spectral reflectance form.} \end{cases}$$

3 classes (nc = 3) and two spectral bands (nb = 2)

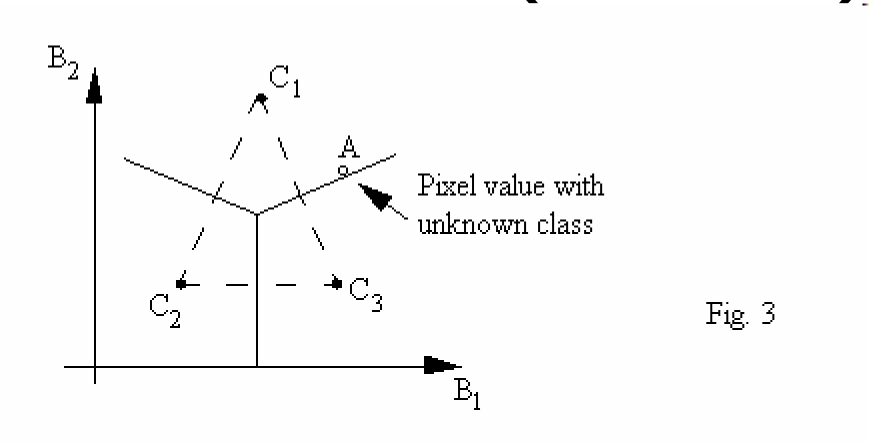


Fig. 3

If we have a pixel with a grey-level vector located in the B1-B2 space shown as A (an empty dot), we are asked to determine to which class it should belong. We can calculate the distances between A and each of the centers. A is assigned to the class whose center has the shortest distance to A.

In a general form, an arbitrary pixel with a grey-level vector g = (g1, g2, ..., gnb)^T, is classified as C_i if

$$d(C_i, g) = \min (d(C_{i1},g1), d(C_{i2},g2), ..., d(C_{inb},gnb))$$

Now, in what form should the distance d take?

In a general form, an arbitrary pixel with a grey-level vector g = (g1, g2, ..., gnb)T, is classified as Ci if

$d(C_i, g) = min\ (d(C_{i1},g1), d(C_{i2},g2), ..., d(C_{inb},gnb))$

The most-popularly used form is the Euclidian distance

$$d_e(Ci,g) = \begin{cases} \sqrt{\sum_{j=1}^{nb} (C_{ij} - g_j)^2} & \text{Numerical form.} \\ [(C_i - g)^T (C_i - g)]^{1/2} & \text{matrix form.} \end{cases}$$

The second popularly used distance is Mahalanobis distance

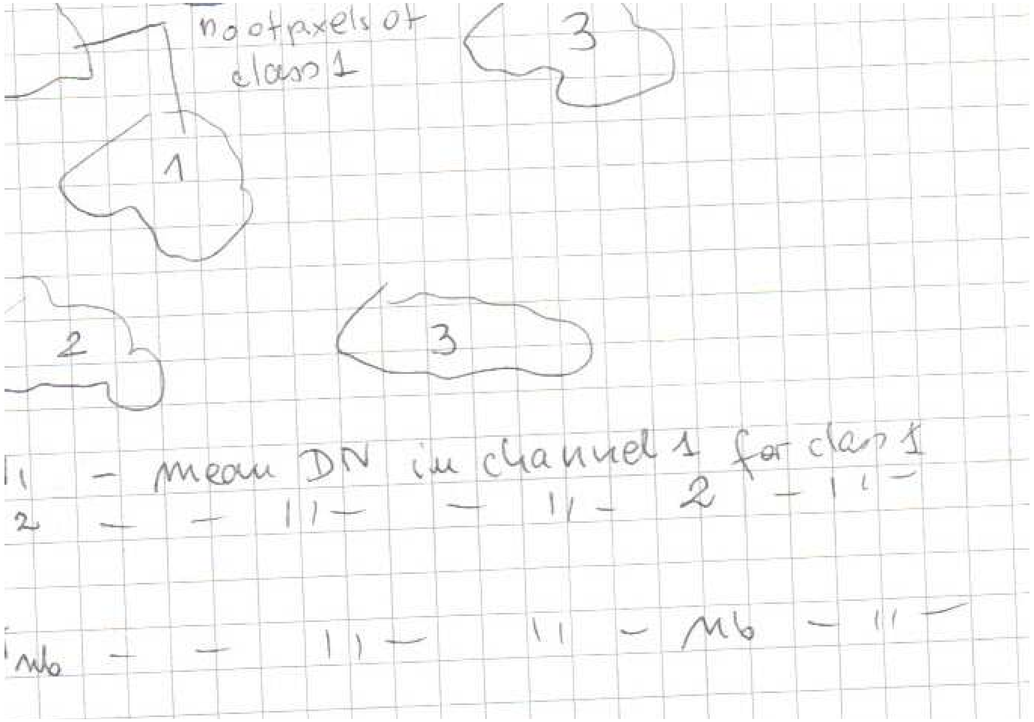
$$d_m(C_i,g) = [(g - C_i)^T V^{-1}(g - C_i)]^{1/2}$$

where V^{-1} is the inverse of the covariance matrix of the data.

If the Mahalanobis distance is used, we call the classifier as a **Mahalanobis Classifier**

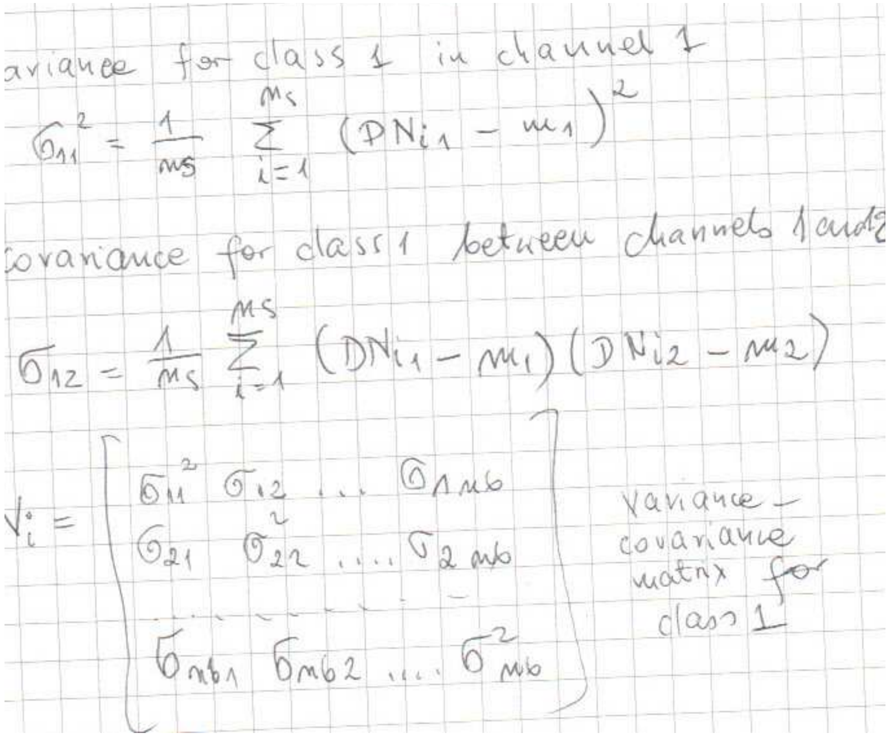
Variance covariance matrix

n_s



1 – mean DN in channel 1 for clas 1
2 – " – " – " – 2 – " –

nb – " – " – " – nb – " –



ariance for class 1 in channel 1

$$\sigma_{11}^2 = \frac{1}{ns} \sum_{i=1}^{ns} (DN_{i1} - m_1)^2$$

ovariance for class 1 between channels 1 and 2

$$\sigma_{12} = \frac{1}{ns} \sum_{i=1}^{ns} (DN_{i1} - m_1)(DN_{i2} - m_2)$$

$$V_i = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12} & \dots & \sigma_{1nb} \\ \sigma_{21} & \sigma_{22}^2 & \dots & \sigma_{2nb} \\ \dots & \dots & \dots & \dots \\ \sigma_{nb1} & \sigma_{nb2} & \dots & \sigma_{nb}^2 \end{bmatrix}$$

variance – covariance matrix for class 1

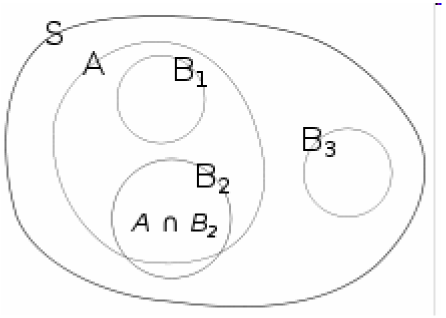
The simplest distance measure is the city-block distance

Class centers C and the data covariance matrix V are usually determined from training samples if a supervised classification procedure is used. They can also be obtained from clustering.

$$d_c(C_i, g) = \sum_{j=1}^{nb} |C_{ij} - g_j|$$

Conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$



$P(A|B_1) = 1$ $\quad$ $P(A|B_3) = 0$

$P(A|B_2)$ *=0.85*

Bayes' theorem

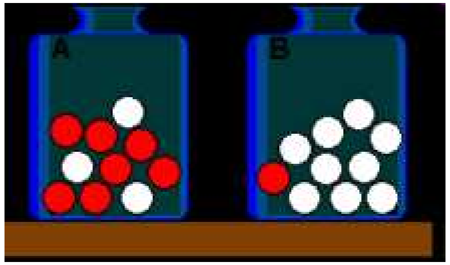
$$P(X \cap T) = P(X|T)P(T) = P(T|X)P(X) \iff P(T|X) = P(X|T)\frac{P(T)}{P(X)}$$

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}.$$

$$P(B) = \sum_{j=1}^{N} P\left(A_j \cap B\right) = \sum_{j=1}^{N} P\left(B \mid A_j\right) \cdot P\left(A_j\right)$$

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}.$$

Example



A result „ball is from box A“,
B result „ball is from box B“ und
R result „ball is red“

$$P(A) = P(B) = \frac{1}{2}$$ (probability of box choice 1/2)

$$P(R|A) = \frac{7}{10}$$ (in box A – 10 balls, 7 red)

$$P(R|B) = \frac{1}{10}$$ (in box B - 10 balls, 1 red)

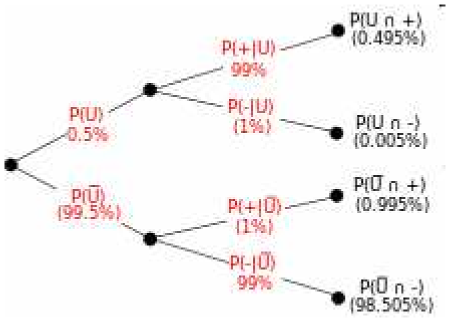
$$P(R) = P(R|A) \cdot P(A) + P(R|B) \cdot P(B) = \frac{7}{10} \cdot \frac{1}{2} + \frac{1}{10} \cdot \frac{1}{2} = \frac{2}{5}$$ (total probability)

$$P(A|R) = \frac{P(R|A) \cdot P(A)}{P(R)} = \frac{\frac{7}{10} \cdot \frac{1}{2}}{\frac{2}{5}} = \frac{7}{8}$$ Probabality choice box A if the ball is red

Example - drug testing

Suppose a drug test is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. If a randomly selected individual tests positive, what is the probability they are a user?

$$
\begin{aligned}
P(\text{User}|+) &= \frac{P(+|\text{User})P(\text{User})}{P(+|\text{User})P(\text{User}) + P(+|\text{Non-user})P(\text{Non-user})} \\
&= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995} \\
&\approx 33.2\%
\end{aligned}
$$



Tree diagram illustrating drug testing example. U, U bar, "+" and "−" are the events representing user, non-user, positive result and negative result. Percentages in parentheses are calculated.

Example - drug testing

Despite the apparent accuracy of the test, if an individual tests positive, it is more likely that they do *not* use the drug than that they do.
This surprising result arises because the number of non-users is very large compared to the number of users, such that the number of false positives (0.995%) outweighs the number of true positives (0.495%). To use concrete numbers, if 1000 individuals are tested, there are expected to be 995 non-users and 5 users. From the 995 non-users, $0.01 \times 995 \approx 10$ false positives are expected. From the 5 users, $0.99 \times 5 \approx 5$ true positives are expected. Out of 15 positive results, only 5, about 33%, are genuine.

Maximum Likelihood Classification (MLC)

MLC is the most common classification method used for remotely sensed data. MLC is based on the Baye's rule.

Let C = (C1, C2, ..., Cnc) denote a set of classes, where nc is the total number of classes. For a given pixel with a grey-level vector x, the probability that x belongs to class ci is

P(Ci|x), i = 1, 2, ..., nc.

If P(Ci|x) is known for every class, we can determine into which class x should be classified._This can be done by comparing P(Ci|x)'s, i = 1, 2, ..., nc.

x => ci, if P(Ci|x) > P(Cj|x) for all j # i. (1)

However, P(Ci|x) is not known directly. Thus, we use Baye's theorem:

P(Ci|x) = p(x|Ci) ï P(Ci)/P(x)

where

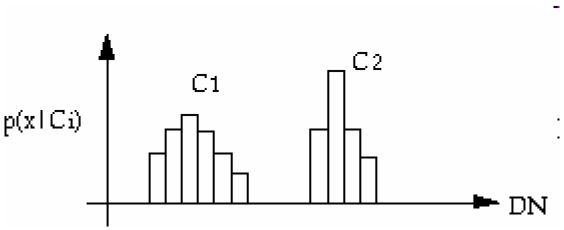
P(Ci) is the probability that Ci occurs in the image. It is called *a priori* probability.

P(x) is the probability of x occurring in each class ci.

$$P(x) = \sum_{i=1}^{nc} p(x \mid C_i) \bullet P(C_i)$$

Maximum Likelihood Classification (MLC)

- However, P(x) is not needed for the classification purpose because if we compare P(C1|x) with P(C2|x), we can cancel P(x) from each side. Therefore, p(x|Ci) i = 1, 2, ..., nc are the conditional probabilities which have to be determined. One solution is through statistical modelling. This is done by assuming that the conditional probability distribution function (PDF) is normal (also called, Gaussian distribution). If we can find the PDF for each class and the *a priori* probability, the classification problem will be solved. For p(*x|ci) we use training samples.



Maximum Likelihood Classification (MLC)

For one-dimensional case, we can see from the above figure that by generating training statistics of two classes, we have their probability distributions. If we use these statistics directly, it will be difficult because it requires a large amount of computer memory. The Gaussian normal distribution model can be used to save the memory. The one-dimensional Gaussian distribution is:

$$p(x|C_i) = \frac{1}{\sqrt{2\pi} \bullet \delta_i} \cdot \exp\{-(x-\mu_i)^2/(2\delta_i^2)\}$$

- where we only need two parameter for each class µi and   , i = 1, 2, ..., nc
- μ_i the mean for C_i
  σ_i the standard deviation of C_i
- μ_i, σ_i can be easily generated from training sample.

Maximum Likelihood Classification (MLC)

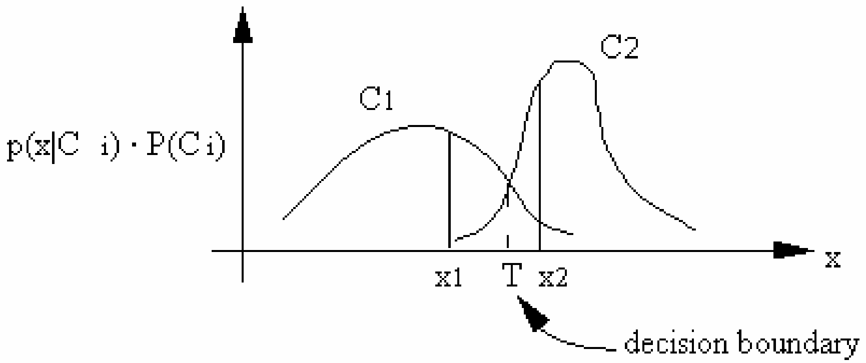
For higher dimensions,

$$p(x|C_i) = \frac{1}{(2\pi)^{nb/2} \cdot \sqrt{|V_i|}} \exp\{-\frac{1}{2}(x - \mu_i)^T V_i^{-1} \cdot (x - \mu_i)\}$$

- where nb is the dimension (number of bands)
- μ_i is the mean vector of c_i
- V_i is the covariance matrix of C_i
- $P(C_i)$ can also be determined with knowledge about an area. If they are not known, we can assume that each class has an equal chance of occurrence.
- i.e. P(C1) = P(C2) = ... = P(Cnc)

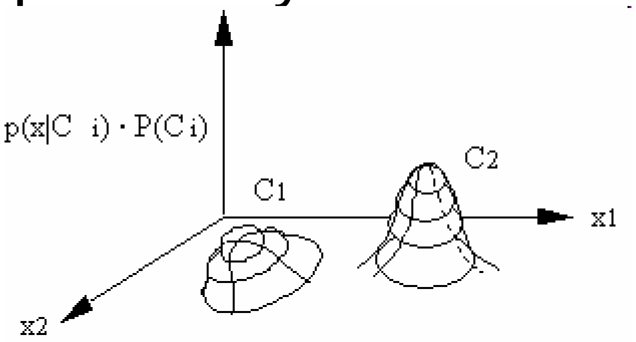
Maximum Likelihood Classification (MLC)

With the knowledge of p(x|Ci) and P(Ci), we can conduct maximum likelihood classification. p(x|Ci) ï P(Ci) i = 1, 2, ..., nc can be compared instead of P(Ci|x) in (1).



Maximum Likelihood Classification (MLC)

The interpretation of the maximum likelihood classifier is illustrated in the above figure. An x is classified according to the maximum p(x|Ci) ï P(Ci). x1 is classified into C1, x2 is classified into C2. The class boundary is determined by the point of equal probability.



In two-dimensional space, the class boundary cannot be easily determined. Therefore we don't use boundaries in maximum likelihood classification and, instead, we compare probabilities.

Actual implementation of MLC

In order to simplify the computation, we usually take a logarithm of p(x|Ci). P(Ci)

$$\log\{p(x|C_i)\cdot P(C_i)\} = -nb/2 \cdot \log 2\pi - \frac{1}{2}\log|V_i| - \frac{1}{2}(x-\mu_i)^T - V_i^{-1}(x-\mu_i) + \log(P(C_i)) \qquad (1)$$

Since - nb/2 ï log 2p is a constant, the RHS can be simplified to

$$g(x) = -\frac{1}{2}\log|V_i| - \frac{1}{2}(x-\mu_i)^T V_i^{-1}(x-\mu_i) + \log P(C_i) \qquad (2)$$

Often, we assume P(Ci) is the same for each class. Therefore (2) can be further simplified to

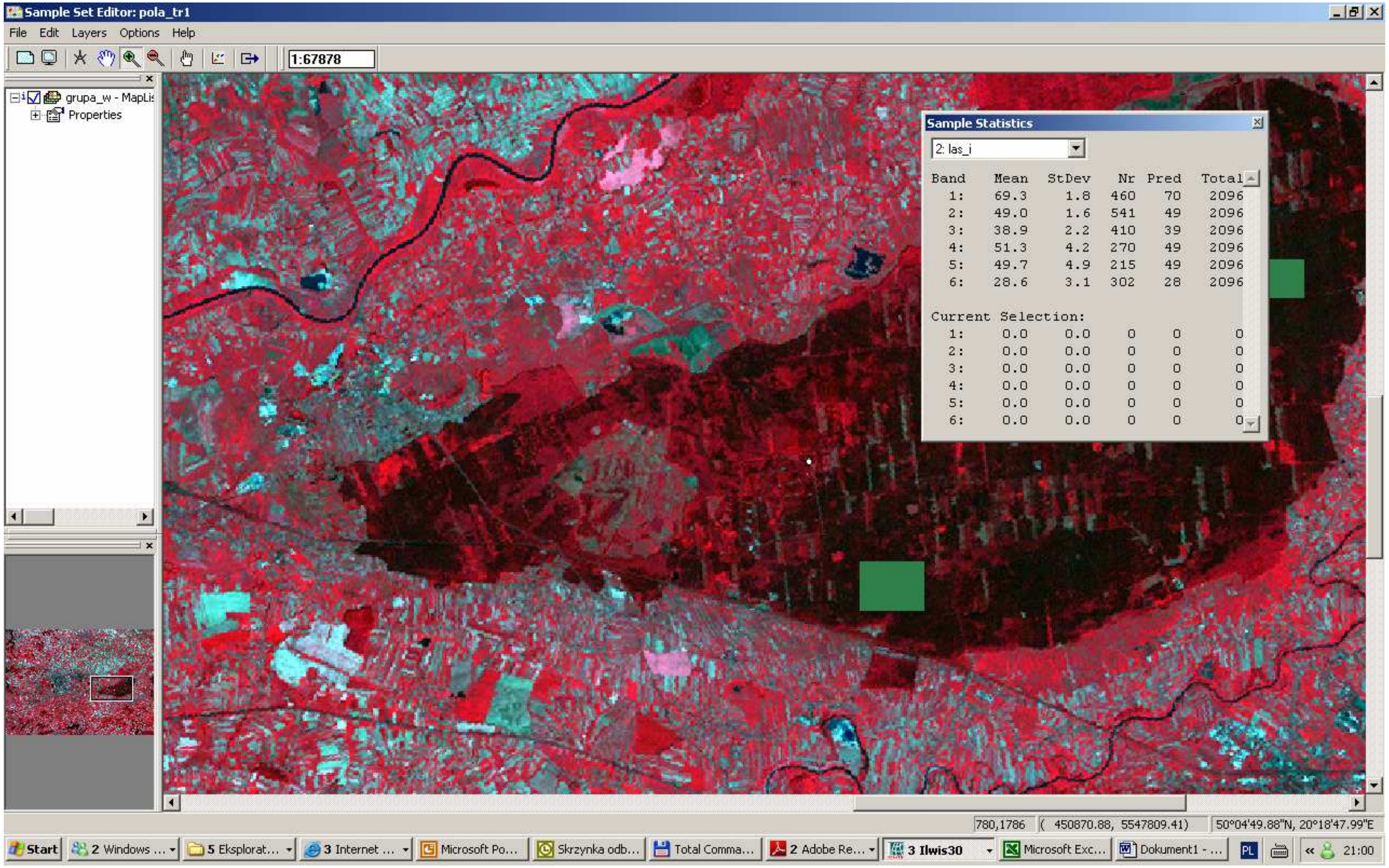
$$g(x) = -\log|V_i| - (x-\mu_i)^T V_i^{-1}(x-\mu_i) \qquad (3)$$

g(x) is referred to as the discriminant function.

By comparing g(x)'s, we can assign x to the proper class

- With the maximum likelihood classifier, it is guaranteed that the error of misclassification is minimal if p(x|Ci) is normally distributed.
- Unfortunately, the normal distribution cannot always be achieved. In order to make the best use of the MLC method, one has to make sure that his training sample will generate distributions as close to the normal distribution as possible.
- How large should one's training sample be? Usually, one needs 10 x nb, preferably 100 x nb, pixels in each class (Swain and Davis, 1978).
- MLC is relatively robust but it has the limitation when handling data at nominal or ordinal scales. The computational cost increases considerably as the image dimensionality increases.

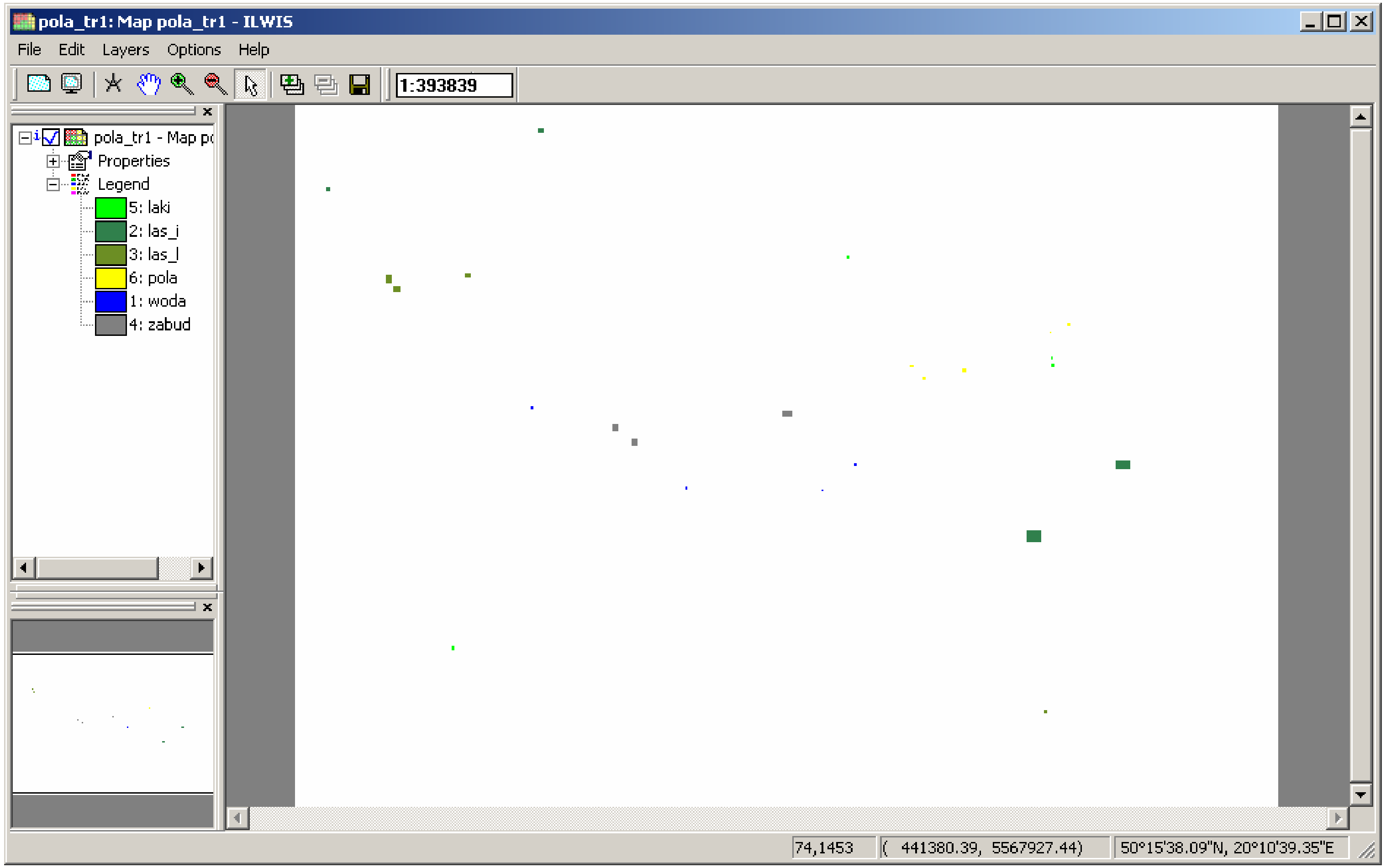
Training fields



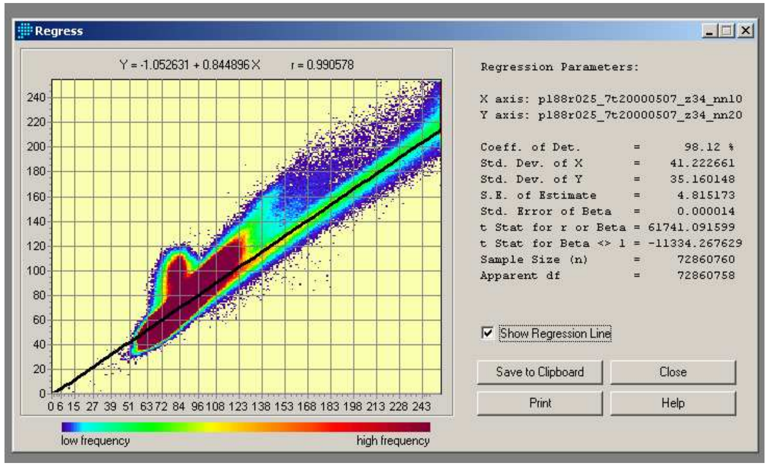
```
Band     Mean   StDev    Nr  Pred   Total
  1:     69.3     1.8   460    70    2096
  2:     49.0     1.6   541    49    2096
  3:     38.9     2.2   410    39    2096
  4:     51.3     4.2   270    49    2096
  5:     49.7     4.9   215    49    2096
  6:     28.6     3.1   302    28    2096

Current Selection:
  1:      0.0     0.0     0     0       0
  2:      0.0     0.0     0     0       0
  3:      0.0     0.0     0     0       0
  4:      0.0     0.0     0     0       0
  5:      0.0     0.0     0     0       0
  6:      0.0     0.0     0     0       0
```

Training fields



Base of automatic classification

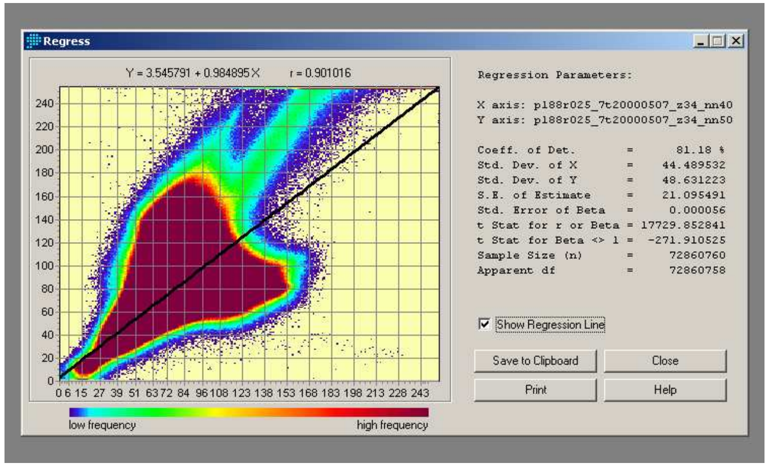


```
Regression Parameters:

X axis: p188r025_7t20000507_z34_nn10
Y axis: p188r025_7t20000507_z34_nn20

Coeff. of Det.          =      98.12 %
Std. Dev. of X          =    41.222661
Std. Dev. of Y          =    35.160148
S.E. of Estimate        =     4.815173
Std. Error of Beta      =     0.000014
t Stat for r or Beta = 61741.091599
t Stat for Beta <> 1 = -11334.267629
Sample Size (n)         =     72860760
Apparent df             =     72860758
```

☑ Show Regression Line

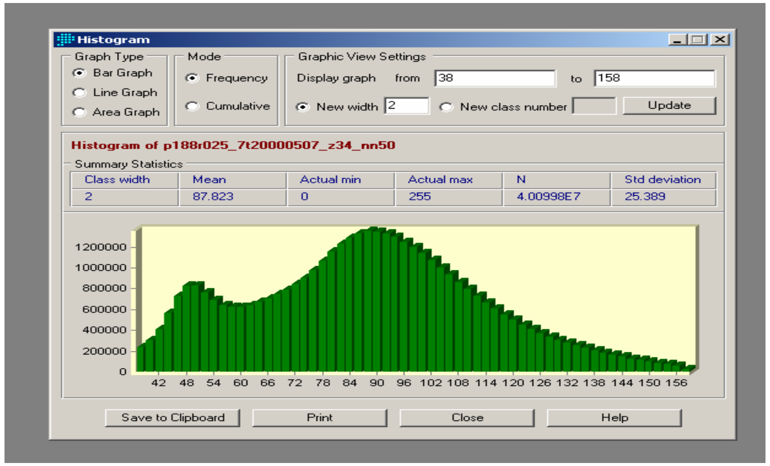


```
Regression Parameters:

X axis: p188r025_7t20000507_z34_nn40
Y axis: p188r025_7t20000507_z34_nn50

Coeff. of Det.          =      81.18 %
Std. Dev. of X          =    44.489532
Std. Dev. of Y          =    48.631223
S.E. of Estimate        =    21.095491
Std. Error of Beta      =     0.000056
t Stat for r or Beta = 17729.852841
t Stat for Beta <> 1 =  -271.910525
Sample Size (n)         =     72860760
Apparent df             =     72860758
```

☑ Show Regression Line



Graph Type: ● Bar Graph ○ Line Graph ○ Area Graph

Mode: ● Frequency ○ Cumulative

Graphic View Settings: Display graph from 38 to 158

● New width 2 ○ New class number

Histogram of p188r025_7t20000507_z34_nn50

Summary Statistics

Class width	Mean	Actual min	Actual max	N	Std deviation
2	87.823	0	255	4.00998E7	25.389

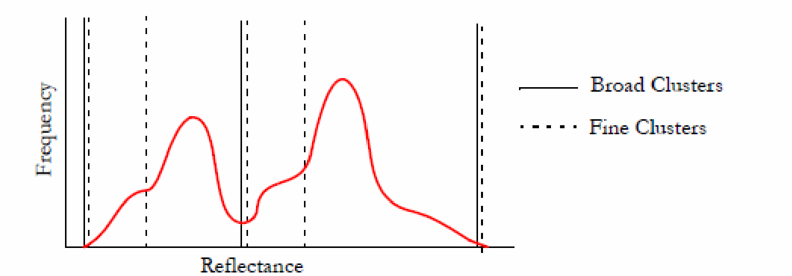
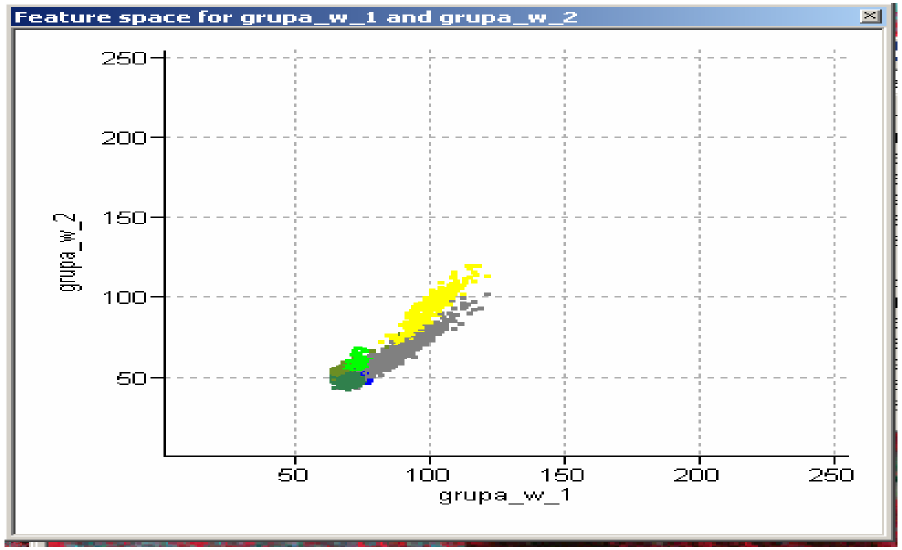
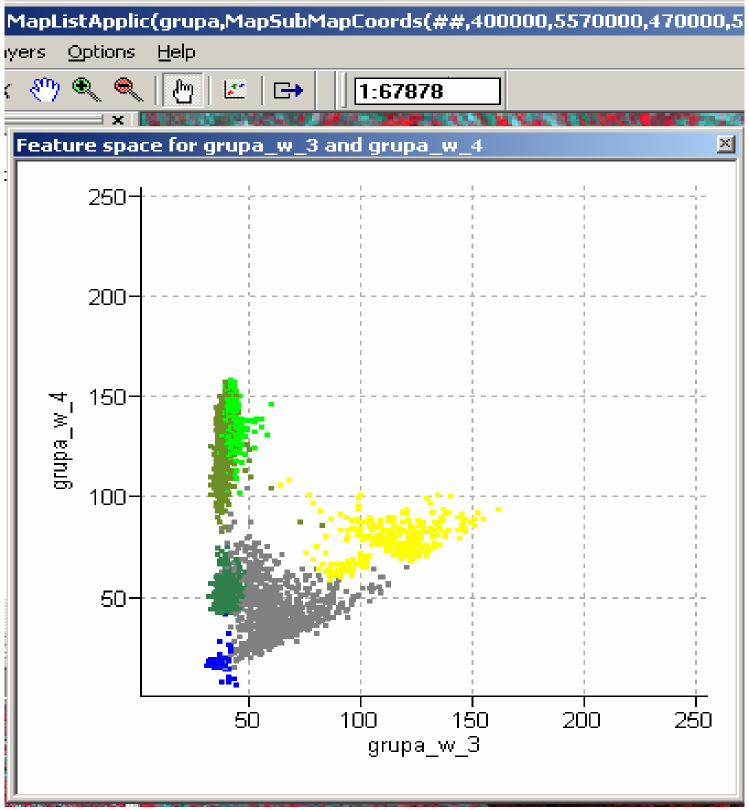


Figure 1

Statistics of training fields





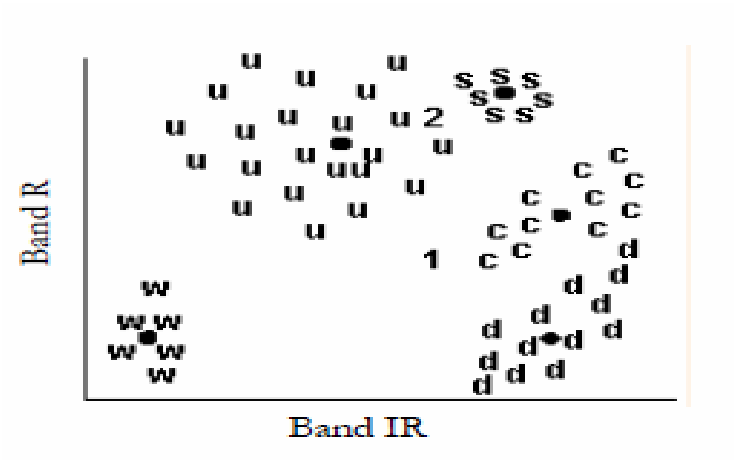
Classification methods

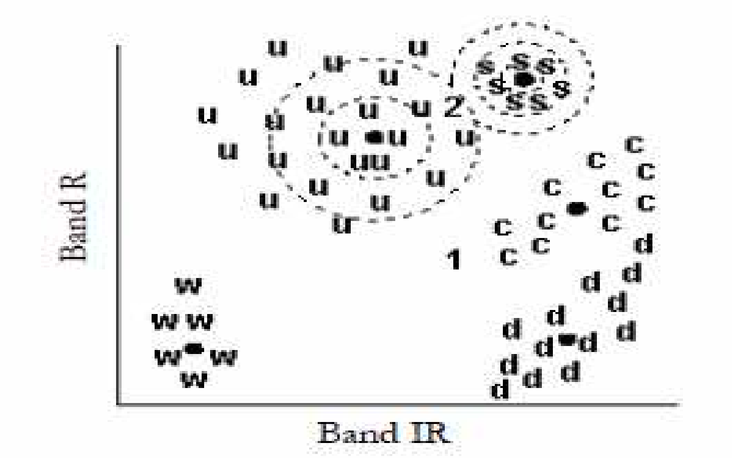
- The following classification methods are available:
- Box classifier, using a multiplication factor,
- Minimum Distance, optionally using a threshold value,
- Minimum Mahalanobis distance, optionally using a threshold value,
- Maximum Likelihood, optionally using a threshold value,
- Maximum Likelihood including Prior Probabilities, optionally using a threshold value.

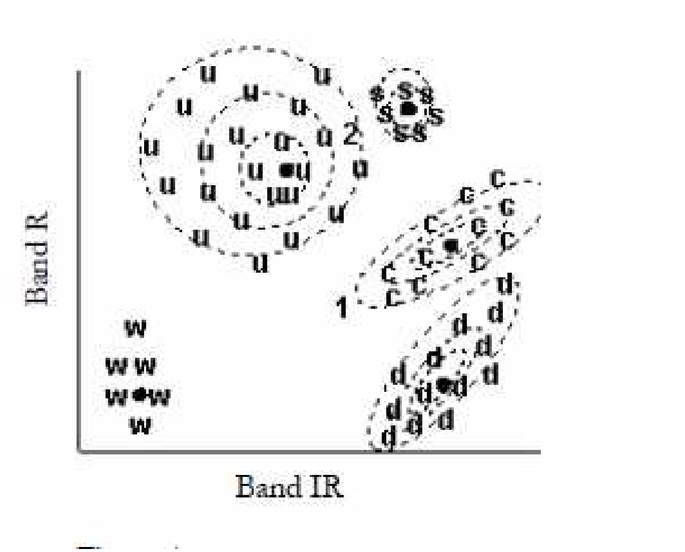
Classification methods

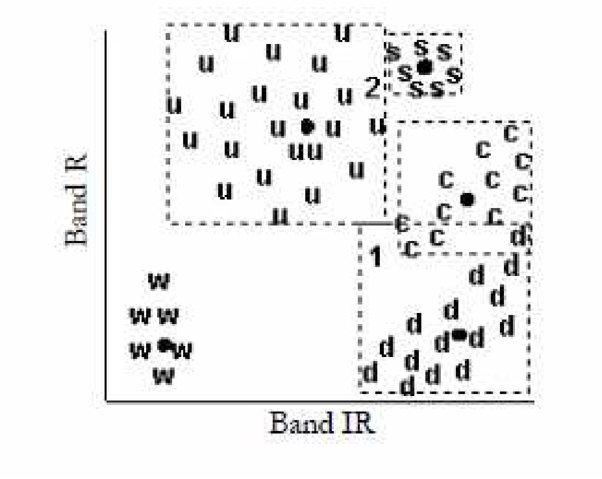
- Prior to any classification, empirical statistics are drawn from the training pixels in the input sample set. These sample statistics are calculated per class of training pixels and per band. For instance, for a single class (*i*), *n* mean values are calculated when there are *n* input bands; these *n* mean values together are called the class mean (vector) for that class (**m**i).
- Depending on the selected classification method, the following statistics are calculated:
- for each *class i* of training pixels:
  - the means of training pixels per band (**m**i),
  - in case of box classifier: the variance of the training pixels per band,
  - the standard deviation of the training pixels per band (should be > 0),
  - the predominant value (mode) per band,
  - in case of Minimum Mahalanobis distance, Maximum Likelihood and Prior Probability classifier: an *n* x *n* variance-covariance matrix (Vi) which stores class variance per band, and class covariance between bands.
- For each feature vector to be classified, these statistics are used to calculate the shortest 'distance' towards the training classes. All classification decisions are thus based on these statistical empirical parameters.

Supervised classification









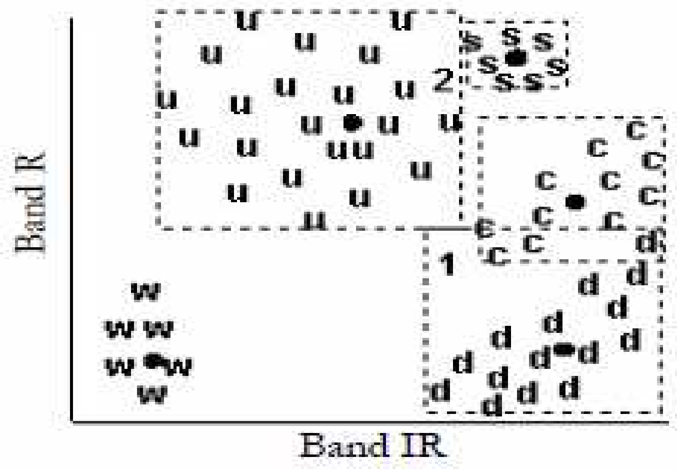
Box classifier

For each class, a multi-dimensional box is drawn around the class mean.

For each class, the size of the box is calculated as:

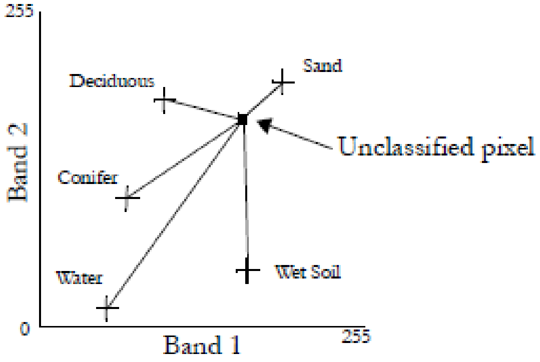
(class mean ± standard deviation per band) * multiplication factor

- If a feature vector falls inside a box, then the corresponding class name is assigned.
- if a feature vector falls within two boxes, the class name of the box with the smallest product of standard deviations is assigned, i.e. the class name of the smallest box.
- if a feature vector does not fall within a box, the undefined value is assigned.

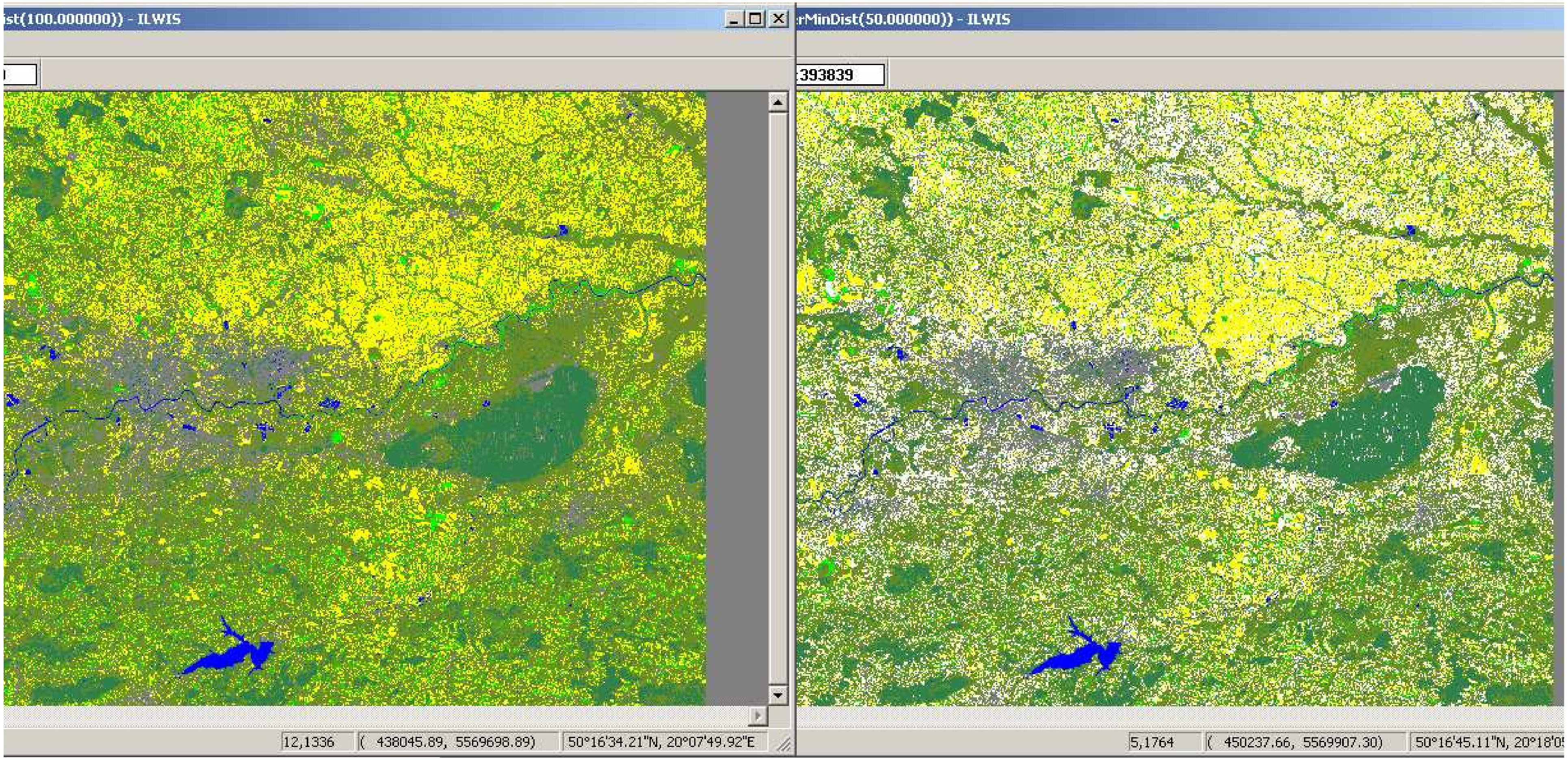


Minimum Distance to Mean

- For each feature vector, the distances towards class means are calculated.
- The shortest Euclidian distance to a class mean is found;
- if this shortest distance to a class mean is smaller than the user-defined threshold, then this class name is assigned to the output pixel.
- else the undefined value is assigned.



Mindist (100 i 50)



Minimum Mahalanobis distance:

For each feature vector, the Mahalanobis distances towards class means are calculated. This includes the calculation of the variance-covariance matrix V for each class *i*.

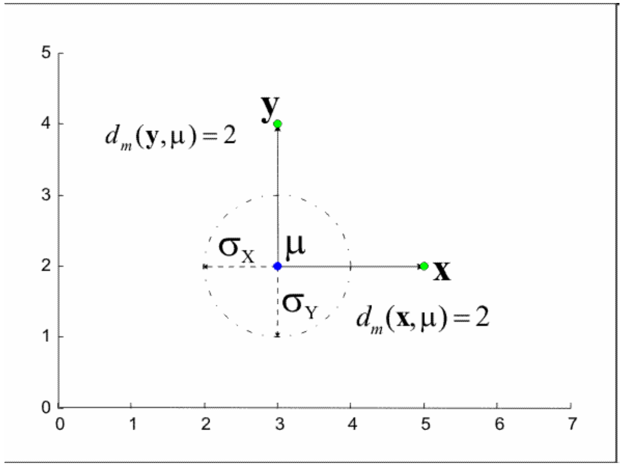
The Mahalanobis distance is calculated as:

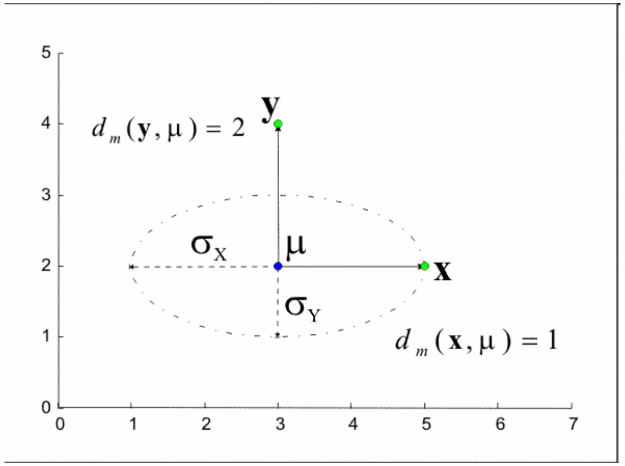
$$\mathrm{di}(\mathbf{x}) = \mathbf{y}^T V_i^{-1} \mathbf{y}$$

For an explanation of the parameters, see Maximum Likelihood classifier.

- For each feature vector **x**, the shortest Mahalanobis distance to a class mean is found;
- if this shortest distance to a class mean is smaller than the user-defined threshold, then this class name is assigned to the output pixel.
- else the undefined value is assigned.

Machalanobis distance





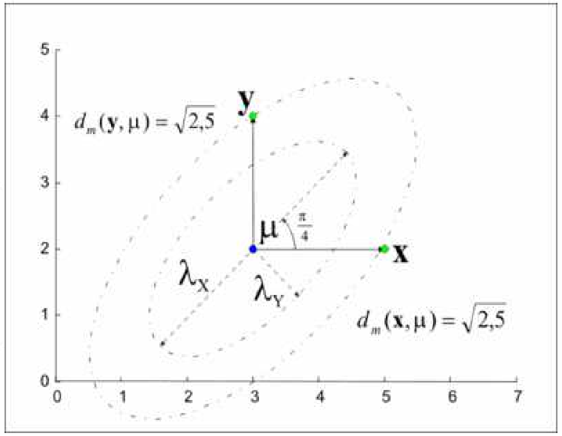
$$d_m(\mathbf{x},\boldsymbol{\mu}) = \sqrt{(x_1-\mu_1)^2 + \ldots + (x_n-\mu_n)^2}$$
$$= \sqrt{(\mathbf{x}-\boldsymbol{\mu})\mathbb{I}^{-1}(\mathbf{x}-\boldsymbol{\mu})^T}$$

$$d_m(\mathbf{x},\boldsymbol{\mu}) = \sqrt{\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \ldots + \frac{(x_n-\mu_n)^2}{\sigma_n^2}}$$
$$= \sqrt{(\mathbf{x}-\boldsymbol{\mu})D^{-1}(\mathbf{x}-\boldsymbol{\mu})^T}$$

gdzie D jest macierzą diagonalną $\mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$.

Machalanobis distance

di(**x**) = $\mathbf{y}^T V_i^{-1} \mathbf{y}$



Maximum Likelihood

For each feature vector, the distances towards class means are calculated. This includes the calculation of the variance-covariance matrix V for each class *i*.

The formula used in Maximum Likelihood reads:

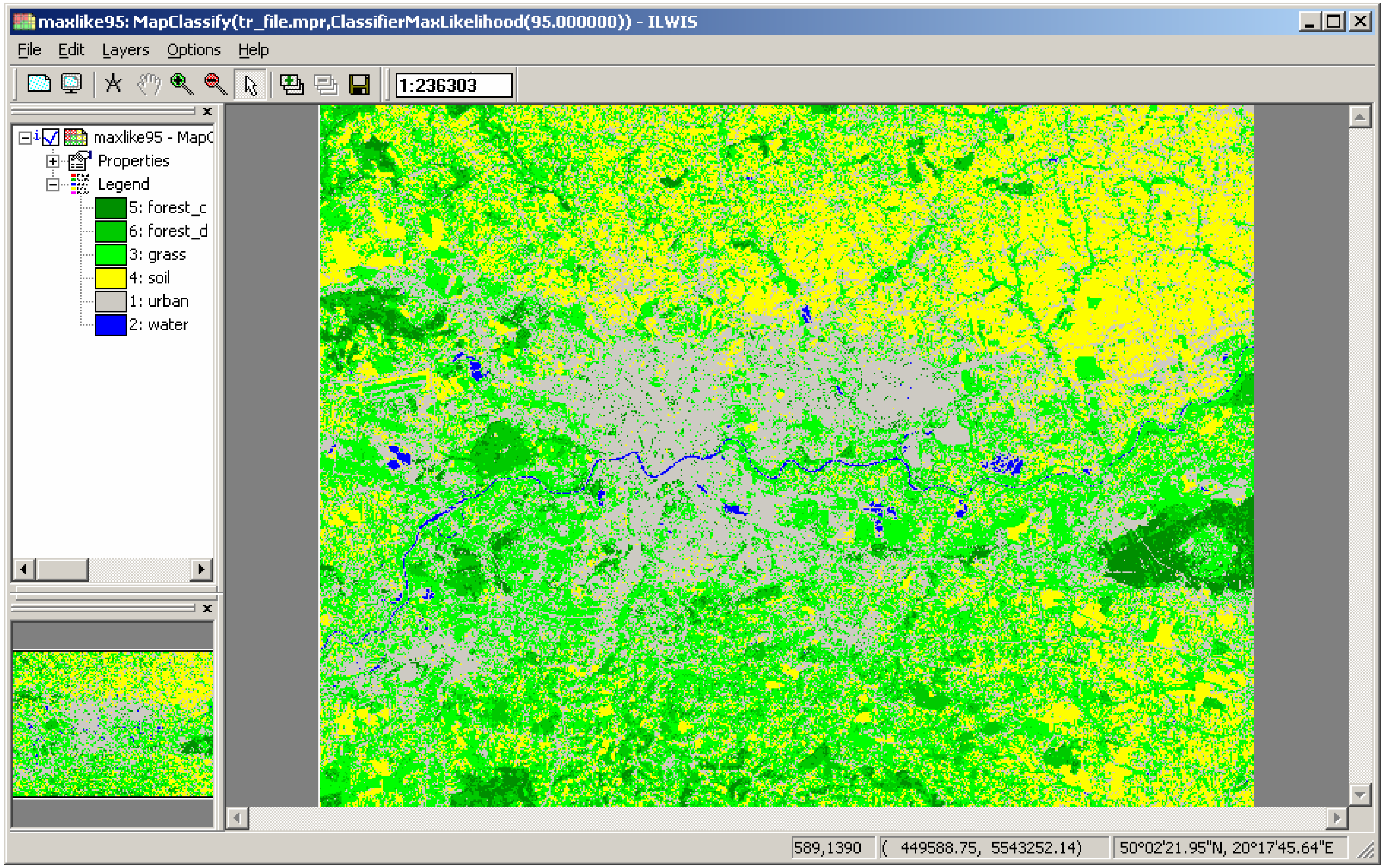
$d_i(\mathbf{x}) = \ln|V_i| + \mathbf{y}^T V_i^{-1} \mathbf{y}$

where:

- d_i - distance between feature vector (**x**) and a class mean ($\mathbf{m}_i$) based on probabilities
- V_i - the *n* x *n* variance-covariance matrix of class *i*, where *n* is the number of input bands
- $|V_i|$ - determinant of V_i
- V_i^{-1} - the inverse of V_i
- **Y - x** - **m**i ; is the difference vector between feature vector **x** and class mean vector $\mathbf{m}_i$
- $\mathbf{y}^T$ - the transposed of **y**

- For each feature vector **x**, the shortest distance di to a class mean $\mathbf{m}_i$ is found;
- if this shortest distance to a class mean is smaller than the user-defined threshold, then this class name is assigned to the output pixel.
- else the undefined value is assigned.

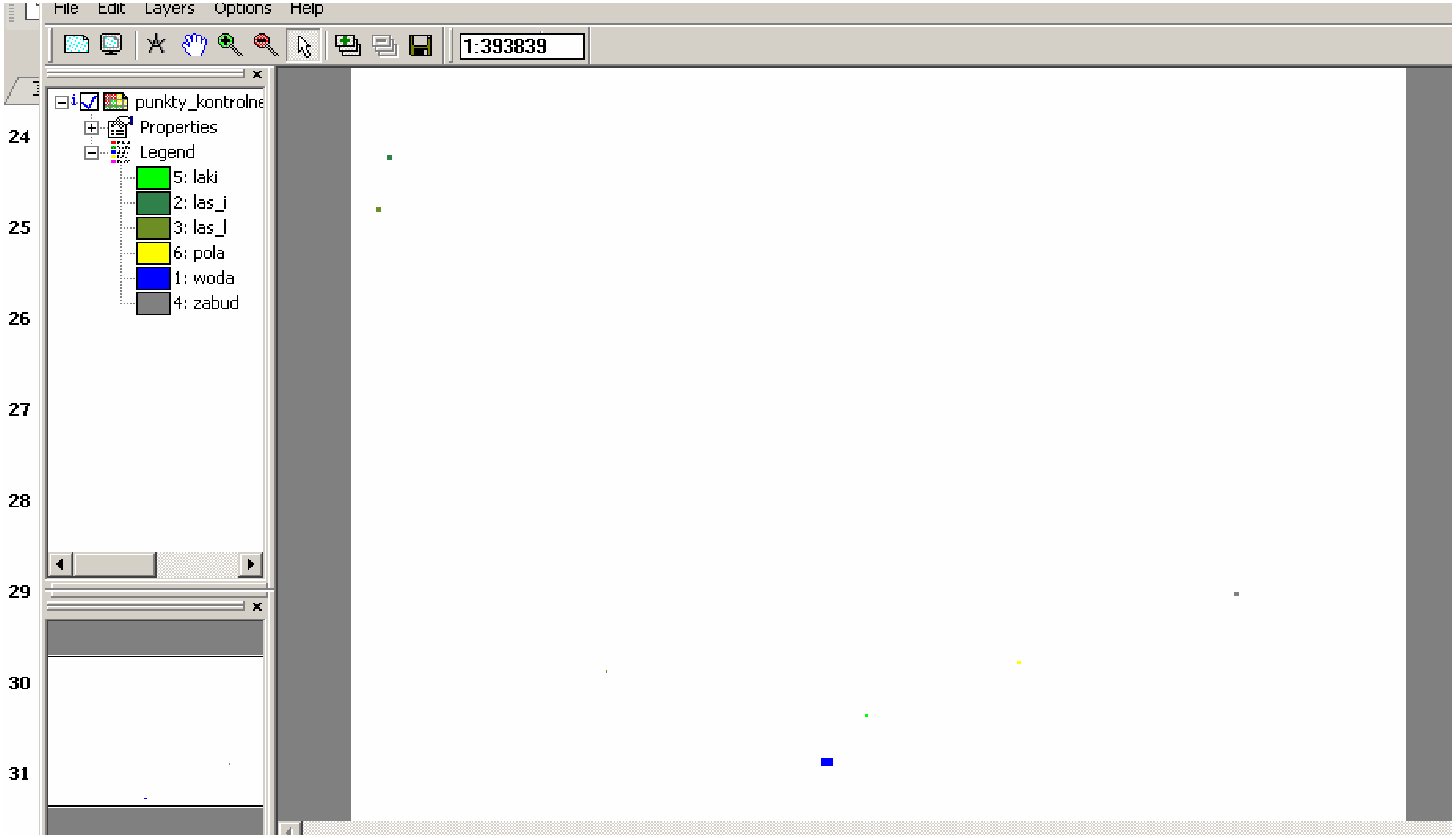
Maximum Likelihood



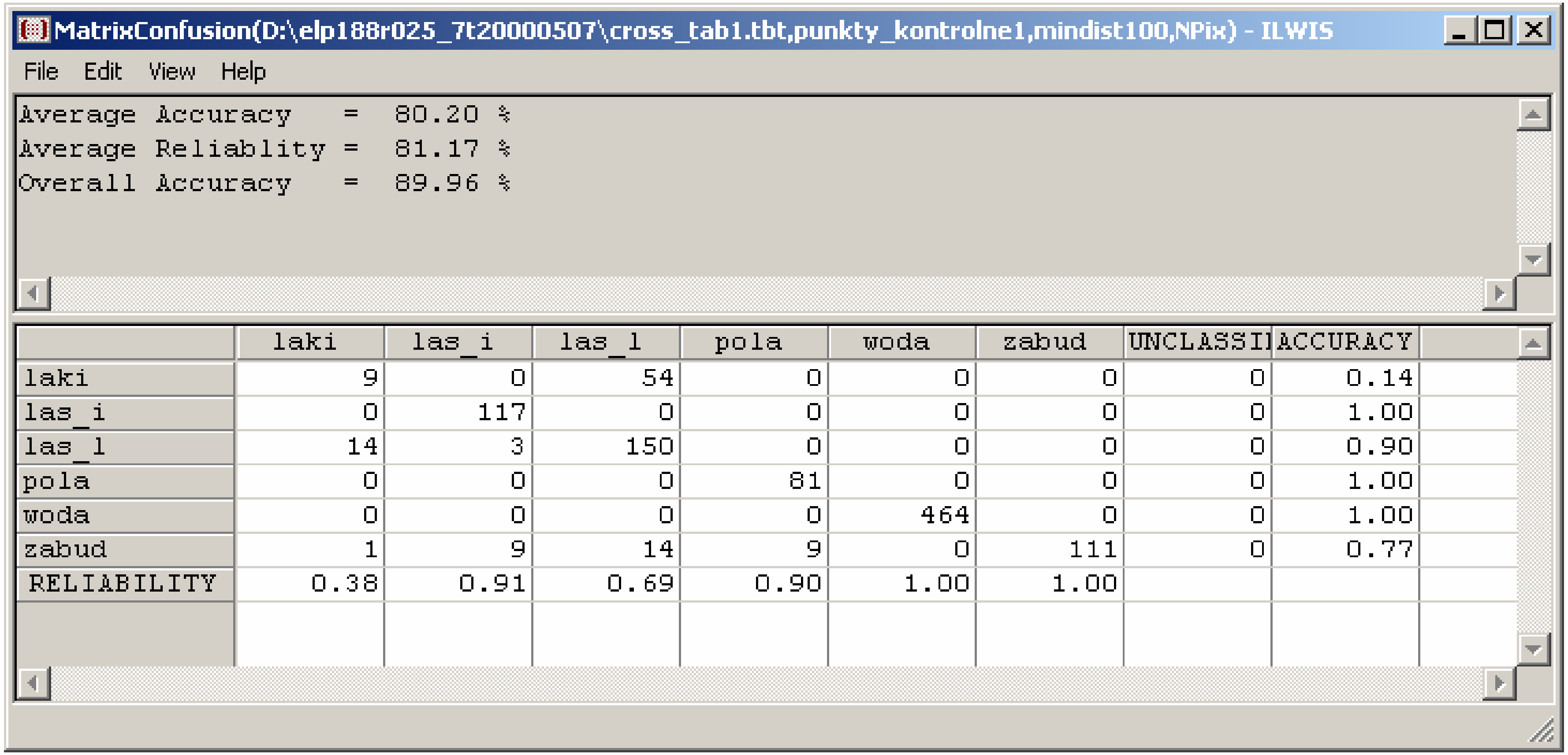
maxlike95: MapClassify(tr_file.mpr,ClassifierMaxLikelihood(95.000000)) - ILWIS

Accuracy analysis

- Control fields



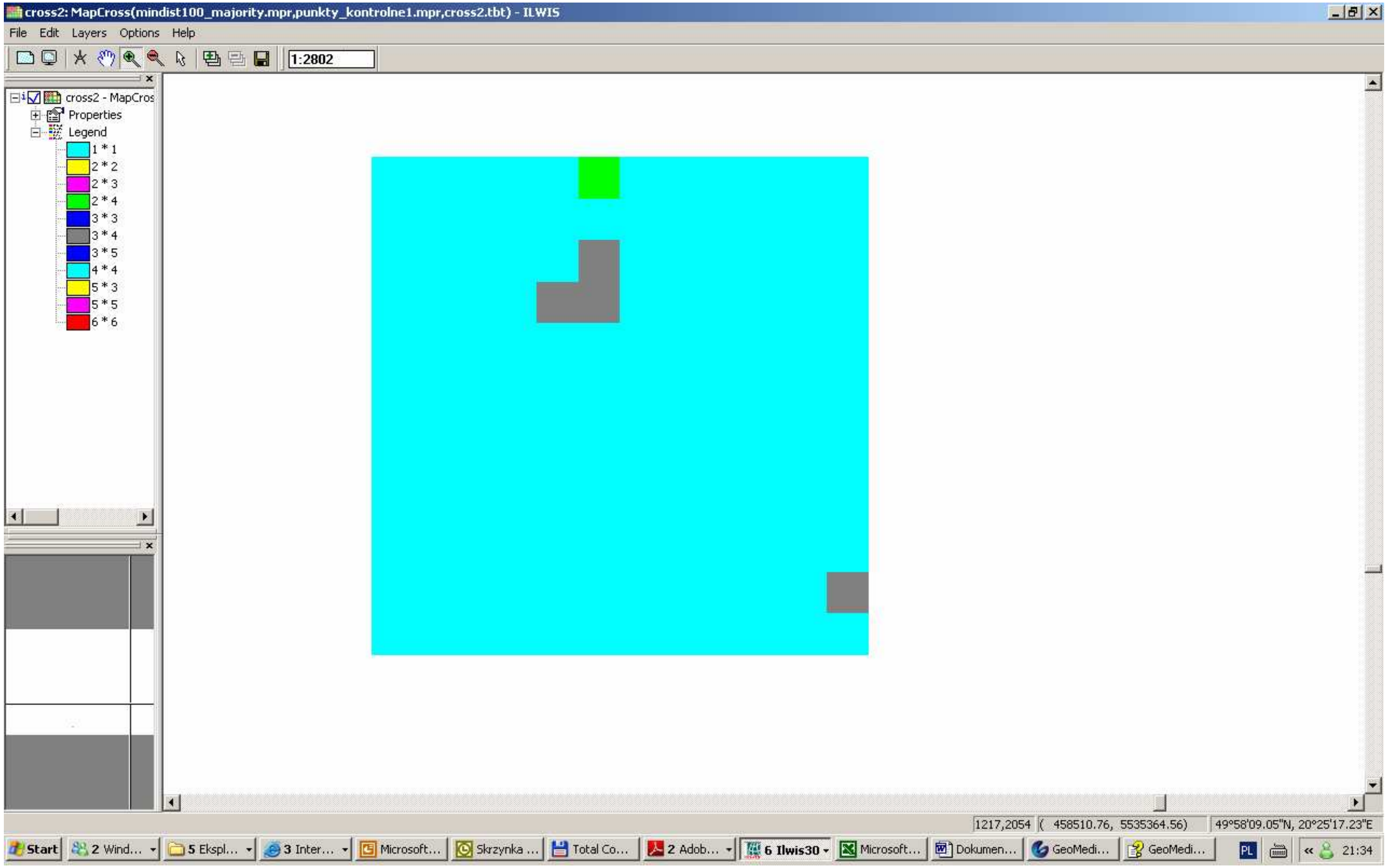
Accuracy analysis – cross matrix


MatrixConfusion(D:\elp188r025_7t20000507\cross_tab1.tbt,punkty_kontrolne1,mindist100,NPix) - ILWIS
File
Edit
View
Help

```
Average Accuracy    =  80.20 %
Average Reliablity =  81.17 %
Overall Accuracy    =  89.96 %
```

	laki	las_i	las_l	pola	woda	zabud	UNCLASSII	ACCURACY
laki	9	0	54	0	0	0	0	0.14
las_i	0	117	0	0	0	0	0	1.00
las_l	14	3	150	0	0	0	0	0.90
pola	0	0	0	81	0	0	0	1.00
woda	0	0	0	0	464	0	0	1.00
zabud	1	9	14	9	0	111	0	0.77
RELIABILITY	0.38	0.91	0.69	0.90	1.00	1.00		

Accuracy analysis



cross2: MapCross(mindist100_majority.mpr,punkty_kontrolne1.mpr,cross2.tbt) - ILWIS

cross2 - MapCros
Properties
Legend

- 🟦 1 * 1
- 🟨 2 * 2
- 🟪 2 * 3
- 🟩 2 * 4
- 🟦 3 * 3
- ▨ 3 * 4
- 🟦 3 * 5
- 🟦 4 * 4
- 🟨 5 * 3
- 🟪 5 * 5
- 🟥 6 * 6

1217,2054 (458510.76, 5535364.56) 49°58'09.05"N, 20°25'17.23"E

Accuracy analysis

$$I_{CP} = \frac{\sum_{p=1}^{k} n_{pp}}{n},$$

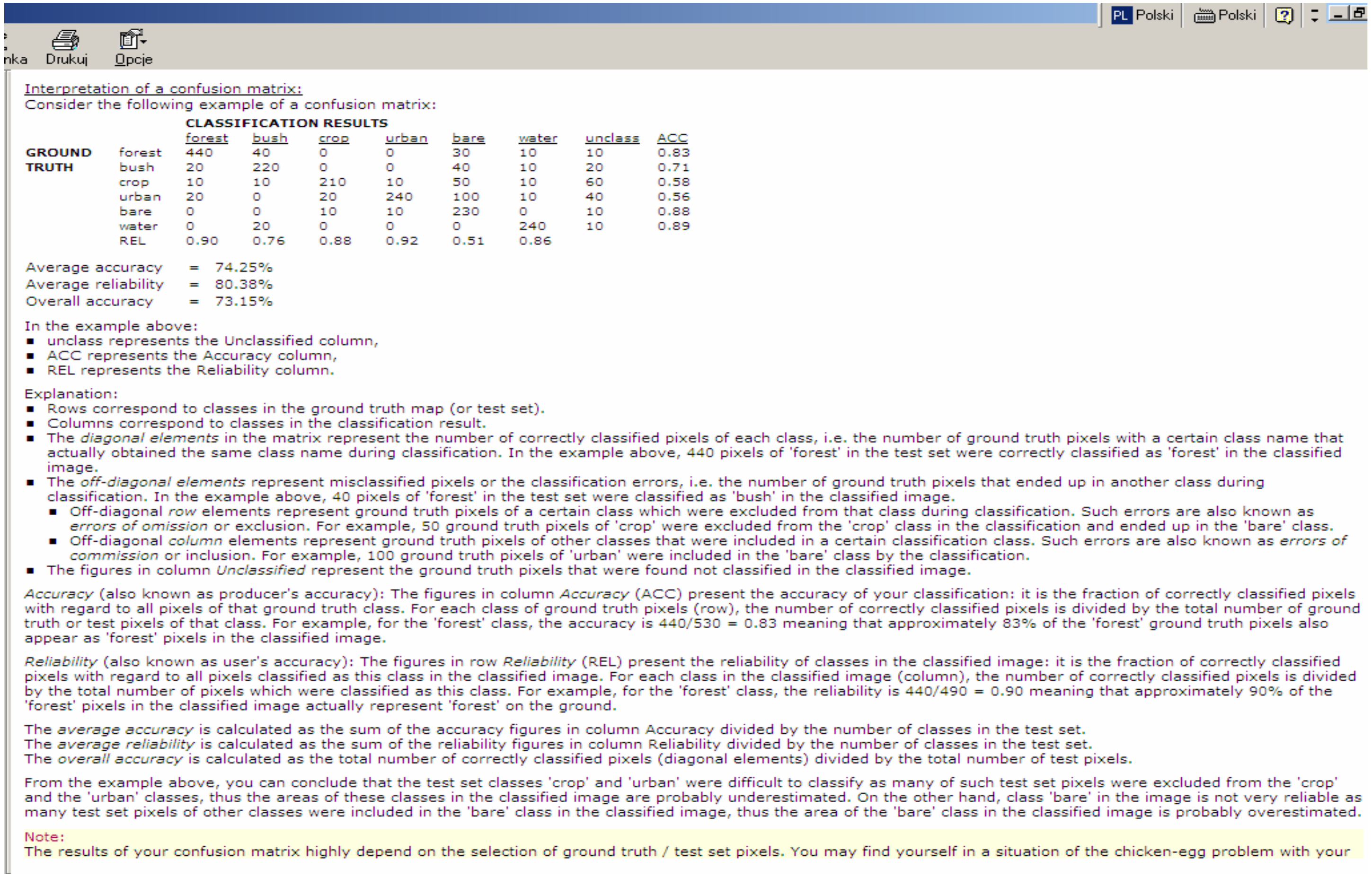
Overall accuracy

$$A_P = \frac{n_{jj}}{\sum_{i=1}^{k} n_{ij}},$$

Producer’s accuracy

$$A_U = \frac{n_{ii}}{\sum_{j=1}^{k} n_{ij}}.$$

User’s accuracy



Interpretation of a confusion matrix:
Consider the following example of a confusion matrix:

		CLASSIFICATION RESULTS							
		forest	bush	crop	urban	bare	water	unclass	ACC
GROUND	forest	440	40	0	0	30	10	10	0.83
TRUTH	bush	20	220	0	0	40	10	20	0.71
	crop	10	10	210	10	50	10	60	0.58
	urban	20	0	20	240	100	10	40	0.56
	bare	0	0	10	10	230	0	10	0.88
	water	0	20	0	0	0	240	10	0.89
	REL	0.90	0.76	0.88	0.92	0.51	0.86		

Average accuracy = 74.25%
Average reliability = 80.38%
Overall accuracy = 73.15%

In the example above:
- unclass represents the Unclassified column,
- ACC represents the Accuracy column,
- REL represents the Reliability column.

Explanation:
- Rows correspond to classes in the ground truth map (or test set).
- Columns correspond to classes in the classification result.
- The *diagonal elements* in the matrix represent the number of correctly classified pixels of each class, i.e. the number of ground truth pixels with a certain class name that actually obtained the same class name during classification. In the example above, 440 pixels of 'forest' in the test set were correctly classified as 'forest' in the classified image.
- The *off-diagonal elements* represent misclassified pixels or the classification errors, i.e. the number of ground truth pixels that ended up in another class during classification. In the example above, 40 pixels of 'forest' in the test set were classified as 'bush' in the classified image.
  - Off-diagonal *row* elements represent ground truth pixels of a certain class which were excluded from that class during classification. Such errors are also known as *errors of omission* or exclusion. For example, 50 ground truth pixels of 'crop' were excluded from the 'crop' class in the classification and ended up in the 'bare' class.
  - Off-diagonal *column* elements represent ground truth pixels of other classes that were included in a certain classification class. Such errors are also known as *errors of commission* or inclusion. For example, 100 ground truth pixels of 'urban' were included in the 'bare' class by the classification.
- The figures in column *Unclassified* represent the ground truth pixels that were found not classified in the classified image.

Accuracy (also known as producer's accuracy): The figures in column *Accuracy* (ACC) present the accuracy of your classification: it is the fraction of correctly classified pixels with regard to all pixels of that ground truth class. For each class of ground truth pixels (row), the number of correctly classified pixels is divided by the total number of ground truth or test pixels of that class. For example, for the 'forest' class, the accuracy is 440/530 = 0.83 meaning that approximately 83% of the 'forest' ground truth pixels also appear as 'forest' pixels in the classified image.

Reliability (also known as user's accuracy): The figures in row *Reliability* (REL) present the reliability of classes in the classified image: it is the fraction of correctly classified pixels with regard to all pixels classified as this class in the classified image. For each class in the classified image (column), the number of correctly classified pixels is divided by the total number of pixels which were classified as this class. For example, for the 'forest' class, the reliability is 440/490 = 0.90 meaning that approximately 90% of the 'forest' pixels in the classified image actually represent 'forest' on the ground.

The *average accuracy* is calculated as the sum of the accuracy figures in column Accuracy divided by the number of classes in the test set.
The *average reliability* is calculated as the sum of the reliability figures in column Reliability divided by the number of classes in the test set.
The *overall accuracy* is calculated as the total number of correctly classified pixels (diagonal elements) divided by the total number of test pixels.

From the example above, you can conclude that the test set classes 'crop' and 'urban' were difficult to classify as many of such test set pixels were excluded from the 'crop' and the 'urban' classes, thus the areas of these classes in the classified image are probably underestimated. On the other hand, class 'bare' in the image is not very reliable as many test set pixels of other classes were included in the 'bare' class in the classified image, thus the area of the 'bare' class in the classified image is probably overestimated.

Note:
The results of your confusion matrix highly depend on the selection of ground truth / test set pixels. You may find yourself in a situation of the chicken-egg problem with your