

---

INF 4300  
30.10.13

Multivariate classification  
Anne Solberg ([anne@ifi.uio.no](mailto:anne@ifi.uio.no))

Based on Chapter 2 (2.1-2.6) in Duda and Hart:  
Pattern Classification

30.10.13

INF 4300

1

---

## Bayes rule for a classification problem

---

- Suppose we have  $J, j=1, \dots, J$  classes.  $\omega$  is the class label for a pixel, and  $\mathbf{x}$  is the observed feature vector).
- We can use Bayes rule to find an expression for the class with the highest probability:

$$P(\omega_j | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_j)P(\omega_j)}{p(\mathbf{x})}$$

$$\text{posterior probability} = \frac{\text{likelihood} \times \text{prior probability}}{\text{normalizing factor}}$$

- $P(\omega_j)$  is the prior probability for class  $\omega_j$ . If we don't have special knowledge that one of the classes occur more frequent than other classes, we set them equal for all classes. ( $P(\omega_j)=1/J, j=1, \dots, J$ ).

30.10.13

INF 4300

2

# Bayes rule explained

$$P(\omega_j | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_j)P(\omega_j)}{p(\mathbf{x})}$$

- $p(\mathbf{x} | \omega_j)$  is the probability density function that models the likelihood for observing feature vector  $\mathbf{x}$  if the pixel belongs to class  $\omega_j$ .
  - Typically we assume a type of distribution, e.g. Gaussian, and the mean and covariance of that distribution is fitted to some data that we know belong to that class. This fitting is called classifier training.
- $P(\omega_j | \mathbf{x})$  is the posterior probability that the pixel actually belongs to class  $\omega_j$ .
- $p(\mathbf{x})$  is just a scaling factor that assures that the probabilities sum to 1.

30.10.13

INF 4300

3

## The conditional density $p(\mathbf{x} | \omega_s)$

- Any probability density function can be used to model  $p(\mathbf{x} | \omega_s)$
- A common model is the multivariate Gaussian density.
- The multivariate Gaussian density:

$$p(\mathbf{x} | \omega_s) = \frac{1}{(2\pi)^{n/2} |\Sigma_s|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_s)^T \Sigma_s^{-1} (\mathbf{x} - \boldsymbol{\mu}_s) \right]$$

- If we have  $n$  features,  $\boldsymbol{\mu}_s$  is a vector of length  $n$  and  $\Sigma_s$  a  $n \times n$  matrix (depends on class  $s$ )

$$\boldsymbol{\mu}_s = \begin{bmatrix} \mu_{1s} \\ \mu_{2s} \\ \vdots \\ \mu_{ns} \end{bmatrix}$$

$$\Sigma_s = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdot & \cdot & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdot & \cdot & \cdot \\ \sigma_{31} & \sigma_{32} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \sigma_{nn-1} & \sigma_{nn} \end{bmatrix}$$

Symmetric  $n \times n$  matrix  
 $\sigma_{ii}$  is the variance of feature  $i$   
 $\sigma_{ij}$  is the covariance between feature  $i$  and feature  $j$

- $|\Sigma_s|$  is the determinant of the matrix  $\Sigma_s$ , and  $\Sigma_s^{-1}$  is the inverse

30.10.13

INF 4300

4

# Mean vectors and covariance matrices in n dimensions

- If  $\mathbf{x}$  is a n-dimensional feature vector for one object/pixel, we can formulate its mean vector and covariance matrix as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad \boldsymbol{\mu} = E[\mathbf{x}] = \begin{bmatrix} E(x_1) \\ E(x_2) \\ \cdot \\ \cdot \\ E(x_n) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \cdot \\ \cdot \\ \mu_n \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdot & \cdot & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdot & \cdot & \sigma_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \cdot & \sigma_{nn} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdot & \cdot & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdot & \cdot & \sigma_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \cdot & \sigma_n^2 \end{bmatrix}$$

- with n features, the mean vector  $\boldsymbol{\mu}$  will be of size  $1 \times n$  and  $\boldsymbol{\Sigma}$  of size  $n \times n$ .

30.10.13

INF 4300

5

## Inspecting $p(\mathbf{x} | \omega_s)$

$$p(\mathbf{x} | \omega_s) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_s|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_s)^T \boldsymbol{\Sigma}_s^{-1} (\mathbf{x} - \boldsymbol{\mu}_s) \right]$$

↑ Scalar  
↑  $1 \times n$  vector transposed  
↑  $n \times n$  matrix  
↑ Inverse of covariance matrix  
↑  $n \times 1$  vector transposed

30.10.13

INF 4300

6

## The mean vectors $\mu_s$ for each class

- The mean vector for class  $s$  is defined as the expected value of  $\mathbf{x}$ :

$$\mu_s = E[\mathbf{x}] = \begin{bmatrix} E(x_1) \\ E(x_2) \\ \cdot \\ \cdot \\ E(x_n) \end{bmatrix} = \begin{bmatrix} \mu_1^s \\ \mu_2^s \\ \cdot \\ \cdot \\ \mu_n^s \end{bmatrix}$$

- with  $n$  features, the mean vector  $\mu$  will be of size  $1 \times n$ .
- If we have  $M_s$  training samples that we know belong to class  $s$ , we can estimate the mean vector as:

$$\hat{\mu}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} \mathbf{x}_m,$$

where the sum is over all training samples belonging to class  $s$

30.10.13

INF 4300

7

## The covariance matrix $\Sigma_s$ for each class

- The covariance for class  $s$  is defined as the expected value of  $(\mathbf{x}-\mu)(\mathbf{x}-\mu)^t$ :

$$\Sigma_s = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdot & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdot & \sigma_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \sigma_{nn} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdot & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdot & \sigma_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \sigma_{n1} & \sigma_{n2} & \cdot & \sigma_n^2 \end{bmatrix}$$

- with  $n$  features, the covariance matrix  $\Sigma_s$  will be of size  $n \times n$ .
- If we have  $M_s$  training samples that we know belong to class  $s$ , we can estimate the covariance matrix  $\Sigma_s$  as:

$$\hat{\Sigma}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} (x_m - \hat{\mu}_s)(x_m - \hat{\mu}_s)^t$$

where the sum is over all training samples belonging to class  $s$

- Each term  $\sigma_{ij}$  is computed as:

$$\sigma_{ij,s} = \frac{1}{M_s} \sum_{m=1}^{M_s} (x_{m,i} - \hat{\mu}_s)(x_{m,j} - \hat{\mu}_s)$$

for the covariance between feature  $i$  and  $j$  for class  $s$

30.10.13

INF 4300

8

## More on the covariance matrix $\Sigma_s$

- The covariance matrix  $\Sigma_s$  will always be symmetric and positive semidefinite.
- If all components of  $x$  have non-zero variance,  $\Sigma_s$  will be positive definite.
- $\sigma_{ij}$  is the covariance between features  $i$  and  $j$ .
- If features  $x_i$  and  $x_j$  are uncorrelated,  $\sigma_{ij} = 0$ .
- In the general case,  $\Sigma_s$  will have  $n(n+1)/2$  different values.

30.10.13

INF 4300

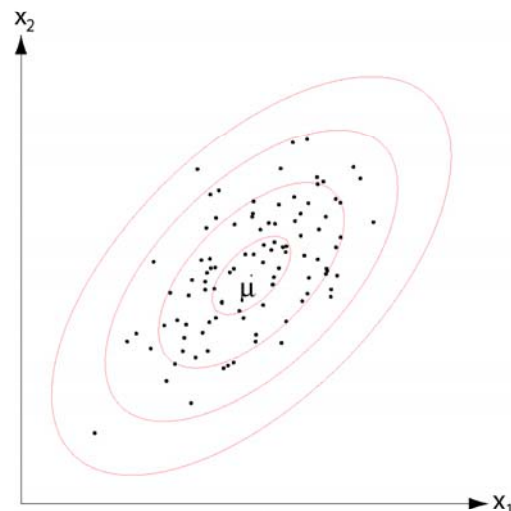
9

## A 2D Gaussian model

- Parameters  $\mu$  and  $\Sigma$  define a density as a "bump"
- The curves on the plot are contours of equal probability, just as the contours on a map
- The matrix  $\Sigma$  in this case has three different elements, variance in each of the axes, and covariance between the axes

$$\Sigma_s = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 \end{bmatrix}$$

- $\sigma_{11}^2$  is the variance for feature 1
- $\sigma_{12} = \sigma_{21}$  is the covariance between feature 1 and 2
- $\sigma_{22}^2$  is the variance for feature 2



30.10.13

INF 4300

10

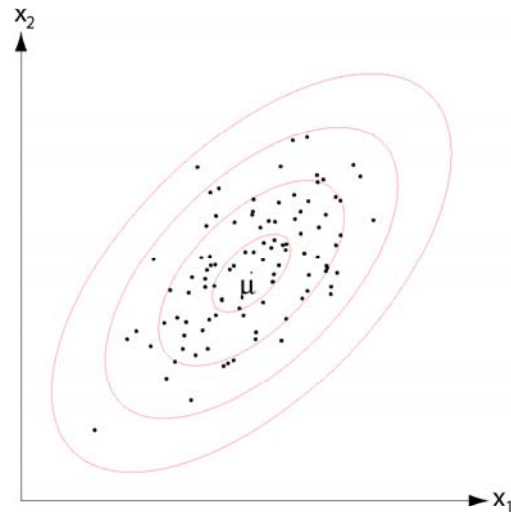
# The covariance matrix and ellipses

- In 2D, the Gaussian model can be thought of as approximating the classes in 2D feature space with ellipses.
- The mean vector  $\mu = [\mu_1, \mu_2]$  defines the center point of the ellipses.
- $\sigma_{12}$ , the covariance between the features defines the orientation of the ellipse.
- $\sigma_{11}$  and  $\sigma_{22}$  defines the width of the ellipse (the a and b parameters).

$$\Sigma_s = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

- The ellipse defines points where the probability density is equal
  - Equal in the sense that the distance to the mean as computed by the Mahalanobis distance is equal.
  - The Mahalanobis distance between a point  $x$  and the class center  $\mu$  is:

$$r^2 = (x - \mu)^T \Sigma^{-1} (x - \mu)$$

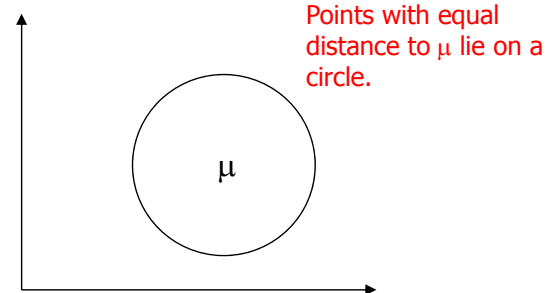


The main axes of the ellipse is determined by the eigenvectors of  $\Sigma$ . The eigenvalues of  $\Sigma$  gives their length.

## Euclidean distance vs. Mahalanobis distance

- Euclidean distance between point  $x$  and class center  $\mu$ :

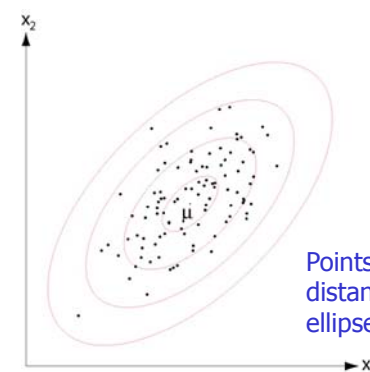
$$(x - \mu)^T (x - \mu) = \|x - \mu\|^2$$



Points with equal distance to  $\mu$  lie on a circle.

- Mahalanobis distance between  $x$  and  $\mu$ :

$$r^2 = (x - \mu)^T \Sigma^{-1} (x - \mu)$$



Points with equal distance to  $\mu$  lie on an ellipse.

# Discriminant functions for the normal density

---

- We saw last lecture that the minimum-error-rate classification can be computed using the discriminant functions

$$g_i(\mathbf{x}) = \ln p(\mathbf{x} | \omega_i) + \ln P(\omega_i)$$

- With a multivariate Gaussian we get:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

- Let us look at this expression for some special cases:

## Case 1: $\boldsymbol{\Sigma}_j = \sigma^2 \mathbf{I}$

---

- Consider that we assume the features uncorrelated (independent) with the same variance  $\sigma^2$
- The covariances  $\sigma_{ij} = 0$  (by definition if the features are uncorrelated).
- The discriminant functions will be simplified to:

$$g_i(\mathbf{x}) = -\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|^2}{2\sigma^2} + \ln P(\omega_i)$$

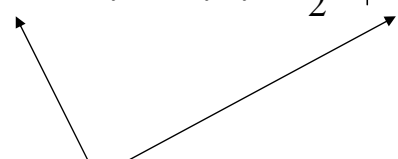
$$\text{where } \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = (\mathbf{x} - \boldsymbol{\mu}_i)^t (\mathbf{x} - \boldsymbol{\mu}_i)$$

- We show this on the next slide
- Thus we model the probabilities as n-dimensional *spheres* because points that have equal discriminant function will lie on a circle around the mean  $\boldsymbol{\mu}_i$ .
- $\boldsymbol{\Sigma}_j^{-1} = \mathbf{I}/\sigma^2$
- $|\boldsymbol{\Sigma}_j| = \sigma^{2n}$

# Case 1: $\Sigma_j = \sigma^2 I$

- The discriminant functions simplifies to **linear** functions using such a shape on the probability distributions

$$g_j(\mathbf{x}) = -\frac{1}{2(\sigma^2 I)} (\mathbf{x} - \boldsymbol{\mu}_j)^T (\mathbf{x} - \boldsymbol{\mu}_j) - \frac{1}{2} \ln |\sigma^2 I| + \ln P(\omega_j)$$

$$= -\frac{1}{2(\sigma^2 I)} (\mathbf{x}^T \mathbf{x} - 2\boldsymbol{\mu}_j^T \mathbf{x} + \boldsymbol{\mu}_j^T \boldsymbol{\mu}_j) - \frac{1}{2} \ln |\sigma^2 I| + \ln P(\omega_j)$$


Common for all classes, no need to compute these two terms  
 Since  $\mathbf{x}^T \mathbf{x}$  is common for all classes,  $g_j(\mathbf{x})$  reduces to a linear function of  $\mathbf{x}$ .

# Case 1: $\Sigma_j = \sigma^2 I$

- An equivalent formulation of the discriminant functions:

$$g_i(\mathbf{x}) = \mathbf{w}'_i \mathbf{x} + w_{i0}$$

$$\text{where } \mathbf{w}_i = \frac{1}{\sigma^2} \boldsymbol{\mu}_i \text{ and } w_{i0} = -\frac{1}{2\sigma^2} \boldsymbol{\mu}'_i \boldsymbol{\mu}_i + \ln P(\omega_i)$$

- The equation  $g_i(\mathbf{x}) = g_j(\mathbf{x})$  can be written as

$$\mathbf{w}' (\mathbf{x} - \mathbf{x}_0) = 0$$

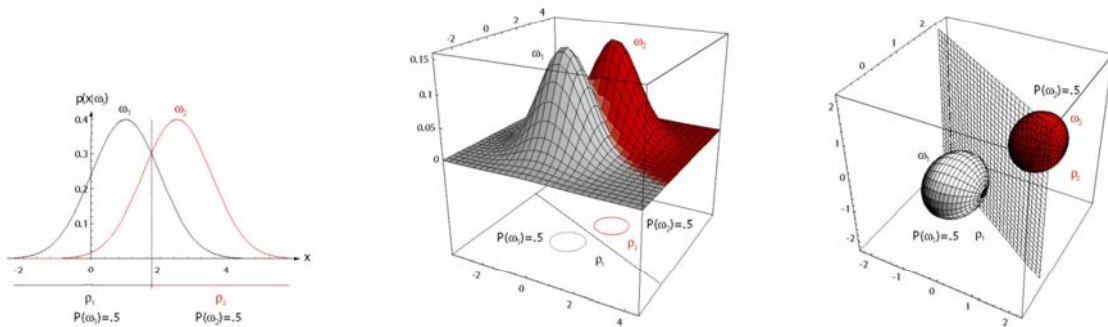
$$\text{where } \mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$$

$$\text{and } \mathbf{x}_0 = \frac{1}{2} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) - \frac{\sigma^2}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} \ln \frac{P(\omega_i)}{P(\omega_j)} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

- $\mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$  is the vector between the mean values.
- This equation defines a hyperplane through the point  $\mathbf{x}_0$ , and orthogonal to  $\mathbf{w}$ .
- If  $P(\omega_i) = P(\omega_j)$  the hyperplane will be located halfway between the mean values.



# A simple model, $\Sigma_j = \sigma^2 I$



- The distributions are spherical in  $n$  dimensions.
- The decision boundary is a generalized hyperplane of  $n-1$  dimensions
- The decision boundary is perpendicular to the line separating the two mean values
- This kind of a classifier is called a linear classifier, or a linear discriminant function
  - Because the decision function is a linear function of  $\mathbf{x}$ .
- If  $P(\omega_i) = P(\omega_j)$ , the decision boundary will be half-way between  $\mu_i$  and  $\mu_j$

30.10.13

INF 4300

17

## Minimum distance classification

- If all classes have equal prior probabilities,  $x_0$  will be the point halfway between the mean vectors.
- Classification will consist of assigning feature vector  $x$  to the same class as the closest mean measured by Euclidean distance  $\|x - \mu_i\|$ .
- A classifier based on the Euclidean distance is called a **minimum distance classifier**.

30.10.13

INF 4300

18

## Case 2: Common covariance, $\Sigma_j = \Sigma$

---

- If we assume that all classes have the same shape of data clusters, an intuitive model is to assume that their probability distributions have the same shape
- By this assumption we can use all the data to estimate the covariance matrix
- This estimate is common for all classes, and this means that also in this case the discriminant functions become linear functions

$$\begin{aligned}g_j(\mathbf{x}) &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_j) - \frac{1}{2} \ln|\boldsymbol{\Sigma}| + \ln P(\omega_j) \\ &= -\frac{1}{2(\sigma^2 I)} (\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j) - \frac{1}{2} \ln|\boldsymbol{\Sigma}| + \ln P(\omega_j)\end{aligned}$$

Common for all classes, no need to compute  
Since  $\mathbf{x}^T \mathbf{x}$  is common for all classes,  $g_j(\mathbf{x})$  again reduces to a linear function of  $\mathbf{x}$ .

30.10.13

INF 4300

19

## Case 2: Common covariance, $\Sigma_j = \Sigma$

---

- An equivalent formulation of the discriminant functions is

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

$$\text{where } \mathbf{w}_i = \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i$$

$$\text{and } w_{i0} = -\frac{1}{2} \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i + \ln P(\omega_i)$$

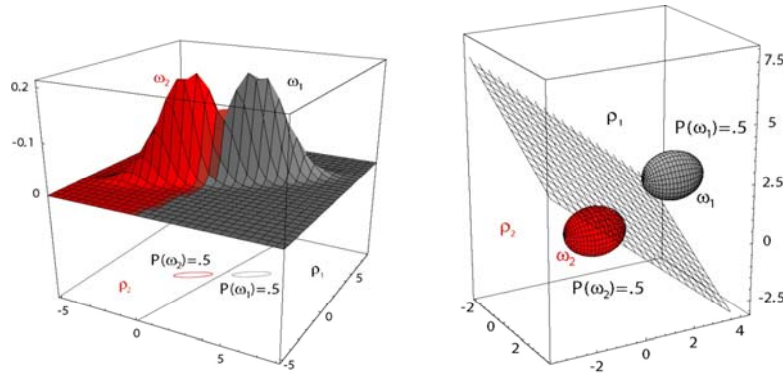
- The decision boundaries are again hyperplanes.
- Because  $\mathbf{w}_i = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$  is not in the direction of  $(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$ , the hyperplane will not be orthogonal to the line between the means.

30.10.13

INF 4300

20

# Common covariance, $\Sigma_j = \Sigma$



- The classes can be described by hyperellipsoids in  $d$  dimensions.
- All hyperellipsoids have the same orientation.
- The decision boundary will again be a hyperplane.
- Because  $\mathbf{w} = \Sigma^{-1}(\mu_i - \mu_j)$  is generally not in the direction of  $\mu_i - \mu_j$  the hyperplane will not be perpendicular to the line between the means.
- Consider a point  $x_0$  on the line  $\mu_i - \mu_j$ , defined by the prior probabilities:
  - If  $P(\omega_i) = P(\omega_j)$ ,  $x_0$  will be half way between the means.
  - The separating hyperplane will *intersect* the line at  $x_0$

# Case 3:, $\Sigma_j = \text{arbitrary}$

- When all classes are modeled as having different *shapes*, the discriminant functions cannot be simplified
- This means that the discriminant functions will be *quadratic* functions
- Decision boundaries will be hyperquadrics and assume any of the general forms:
  - hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, hyperhyperboloids...

## Case 3:, $\Sigma_j = \text{arbitrary}$

---

- The discriminant functions will be quadratic:

$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} + w_{i_0}$$

$$\text{where } \mathbf{W}_i = -\frac{1}{2} \Sigma_i^{-1}, \quad \mathbf{w}_i = \Sigma_i^{-1} \boldsymbol{\mu}_i$$

$$\text{and } w_{i_0} = -\frac{1}{2} \boldsymbol{\mu}_i^t \Sigma_i^{-1} \boldsymbol{\mu}_i - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

- The decision surfaces are hyperquadrics and can assume any of the general forms:
  - hyperplanes
  - hyperspheres
  - pairs of hyperplanes
  - hyperellipsoids,
  - Hyperparaboloids, ..
- The next slides show examples of this.
- In this general case we cannot intuitively draw the decision boundaries just by looking at the mean and covariance.

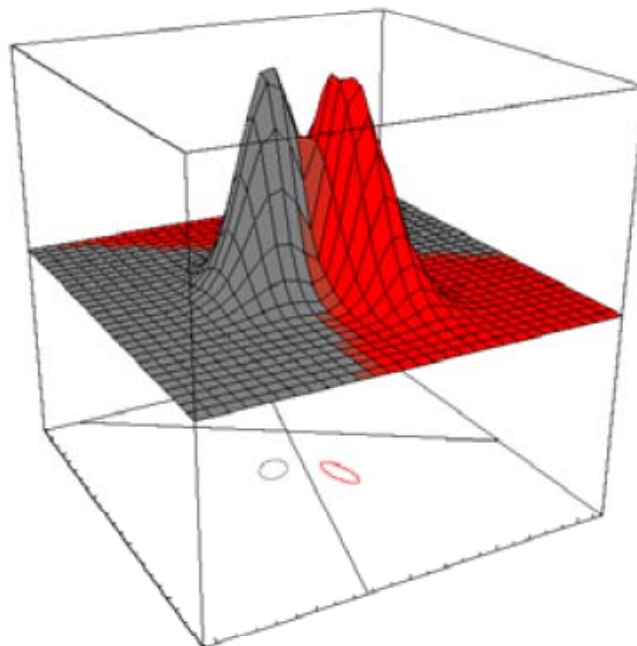
30.10.13

INF 4300

23

## The full model, $\Sigma_j = \text{arbitrary}$ - example

---



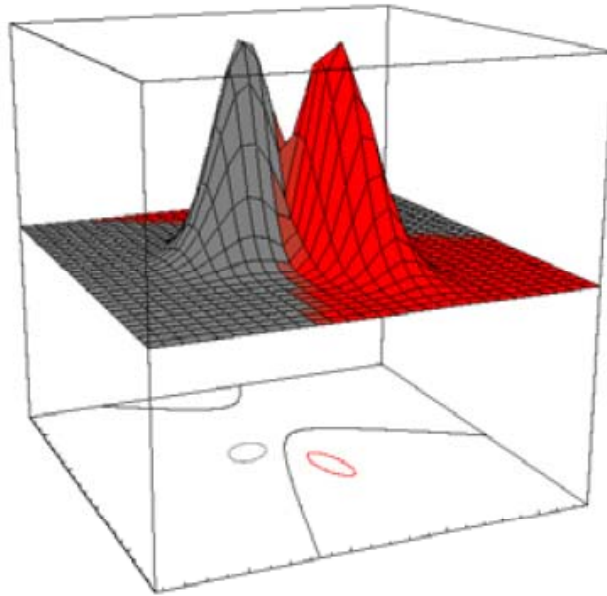
30.10.13

INF 4300

24

## The full model, $\Sigma_j = \text{arbitrary}$ - example

---



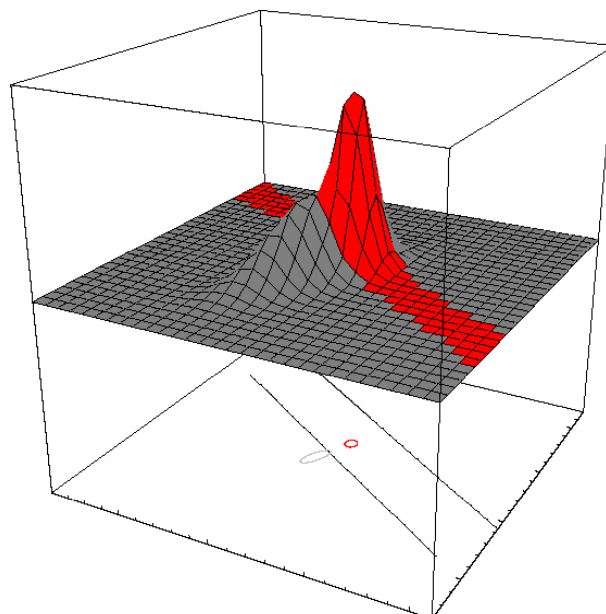
30.10.13

INF 4300

25

## The full model, $\Sigma_j = \text{arbitrary}$ - example

---



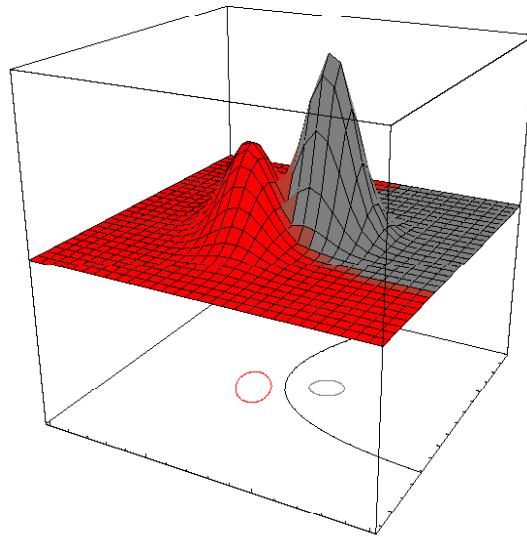
30.10.13

INF 4300

26

## The full model, $\Sigma_j = \text{arbitrary}$ - example

---



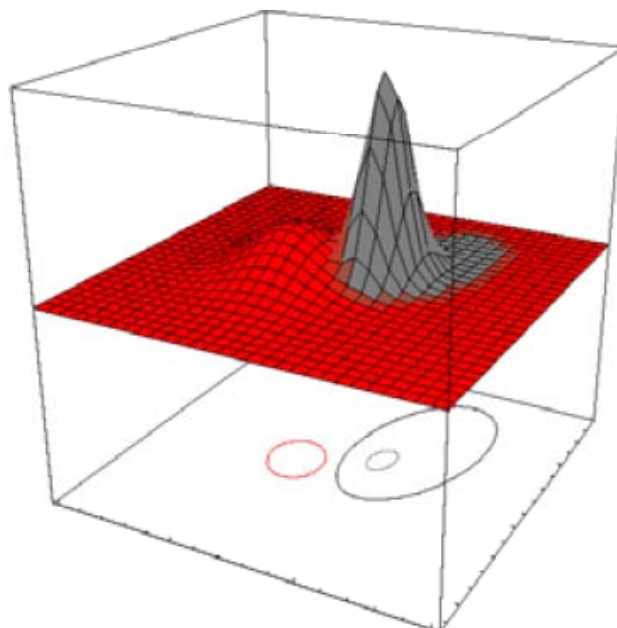
30.10.13

INF 4300

27

## The full model, $\Sigma_j = \text{arbitrary}$ - example

---



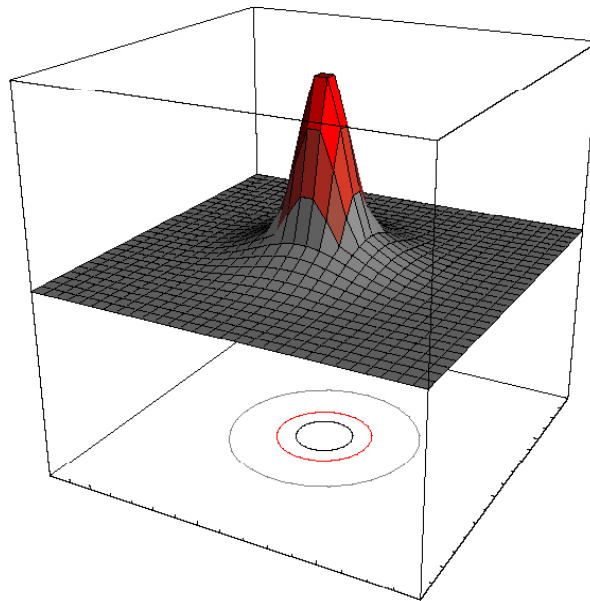
30.10.13

INF 4300

28

# The full model, $\Sigma_j = \text{arbitrary}$ - example

---



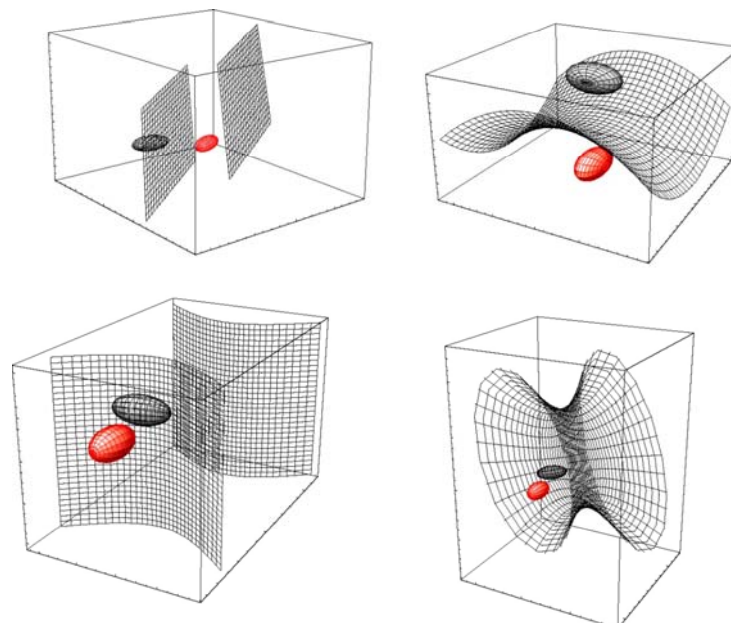
30.10.13

INF 4300

29

# The full model, $\Sigma_j = \text{arbitrary}$ - example

---



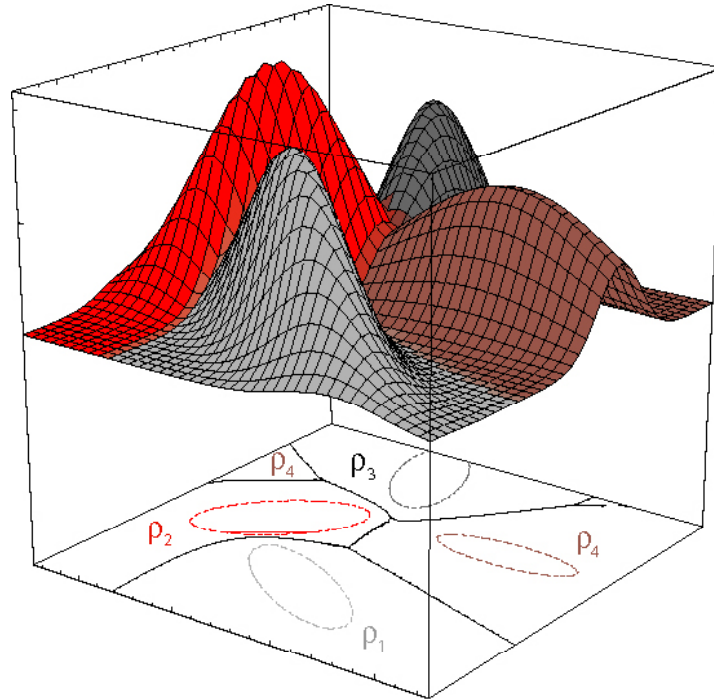
30.10.13

INF 4300

30

# A multiclass example

---



30.10.13

INF 4300

31

## Is the Gaussian classifier the only choice?

---

- The Gaussian classifier gives linear or quadratic discriminant function.
- Other classifiers can give arbitrary complex decision surfaces (often piecewise-linear)
  - Mixtures of Gaussians
  - Neural networks
  - Support vector machines
  - kNN (k-Nearest-Neighbor) classification (next week)

30.10.13

INF 4300

32



# Learning goals from this lecture

---

- Be able to use and implement Bayes rule with a n-dimensional Gaussian distribution.
- Know how  $\mu_s$  and  $\Sigma_s$  are estimated.
- Understand the 2-dimensional case where a covariance matrix is illustrated as an ellipse.
- Be able to simplify the general discriminant function for 3 cases.
- Have a geometric interpretation of classification with 2 features.

## If we have time...

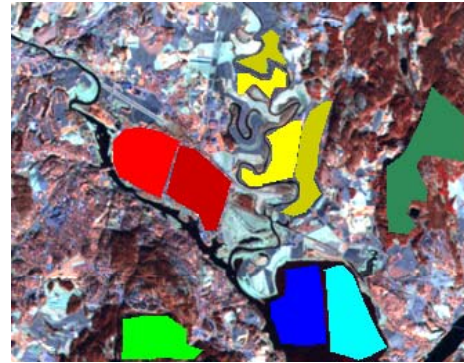
---

- If time allows, we include the following slides in this lecture

# Using masks to train and test

---

- Training mask: a mask where regions to train each class are marked using different pixel values, e.g. class label=1 for class 1, 2 for class 2 etc.
- Test mask: a similar mask as training, but to estimate classifier accuracy only.



## Training a classifier

---

- Obtain as many ground truth samples for each class as possible
  - If visual inspection is reliable, experts can mark training regions interactively.
  - For remote sensing, go out in the field and collect field samples (or use images from a different sensor)
  - For symbol recognition, mark a set of symbols manually.
  - For medical applications, use e.g. tissue samples or interpretations made by experts.
- Divide the ground truth into a training set and a test set.
- Use feature extraction and feature selection/evaluation to determine the best set of features.
- Decide if a linear or quadratic classifier is needed.
  - $\hat{\mu}_s$  has  $n$  elements
  - $\hat{\Sigma}_s$  has  $n(n-1)/2$  elements
- For each class, compute  $\mu_s$  (and  $\Sigma_s$ ) using the given Maximum Likelihood estimates.

# Classifying new data

---

- For each sample, compute the posterior probabilities for each class.

$$P(\omega_s | x) \propto p(x | \omega_s) P(\omega_s)$$
$$= \left[ \frac{1}{(2\pi)^{P/2} |\Sigma_s|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu_s)^t \Sigma_s^{-1} (x - \mu_s) \right] \right] P(\omega_s)$$

- Classify the sample to the class with the highest posterior probability.
- Evaluate the performance of the classifier.
- We can also produce images of the posterior probability for each class.

# Validating classifier performance

---

- Classification performance is evaluated on a different set of samples with known class - the **test set**.
- The training set and the test set must be independent!
- Normally, the set of ground truth pixels (with known class) is partitioned into a set of training pixels and a set of test pixels of approximately the same size.
- This can be repeated several times to compute more robust estimates as average test accuracy over several different partitions of test set and training set.
  - By selecting e.g. 10 random partitions of the set of samples into a training set and a test set.

# Confusion matrices

- A matrix with the true class label versus the estimated class labels for each class

## Estimated class labels

True class labels

	Class 1	Class 2	Class 3	Total #samples
Class 1	80	15	5	100
Class 2	5	140	5	150
Class 3	25	50	125	200
Total	110	205	135	450

30.10.13

INF 4300

39

## Confusion matrix - cont.

Alternatives:

- Report no. correctly classified pixels for each class.
- Report the percentage of correctly classified pixels for each class.
- Report the percentage of correctly classified pixels in total.
  - Why is this not a good measure if the number of test pixels from each class varies between classes?

	Class 1	Class 2	Class 3	Total #samples
Class 1	80	15	5	100
Class 2	5	140	5	150
Class 3	25	50	125	200
Total	110	205	135	450

30.10.13

INF 4300

40

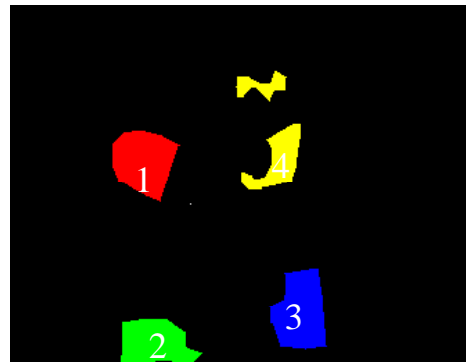
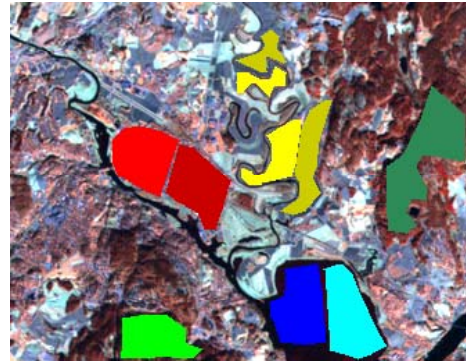
# A classification example

---

Landsat image with 6 spectral bands  
The 6 bands will be the features  
Training areas and test areas shown  
in mask

Upper part: RGB-false color image created from bands  
4,5 and 6 with training and test regions overlaid.

Lower part: image of training regions only



30.10.13

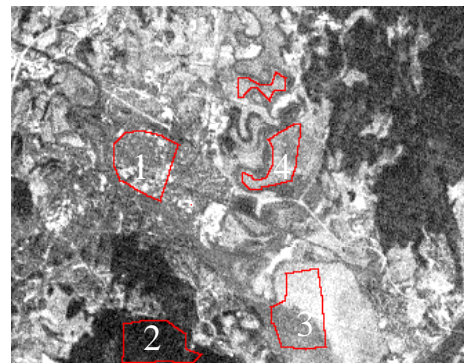
INF 4300

41

## Visual inspection of feature 1

---

Class 2 (forest) seems to be well separated,  
Maybe also class 1 (urban)



30.10.13

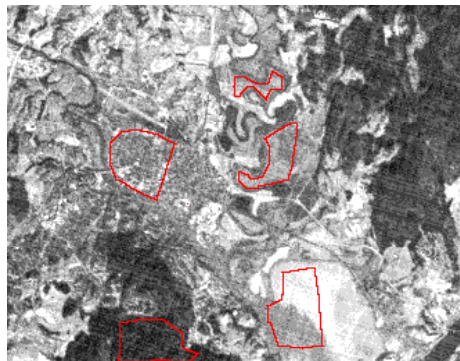
INF 4300

42

## Visual inspection of feature 2

---

Class 2 (forest) seems to be well separated



30.10.13

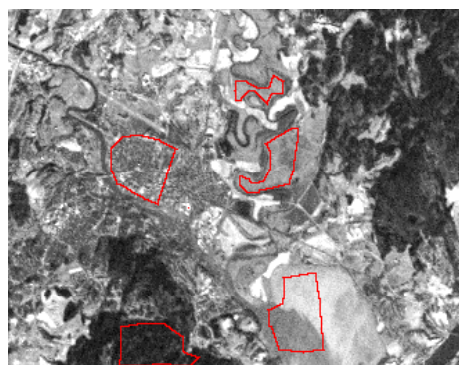
INF 4300

43

## Visual inspection of feature 3

---

Class 2 (forest) seems to be well separated,  
Class 1 (urban) seems to be well separated



30.10.13

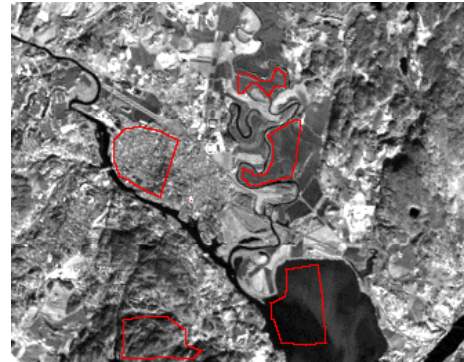
INF 4300

44

## Visual inspection of feature 4

---

Class 1 (water) seems to be well separated,  
Maybe also class 4 (agricultural)



30.10.13

INF 4300

45

## Visual inspection of feature 5

---

Water and forest appears similar  
- but the variance might be  
different

Urban and agricultural appears  
similar – but the variance might  
be different



30.10.13

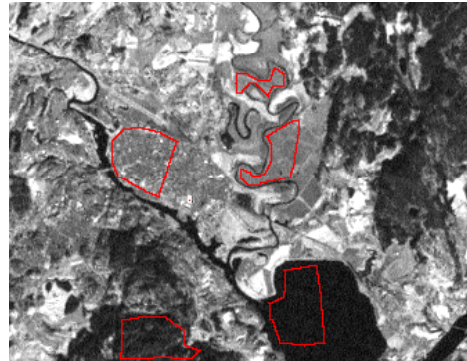
INF 4300

46

## Visual inspection of feature 6

---

Seems similar to feature 5,  
but with better contrast



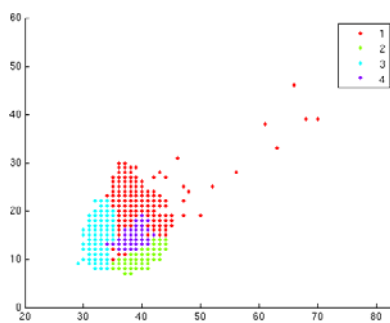
30.10.13

INF 4300

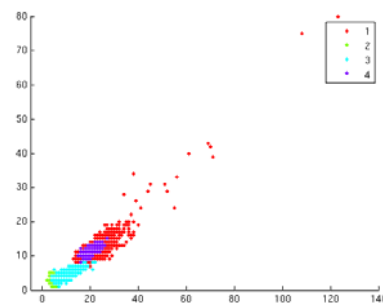
47

## Selected scatter plots (gscatter)

---



Scatterplot between feature 1 and 4



Scatterplot between feature 5 and 6

30.10.13

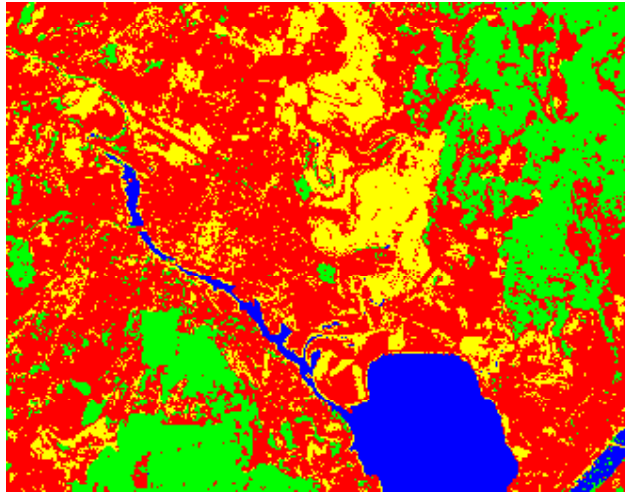
INF 4300

48



# Classified images

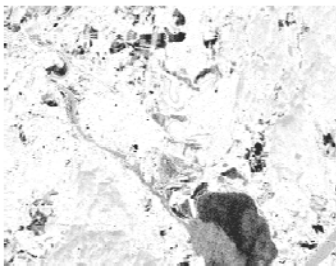
---



The entire image classified to the most probable class

## Display the posterior probabilities as images

---



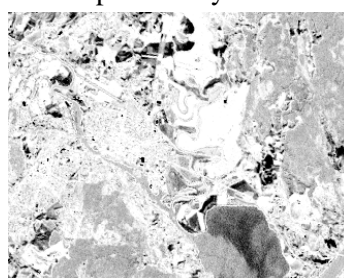
Posterior probability for class urban



Posterior probability for class forest



Posterior probability for class water



Posterior probability for class agricultural

Dark values:  
Probabilities close to 0

Bright values:  
Probabilities close to 1

## Confusion matrix for the training set

---

<b>True class</b>	<b>Assigned to Class1</b>	<b>Assigned to Class2</b>	<b>Assigned to Class 3</b>	<b>Assigned to Class4</b>
Class 1	1340	2	0	310
Class 1	43	1253	0	2
Class 3	0	0	1738	0
Class 4	131	3	0	1266

Accuracy per class: Averaged over all classes: 91.7%

Class1: 81%

Class2: 96%

Class3: 100%

Class4: 90%

30.10.13

INF 4300

51

## Confusion matrix for the test set

---

<b>True class</b>	<b>Assigned to Class1</b>	<b>Assigned to Class2</b>	<b>Assigned to Class 3</b>	<b>Assigned to Class4</b>
Class 1	1474	3	1	251
Class 1	513	2311	0	0
Class 3	14	0	1953	0
Class 4	213	2	0	1390

Accuracy per class: Averaged over all classes: 87.5%

Class1: 85%

Class2: 81%

Class3: 98%

Class4: 86%

30.10.13

INF 4300

52