IN 5520
14.10.20

Multivariate classification

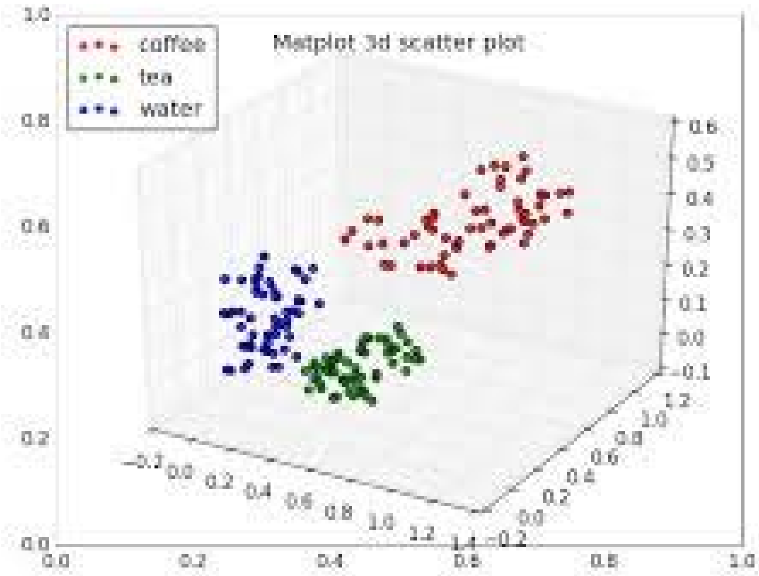
Anne Solberg (anne@ifi.uio.no)

Based on Chapter 2.1-2.4 in Pattern Recognition, Theodoridis and Koutroumbas

Mandatory 2

- Available before next group session
- Implement your own classification algorithm

The goal of supervised classification



Find a partition of multivariate feature space that we believe will work well when classifying new data
A simple model is easier to interpret/explain

Classification in multivariate feature space

- Goal:
  - Learn by concept
  - Learn by mathematics
  - Learn by geometry
  - Learn by implementation

Today’s focus

- Basics of probability theory
- Bayes rule
- From a l-dimensional feature vector $\mathbf{x}=[x_1,....x_s]^T$
- The multivariate Gaussian density
- Discriminant functions for the Gaussian density
- If time: a classification example

Bayes rule for a classification problem

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

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

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

- $P(\omega_j)$ is the prior probability for class ω_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.,,,J).

Probability theory

- Let x be a discrete random variable that can assume any of a finite number of M different values (*in our case M classes*).

- The probability that x belongs to class m is

  $p_m = Pr(x=m), m=1,...M$

- A probability distribution must sum to 1 and probabilities must be positive so $p_m \geq 0$ and $\sum_{m=1}^{M} p_m = 1$

Expected values - definition

- The expected value or mean of a random variable x is:

$$E[x] = \mu = \sum_x xP(x)$$

- The variance or second order moment σ^2 is:

$$E[x^2] = \sum_x x^2 P(x)$$

$$Var[x] = \sigma^2 = E[(x-u)^2] = \sum_x (x-u)^2 P(x)$$

Pairs of random variables - definitions

- Let *x* and *y* be two random variables.
- The joint probability of observing a **pair** of values (x=i,y=j) is $p_{ij.}$
- Alternatively we can define a joint probability distribution function P(x,y) for which

$$P(x,y) \geq 0, \quad \sum_x \sum_y P(x,y) = 1$$

- The marginal distributions for x and y (if we want to eliminate one of them) is:

$$P_x(x) = \sum_y P(x,y)$$

$$P_y(y) = \sum_x P(x,y)$$

Expected values of two variables

- Expected values of two variables:

$$E(f(x,y)) = \sum_x \sum_y f(x,y)P(x,y)$$

$$\mu_x = E(x) = \sum_x \sum_y xP(x,y)$$

$$\mu_y = E(y) = \sum_x \sum_y yP(x,y)$$

$$\sigma_x^2 = E\left[(x-\mu_x)^2\right] = \sum_x \sum_y (x-\mu_x)^2 P(x,y)$$

Variance of feature x

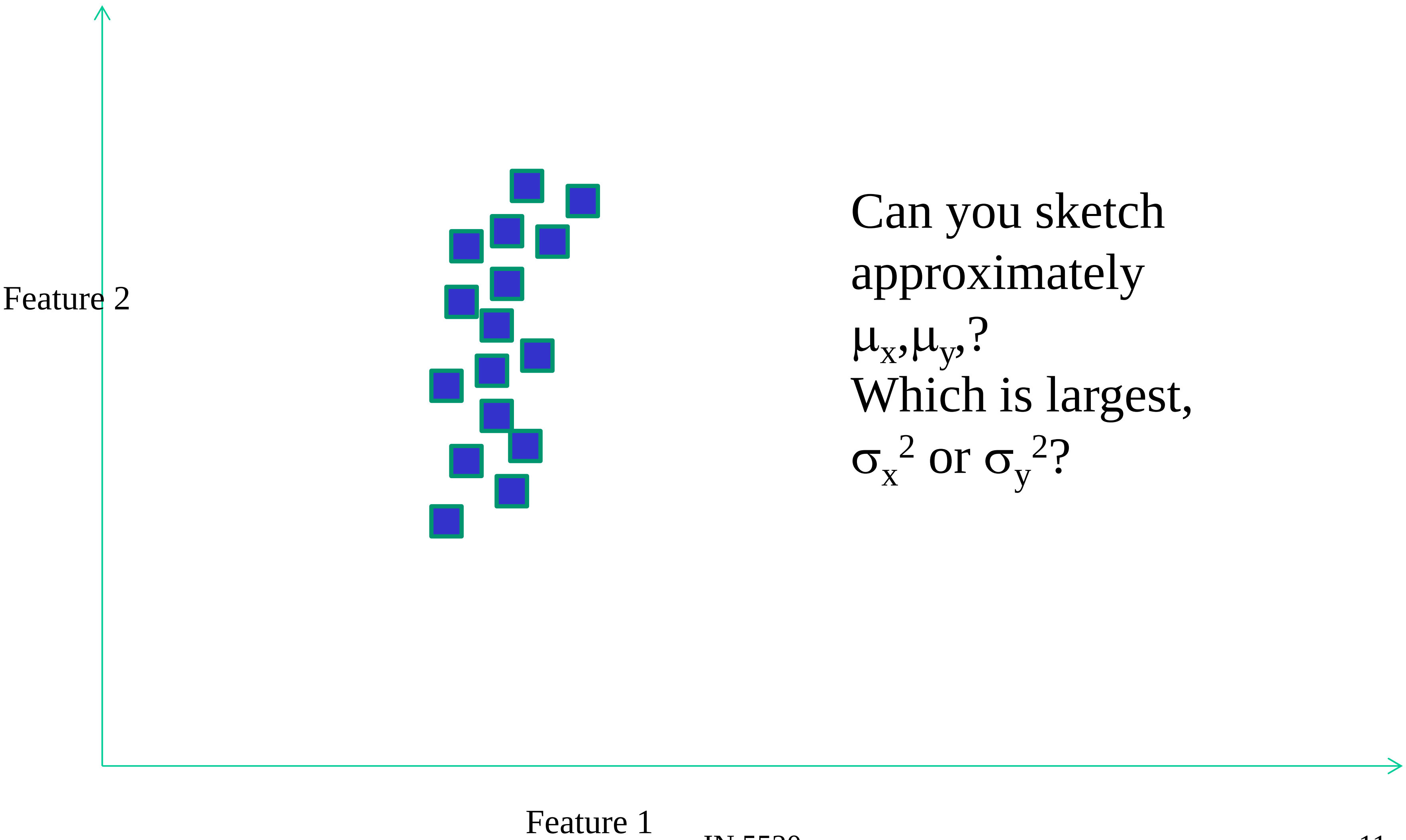
$$\sigma_y^2 = E\left[(y-\mu_y)^2\right] = \sum_x \sum_y (y-\mu_y)^2 P(x,y)$$

Variance of featureyx

$$\sigma_{xy} = E\left[(x-\mu_x)(y-\mu_y)\right] = \sum_x \sum_y (x-\mu_x)(y-\mu_y)P(x,y)$$

Covariance of feature x and y

Where (in this course) have you seen similar formulas?



Can you sketch approximately μ_x, μ_y,?
Which is largest, σ_x^2 or σ_y^2?

Statistical independence - definitions

- Variables *x* and *y* are statistical independent if and only if

$$P(x,y) = P_x(x)P_y(y)$$

- In words: two variables are indepentent if the occurrence of one does not affect the other.
- If two variables are not independent, they are dependent.
- If two variables are independent, they are also uncorrelated.
- For more than two variables: all pairs must be independent.
- Two variables are uncorrelated if

$$\sigma_{xy} = 0$$

- If Cov[X ,Y] = E[X Y] – E[X]E[Y] =0, we must have E[X Y] = E[X]E[Y]

- If two variables are uncorrelated, they *can* still be dependent.

Conditional probability

- If two variables are statistically dependent, knowing the value of one of them lets us get a better estimate of the value of the other one. We need to consider their covariance.
- The conditional probability of *x* given *y* is defined:

$$\Pr[x = i \mid y = j] = \frac{\Pr[x = i, y = j]}{\Pr[y = j]}$$

and for distributions:

$$\mathrm{P}(\mathrm{x} \mid \mathrm{y}) = \frac{\mathrm{P}(\mathrm{x}, \mathrm{y})}{\mathrm{P}(\mathrm{y})}$$

- Example: Draw two cards from a deck. Drawing a king in the first draw has probability 4/52. Drawing a king in the second draw (given that the first draw gave a king) is 3/51.


QUESTIONS
ANSWERS

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

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

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

- If we have l features, μ_s is a vector of length l and and Σ_s a $l \times l$ matrix (depends on class *s*)

$$\boldsymbol{\mu}_S = \begin{bmatrix} \mu_{1s} \\ \mu_{2s} \\ \\ \\ \mu_{ls} \end{bmatrix} \qquad \mathbf{\Sigma}_S = \begin{bmatrix} \sigma_{11} & \sigma_{12} & . & . & \sigma_{1l} \\ \sigma_{21} & \sigma_{22} & . & . & . \\ \sigma_{31} & \sigma_{11} & . & . & . \\ . & . & . & . & . \\ \sigma_{l1} & \sigma_{l2} & . & \sigma_{ll-1} & \sigma_{ll} \end{bmatrix}$$

Symmetric l×l matrix
σ_{ii} is the variance of feature i
σ_{ij} is the covariance between feature i and feature j
Symmetric because $\sigma_{ij} = \sigma_{ji}$

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

Inspecting p(**x**|ωs)

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

Scalar probability

Scalar

$1 \times l$ vector transposed

$l \times 1$ vector transposed

$l \times l$ matrix
Inverse of covariance matrix

Hint: Set up an example with $x=[x_1, x_2]^T$ and check dimensions

The mean vectors μ_s for each class

- The mean vector for class s is defined as the expected value of **x**:

$$\boldsymbol{\mu}_s = E[\mathbf{x}] = \begin{bmatrix} E(x_1) \\ E(x_2) \\ . \\ . \\ E(x_l) \end{bmatrix} = \begin{bmatrix} \mu_1^s \\ \mu_2^s \\ . \\ . \\ \mu_l^s \end{bmatrix}$$

class s

feature number l

- with l features, the mean vector μ will be of size $1 \times l$

Link to moments

- From lecture on moments:

$$m_{10} = \sum_x \sum_y x f(x,y) = \bar{x} m_{00} \quad \Rightarrow \quad \bar{x} = \frac{m_{10}}{m_{00}}$$

$$m_{01} = \sum_x \sum_y y f(x,y) = \bar{y} m_{00} \quad \Rightarrow \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

  - m_{00} was the number of pixels in the object

- If **f**=[x,y] is a sample from distribution p(x,y), the mean is defined as

$$\mu_x = \sum_x \sum_y x p(x,y)$$

$$\mu_y = \sum_x \sum_y y p(x,y)$$

Remark – what is maximum likelihood estimation

- The true value of μ and Σ is unknown.
- A distribution has some unknown parameters
- Maximum likelihood estimation:
  - These parameters are assumed unknown, but deterministic (not random), meaning that they have a single true, uknown value (and no uncertainty)
  - Estimate by finding the value that maximize the likelihood given the set of observed samples
- Bayesian estimation, on the other hand, assumes that these parameters are random variables from some distribution.
  - A set of samples gives us the maximum aposteriori value of the parameters.

Maximum likelihood estimation

- We assume the the feature vector **x** is distributed according to $p(\mathbf{x}|\omega_k)$ if it belongs to class ω_k.
- In this case we assume $p(\mathbf{x}|\omega_k)$ is a Gaussian distribution with unknown parameters θ (μ_k and Σ_k for the Gaussian distribution).
- Let $X=[\mathbf{x}_1, \ldots \mathbf{x}_M]$ be M random samples drawn from $p(\mathbf{x}|\omega_k)$.
- If all samples are independent,
- $P(X;\, \theta_k) = \prod_{m=1}^{M} p(x_m;\, \theta_k)$
- The Maximum likelihood method estimates θ_k as the value that maximize the likelihood function:
- $\widehat{\theta_k} = \underset{\theta_k}{\text{argmax}} \prod_{m=1}^{M} p(x_m\,;\, \theta_k)$
- This is equivalent to maximizing the logarithm of this, called the log-likelihood

Estimating the mean vectors μ_s

- If we have M_s training samples that we know belong to class s, we can estimate the mean vector as (Maximum likelihood estimates given the observed samples):

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

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

For a derivation of this, see e.g.:
https://towardsdatascience.com/maximum-likelihood-estimation-explained-normal-distribution-6207b322e47f


QUESTIONS
ANSWERS

The covariance matrix Σ_s for each class

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

$$\boldsymbol{\Sigma}_s = \begin{bmatrix} \sigma_{11} & \sigma_{12} & . & . & \sigma_{1l} \\ \sigma_{21} & \sigma_{22} & . & . & \sigma_{l} \\ . & . & . & . & . \\ . & . & . & . & . \\ \sigma_{l1} & \sigma_{l2} & . & . & \sigma_{ll} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & . & . & \sigma_{1l} \\ \sigma_{21} & \sigma_2^2 & . & . & \sigma_{2l} \\ . & . & . & . & . \\ . & . & . & . & . \\ \sigma_{l1} & \sigma_{l2} & . & . & \sigma_l^2 \end{bmatrix}$$

- with *l* features, the covariance matrix Σ_s will be of size *lxl*.

$\hat{f}$

More on the covariance matrix Σ_s

- The covariance matrix Σ_s will always be symmetric and positive semidefinite.
- If all components of x have non-zero variance, Σ_s will be positive definite.
- σ_{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, Σ_s will have $l(l+1)/2$ different values.

Estimating the covariance matrix Σ_s for each class

- If we have M_s training samples that we know belong to class s, we can estimate the covariance matrix Σ_s . (The estimate of a random variable f is denoted $\hat{f}$)

$$\hat{\Sigma}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} (\mathbf{x}_m - \hat{\boldsymbol{\mu}}_s)(\mathbf{x}_m - \hat{\boldsymbol{\mu}}_s)^t$$

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

- The Maximum likelihood estimate of each term σ_{ij} is computed as:

$$\sigma_{ij,s}{}^2 = \frac{1}{M_s} \sum_{m=1}^{M_s} \left(x_{m,i} - \hat{\mu}_{i,s}\right)\left(x_{m,j} - \hat{\mu}_{j.s}\right)^t$$

for the covariance between feature i and j for class s

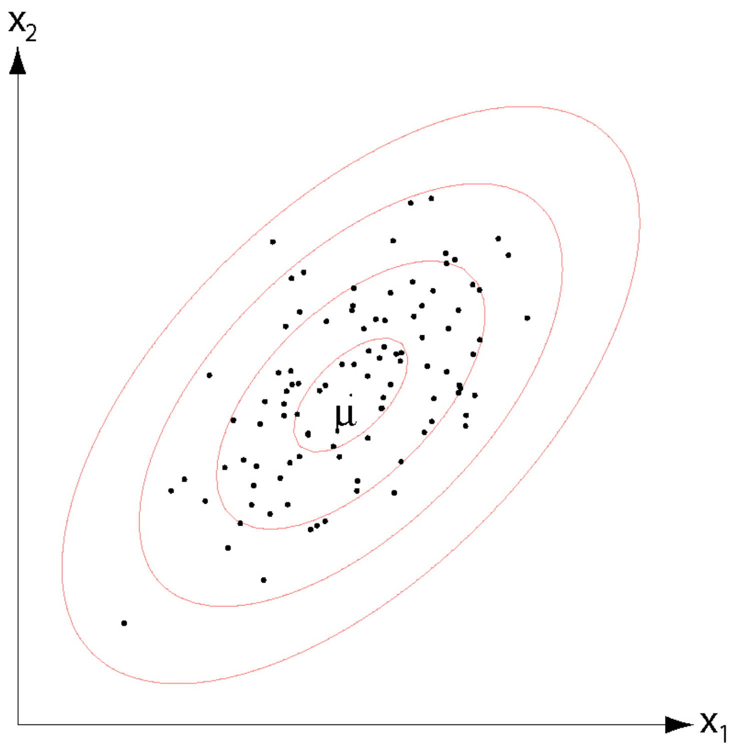
The covariance matrix and ellipses

- 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 the center point of the ellipses.
- σ_{12}, the covariance between the features defines the orientation of the ellipse.
- σ_{11} and σ_{22} defines the width of the ellipse.

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

- 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 μ is:

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



The main axes of the ellipse is determined by the eigenvectors of Σ.
The eigenvalues of Σ gives their length.

- Let us consider two features with mean 0, feature 1 has variance σ_1^2 , feature 2 variance, σ_2^2 and feature 1 and 2 has covariance 0.
- The curve of points with equal probability is given as

$$\mathbf{x}^T \Sigma^{-1} \mathbf{x} = [x_1, x_2] \begin{bmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C \quad \text{or}$$

$$\frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2} = C$$

for some constant C

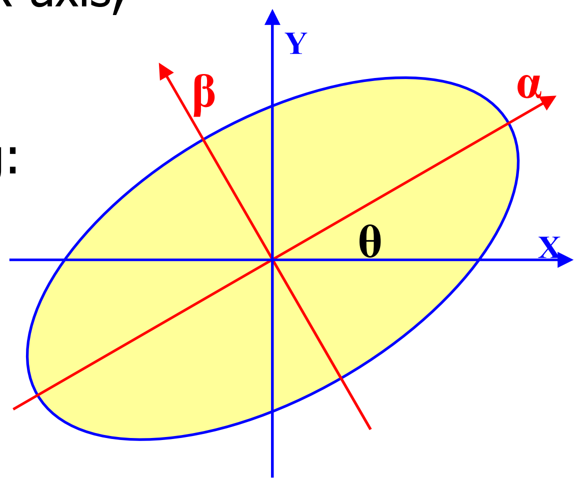
From lecture on moments: Object orientation

- Orientation is defined as the angle, relative to the X-axis, of an axis through the centre of mass that gives the lowest moment of inertia.
- Orientation θ relative to X-axis found by minimizing:

$$I(\theta) = \sum_{\alpha}\sum_{\beta} \beta^2 \; f(\alpha, \beta)$$

where the rotated coordinates are given by

$$\alpha = x\cos\theta + y\sin\theta, \qquad \beta = -x\sin\theta + y\cos\theta$$



- We found that object orientation was given by:

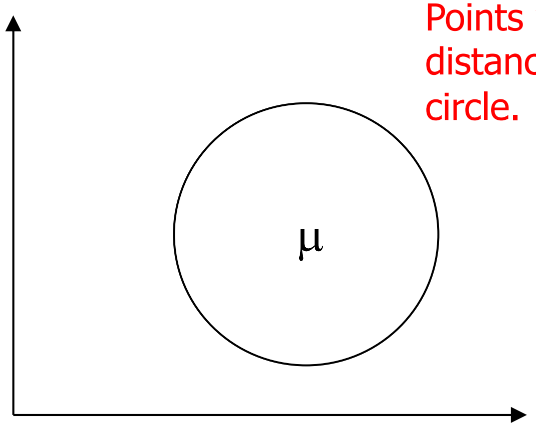
$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{11}}{(\mu_{20}-\mu_{02})}\right], \qquad \textit{where} \;\; \theta \in [0, \pi/2] \; \textit{if} \; \mu_{11} > 0, \;\; \theta \in [\pi/2, \pi] \; \textit{if} \; \mu_{11} < 0$$

Can we use this to find the orientation of the covariance matrix?

Euclidean distance vs. Mahalanobis distance

- Euclidean distance between point x and class center μ:

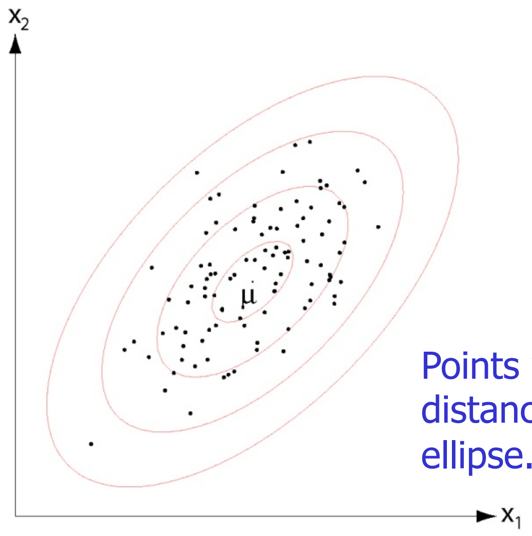
$$(x-\mu)^T(x-\mu)=\|x-\mu\|^2$$



Points with equal distance to μ lie on a circle.

- Mahalanobis distance between x and μ:

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



Points with equal distance to μ lie on an ellipse.

Back to the Gaussian:

- We now have all the terms to compute

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

Training a multivariate Gaussian classifier

- Training the classifier then consists of computing μ_s and Σ_s for all pixels with class label s in the mask file.
- For all pixels x_i with label s in the training mask, compute

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

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

$$\hat{\Sigma}_s = \frac{1}{M_s} \sum_{m=1}^{M_s} \left(\mathbf{x}_m - \hat{\boldsymbol{\mu}}_s\right)\left(\mathbf{x}_m - \hat{\boldsymbol{\mu}}_s\right)^t$$

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

How do to classification with a multiivariate Gaussian

- Decide on values for the prior probabilities, $P(\omega_j)$. If we have no prior information, assume that all classes are equally probable and $P(\omega_j)=1/J$. l is the number of features.
- Estimate μ_j and σ_j^2 based on training data based on the formulae on the previous slide. (Training)
- For each pixel in a new image:

  For class j=1,….J, compute the discriminant function

$$\mathrm{P}(\omega_j|x) = p(\mathbf{x}|\omega_j)P(\omega_j) = \frac{1}{(2\pi)^{l/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^t \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right] P(\omega_j)$$

  Assign pixel x to the class C with the highest value of $P(\omega_j|x)$ by setting label_image(x,y)= C

The result after classification is an image with class labels corresponding to the most probable class for each pixel.


QUESTIONS
ANSWERS

How a Gaussian classifier partions feature space

Discriminant functions for the normal density

- When finding the class with the highest probability, these functions are equivalent:

$$g_i(\mathbf{x}) = P(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i)P(\omega_i)}{p(\mathbf{x})}$$

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i)P(\omega_i)$$

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

- Let us now look at $g_i(\mathbf{x}) = \ln p(\mathbf{x} \mid \omega_i) + \ln P(\omega_i)$
- With a multivariate Gaussian we get:

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

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

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

- $\boldsymbol{\Sigma}_j^{-1}=\mathbf{I}/\sigma^2$
- $|\boldsymbol{\Sigma}_j|= \sigma^{2n}$

- The discriminant functions can be expressed as:

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

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

- Thus we model the probabilities as n-dimensional *spheres* because points that have equal discriminant function will lie on a circle around the mean μ_i.

Case 1: $\Sigma_j=\sigma^2 I$ – simplifying the expression

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

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

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

Common for all classes, no need to compute these terms

Since ***$x^T x$* is common for all classes**, an equivalent *g_j*(***x***) is a linear function of ***x***: .

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

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

- Now we get an equivalent formulation of the discriminant functions:

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

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

- An equation for the decision boundary $g_i(\mathbf{x})=g_j(\mathbf{x})$ can be written as

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

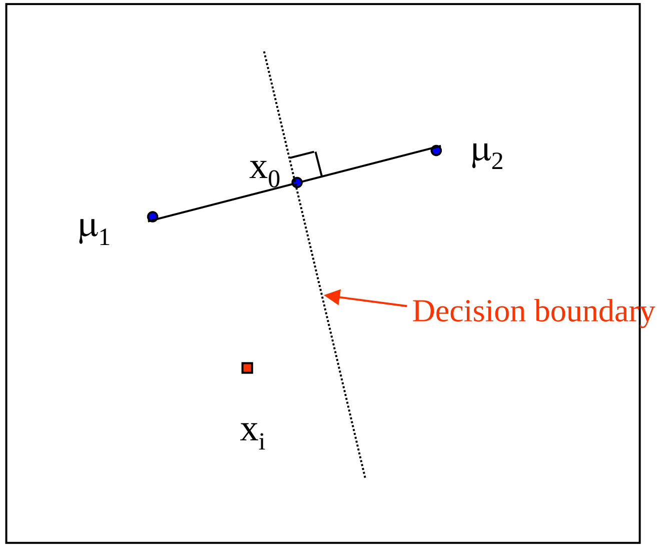
$$\text{where } \mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$$

$$\text{and } x_0 = \frac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) - \frac{\sigma^2}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2}\ln\frac{P(\omega_i)}{P(\omega_j)}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

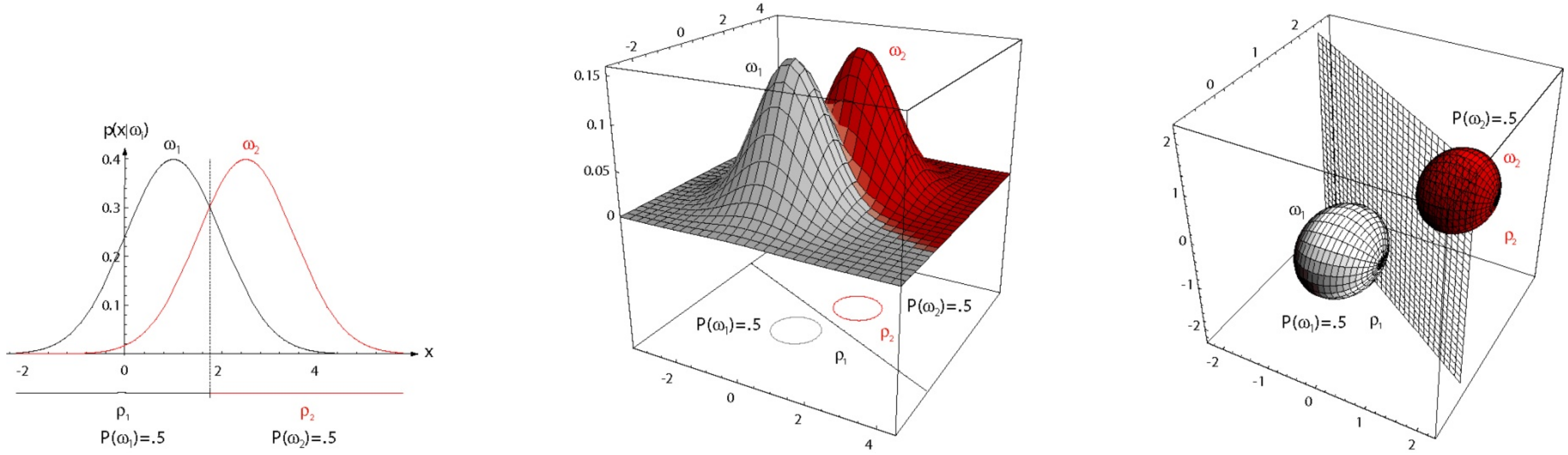
- $\mathbf{w}=\mu_i-\mu_j$ is the vector between the mean values.
- This equation defines a hyperplane through the point x_0, and orthogonal to **w**.
- If $P(\omega_i)=P(\omega_j)$ the hyperplane will be located halfway between the mean values.
- Proving this involves some algebra, see the proof at https://www.byclb.com/TR/Tutorials/neural_networks/ch4_1.htm

Case 1: $\Sigma_j=\sigma^2 I$ – Decision boundary

- The discriminant function (when $\Sigma_j=\sigma^2 I$) that defines the border between class 1 and 2 in the feature space is a straight line.
- The discriminant function intersects the line connecting the two class means at the point $x_0=(\mu_1- \mu_2)/2$ (if we do not consider prior probabilities).
- The discriminant function will also be normal to the line connecting the means.



With l features, $\Sigma_j = \sigma^2 I$



- The distributions are spherical in *l* dimensions.
- The decision boundary is a generalized hyperplane of *l-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 ***x***.
- If $P(\omega_i) = P(\omega_j)$, the decision boundary will be half-way between μ_i and μ_j

Minimum distance classification

- If all classes have equal diagonal covariance matrix and 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**.

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

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

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

Common for all classes, no need to compute

Since $\boldsymbol{x}^T\boldsymbol{x}$ is common for all classes, $g_j(\boldsymbol{x})$ again reduces to a linear function of $\boldsymbol{x}$.

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

- An equivalent formulation of the discriminant functions is

$$g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} + wi_0$$

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

$$\text{and } wi_0 = -\frac{1}{2}\boldsymbol{\mu}_i^t \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

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

Case 2

- Do an eigenvector decomposition of Σ

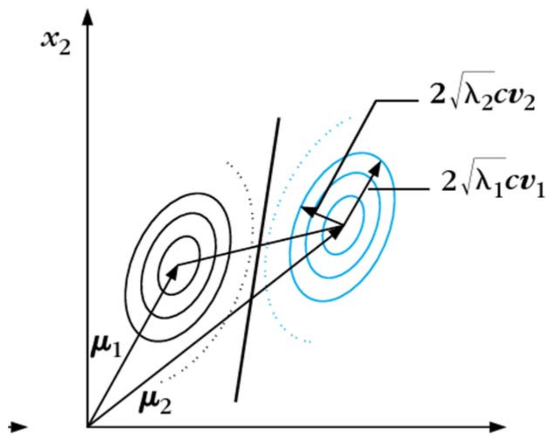
$$\text{Eigenvalues}: \lambda_1, \ldots \lambda_l$$
$$\text{Eigenvectors}: \Phi = [v_1, \ldots, v_l]$$

- Project the data onto the eigenvectors by setting **x′**=Φ^T**x**
- It can be shown that the contours with equal probability in the transformed space is:

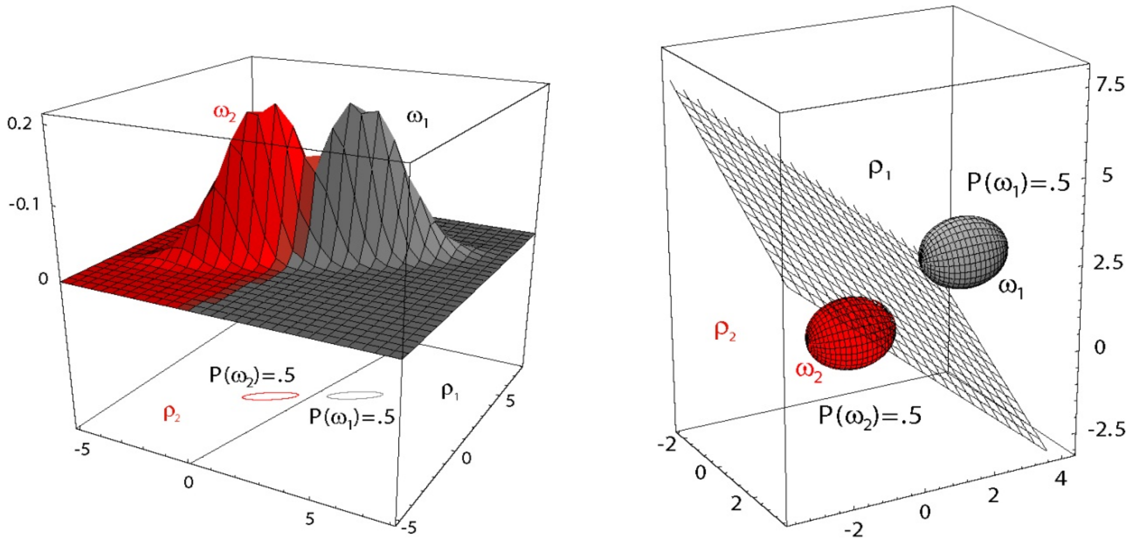
$$\frac{(x_1' - \mu_{i1})}{\lambda_1} + \ldots + \frac{(x_l' - \mu_{il})}{\lambda_l} = C^2$$

- The center of mass of the ellipses are a μ_i, the principal axes align with the eigenvalues and have length

$$2\sqrt{\lambda_k}C$$



Case 2:, $\Sigma_j = \Sigma$



- The classes can be described by hyperellipsoides in *l* dimensions.
- All hyperellipsoids have the same orientation.
- The decision boundary will again be a hyperplane.
- Because $\boldsymbol{w} = \Sigma^{-1}(\mu_i - \mu_j)$ 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:, Σ_j=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, hyperellisoides, hyperparaboloids, hyperhyperboloids...

Case 3:, Σ_j=arbitrary

- The discriminant functions will be quadratic:

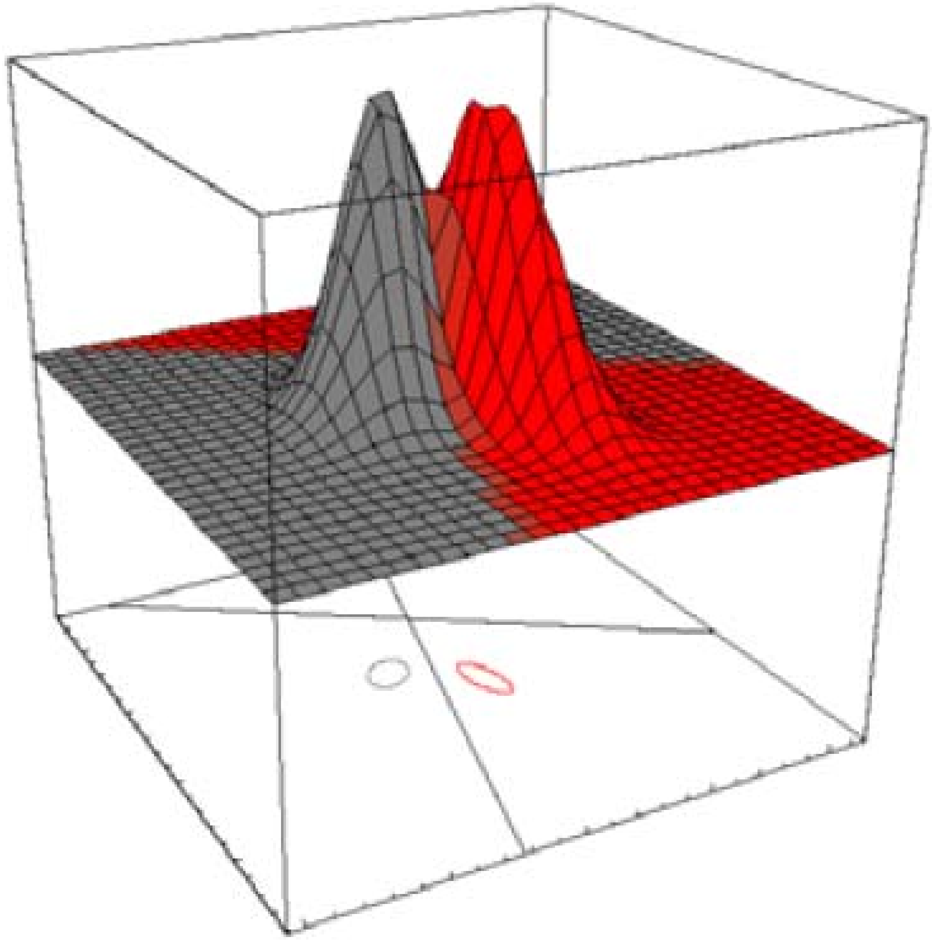
$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} + wi_0$$

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

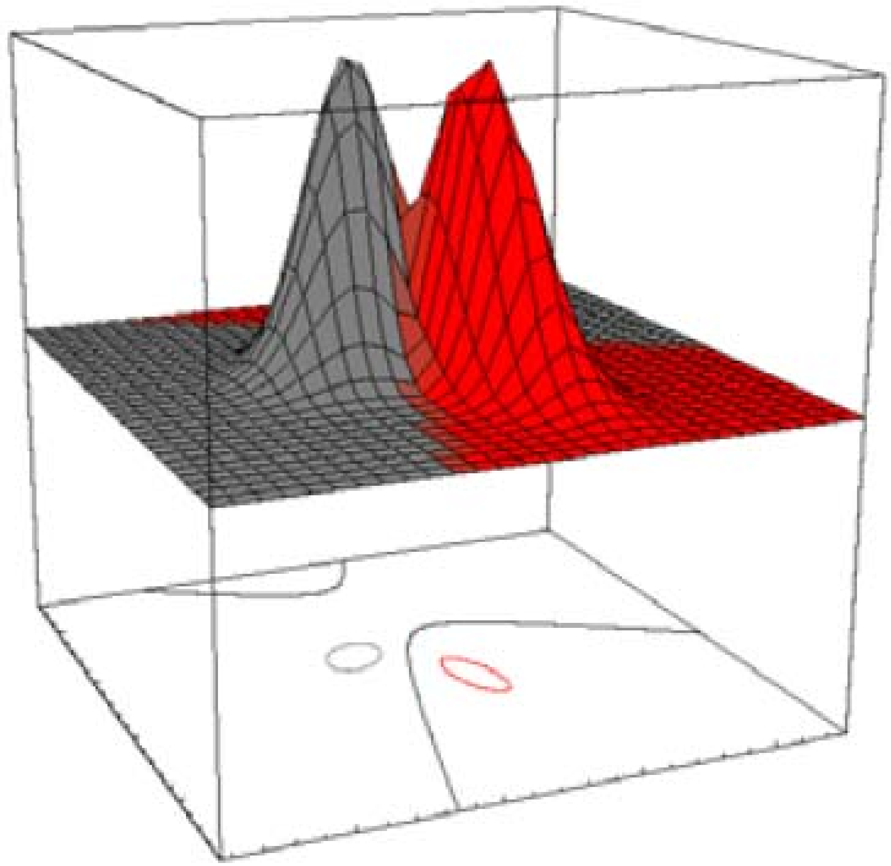
$$\text{and } wi_0 = -\frac{1}{2}\boldsymbol{\mu}_i^{\,t}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln\left|\boldsymbol{\Sigma}_i\right| + \ln P(\omega_i)$$

- The decision surfaces are hyperquadrics and can assume any of the general forms:
  - hyperplanes
  - hypershperes
  - pairs of hyperplanes
  - hyperellisoids,
  - 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.

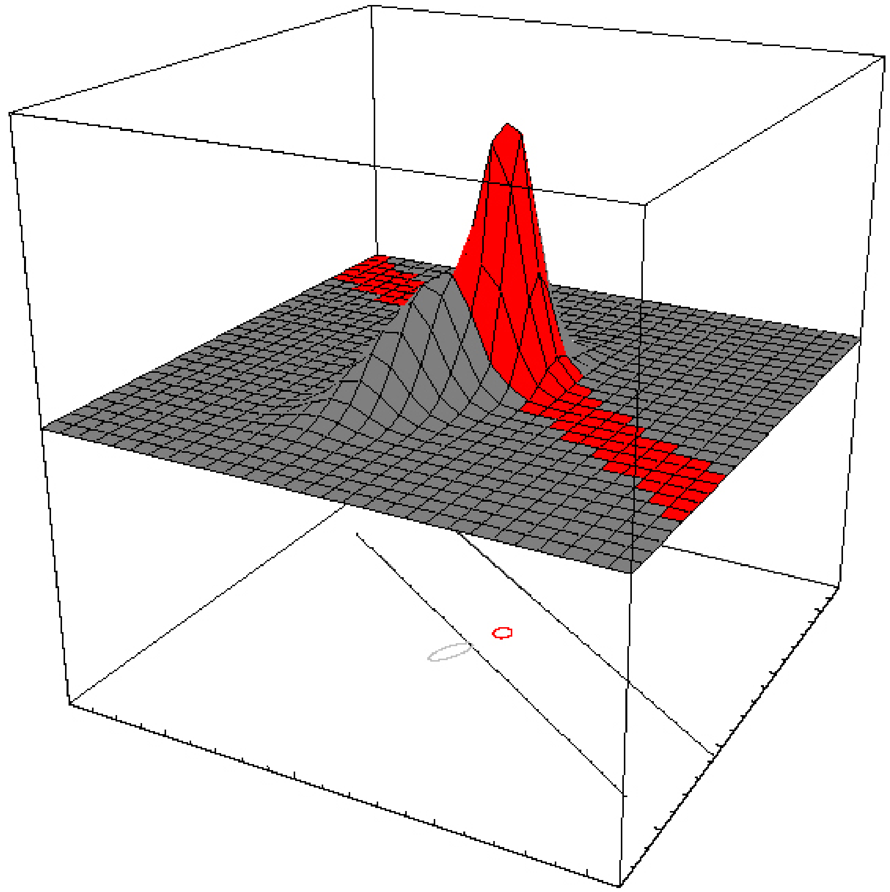
The full model, Σ_j=arbitrary - example



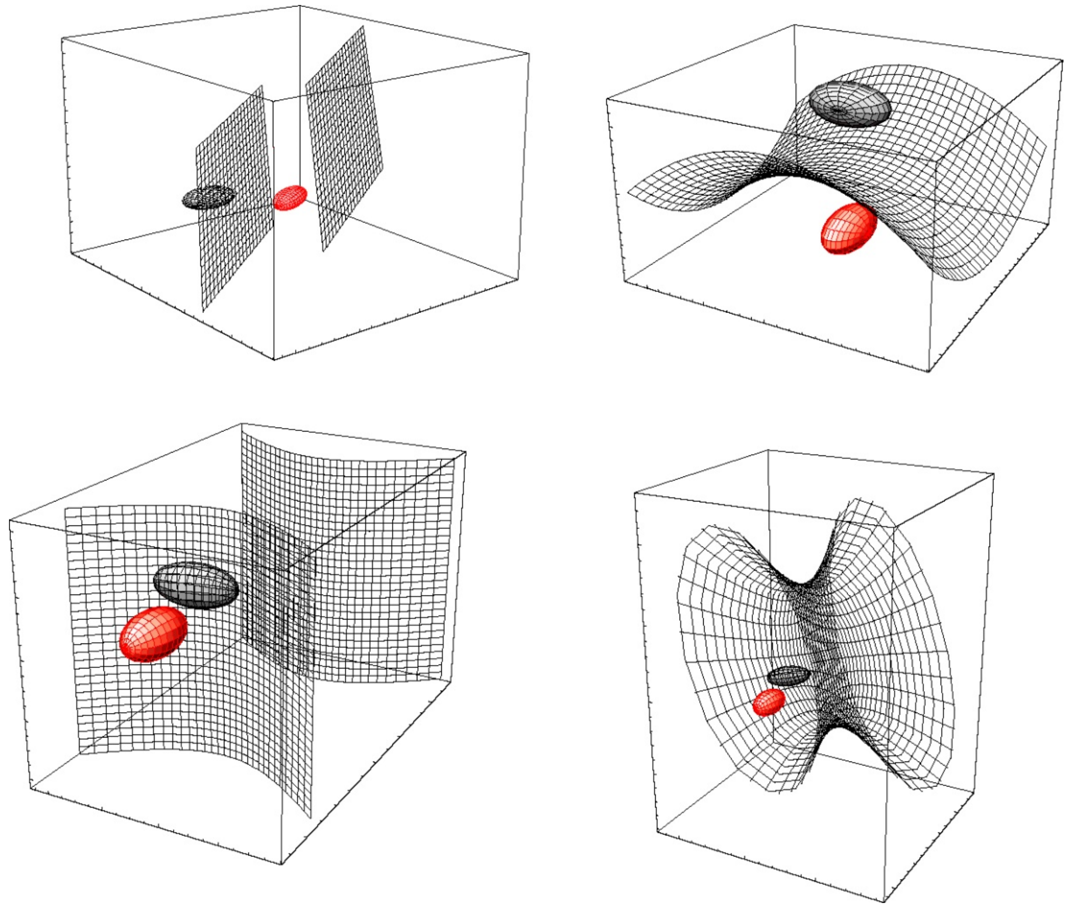
The full model, Σ_j=arbitrary - example



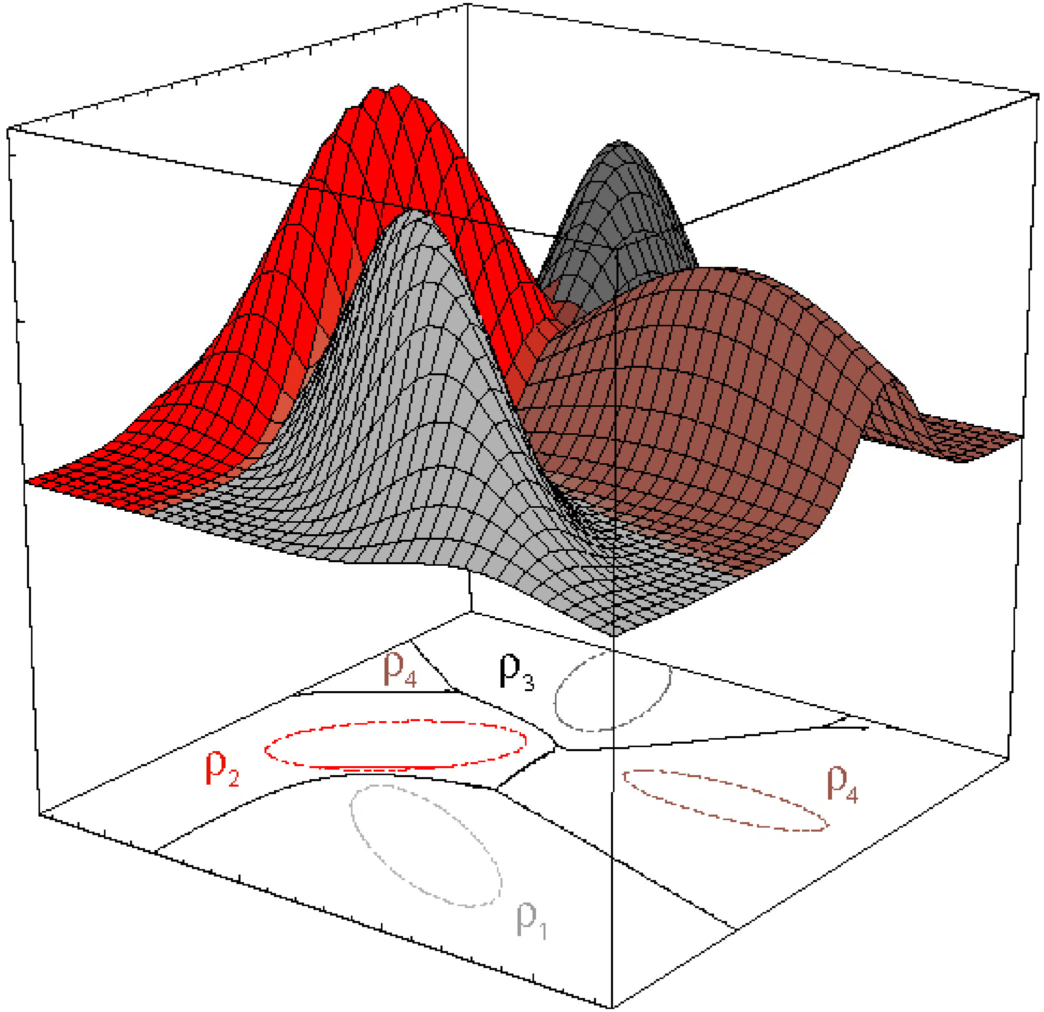
The full model, Σ_j=arbitrary - example



The full model, Σj=arbitrary - example



A multiclass example




QUESTIONS
ANSWERS

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
  - Other probability density functions (t-distribution, exponential distributions).
  - Softmax-classifier
  - Neural networks
  - Support vector machines
  - Ensembles of simple classifiers
    ADAboost
    Random forest/decision trees
  - kNN (k-Nearest-Neighbor) classification
  - Logistic classification

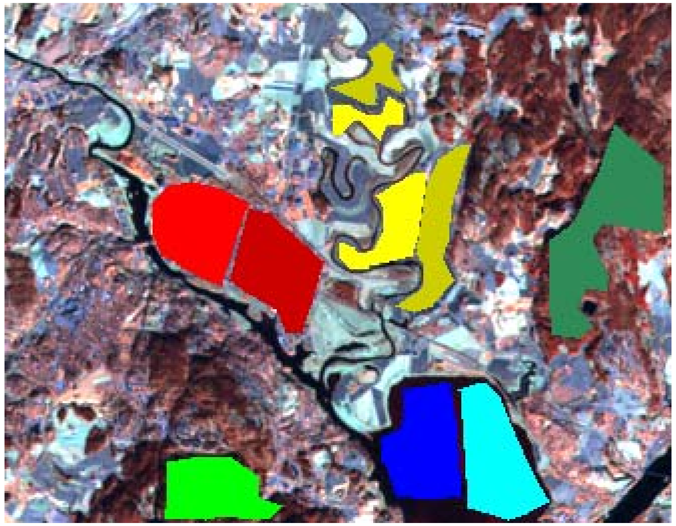
A classification example

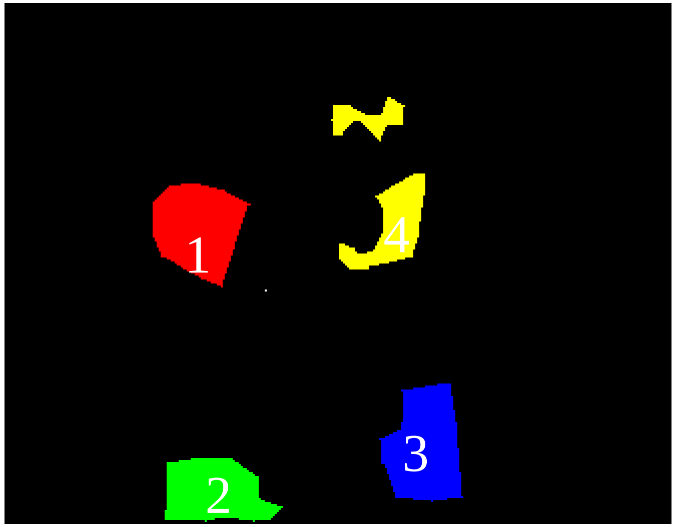
Landsat image with 6 spectral bands
The 6 bands will be the features
Training areas and test areas shown
in mask

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

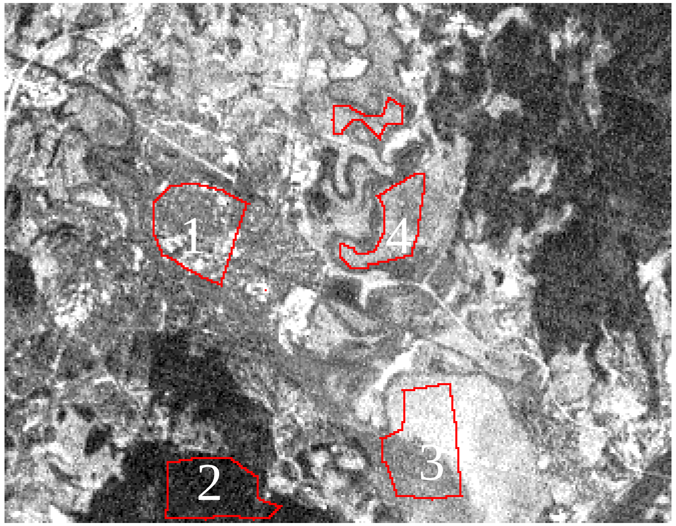
•





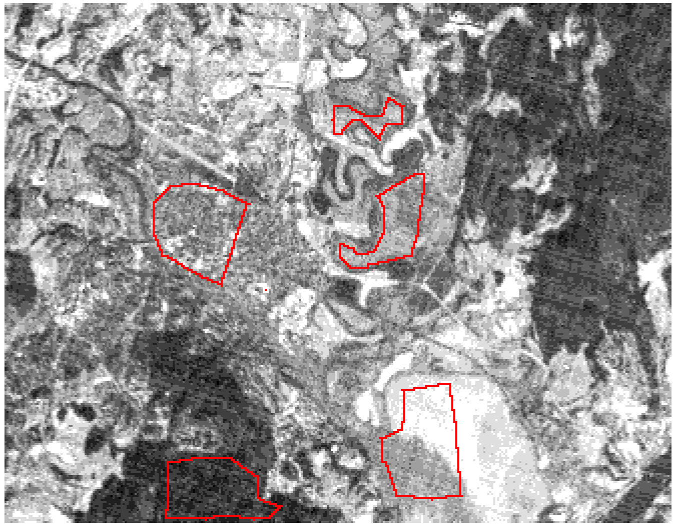
Visual inspection of feature 1

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



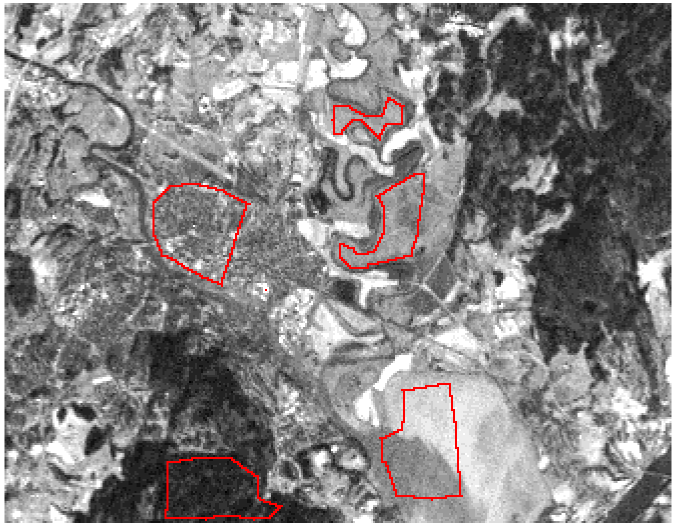
Visual inspection of feature 2

Class 2 (forest) seems to be well separated



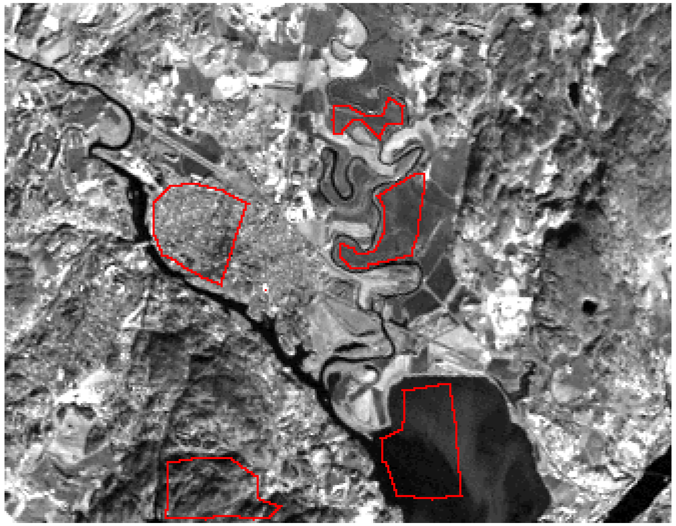
Visual inspection of feature 3

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



Visual inspection of feature 4

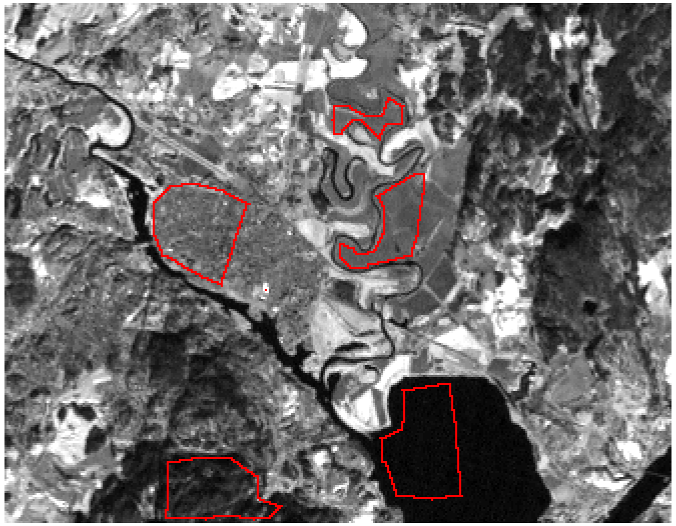
Class 1 (water) seems to be well separated,
Maybe also class 4 (agricultural)



Visual inspection of feature 5

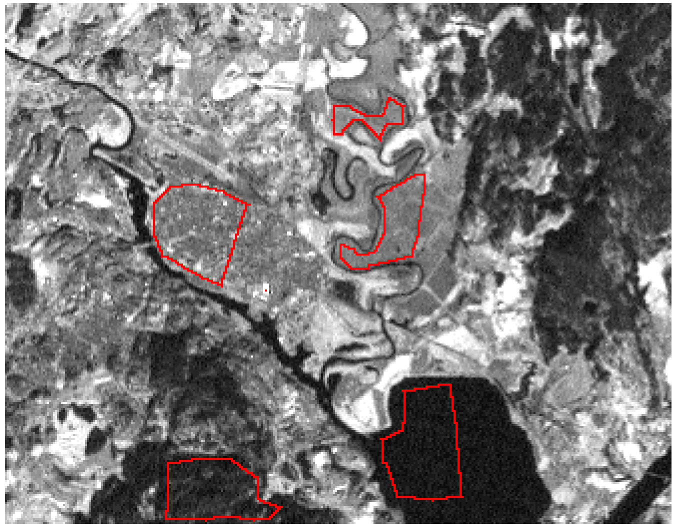
Water and forest appears similar - but the variance might be different

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

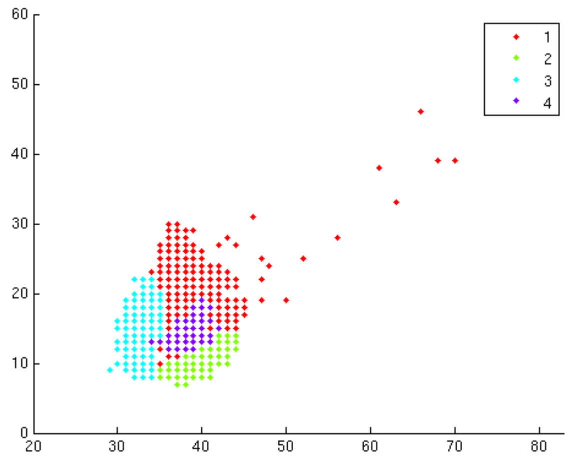


Visual inspection of feature 6

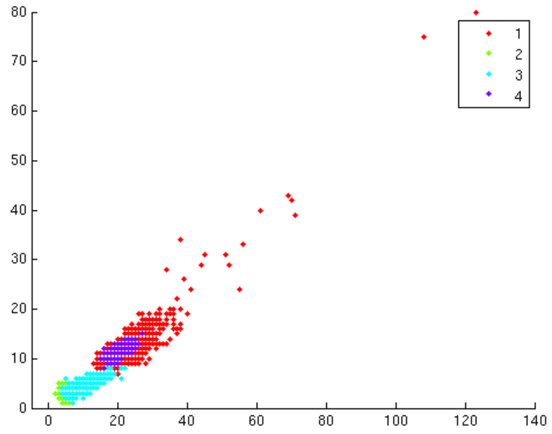
Seems similar to feature 5,
but with better contrast



Selected scatter plots (gscatter)

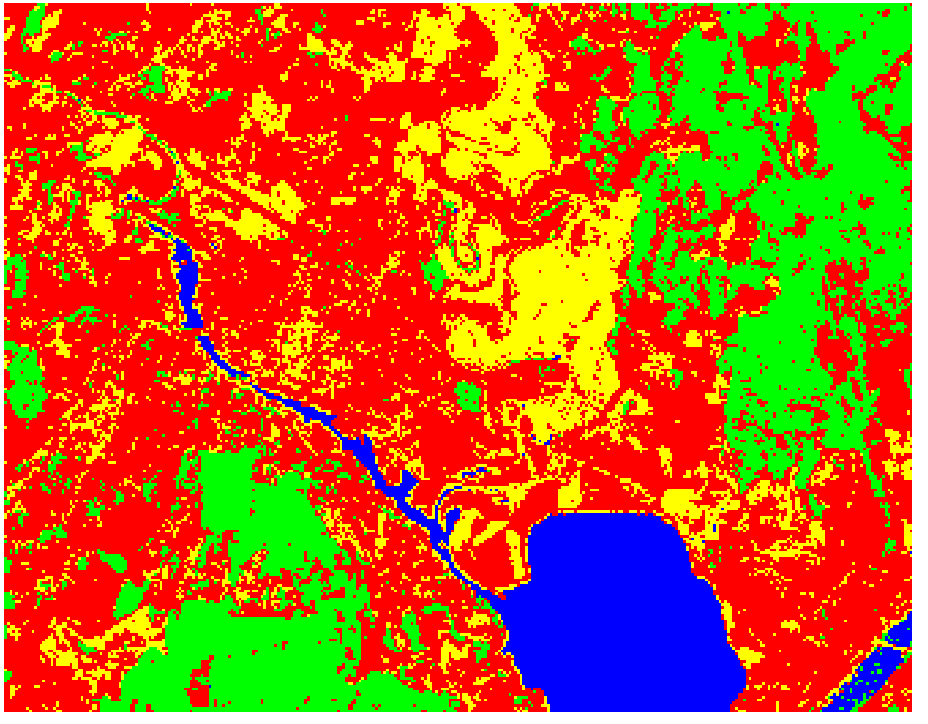


Scatterplot between feature 1 and 4



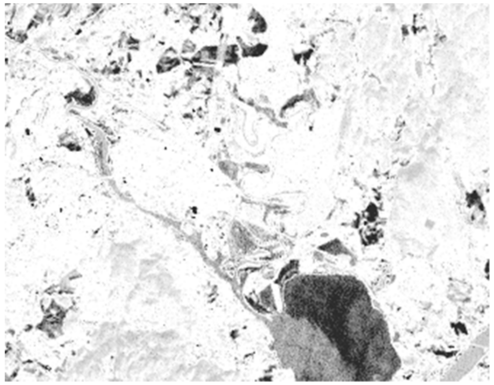
Scatterplot between feature 5 and 6

Classified images

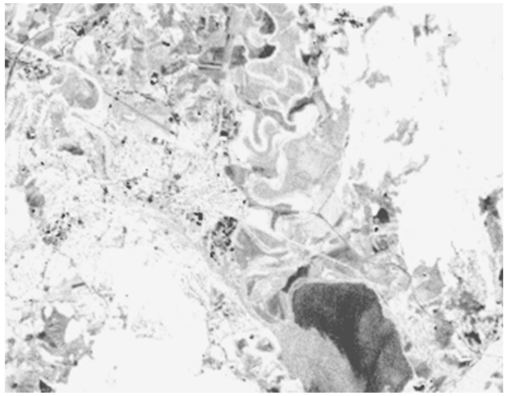


The entire image classified to the most probable class
A color table is used to display the different classes.

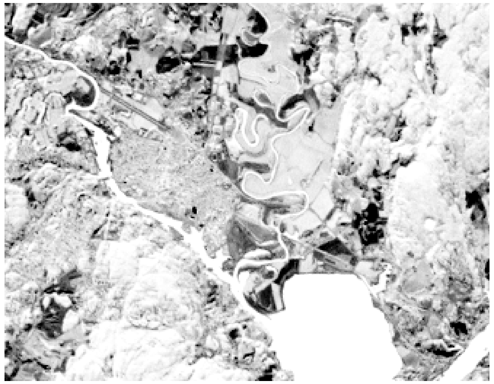
Display the posterior probabilities as images



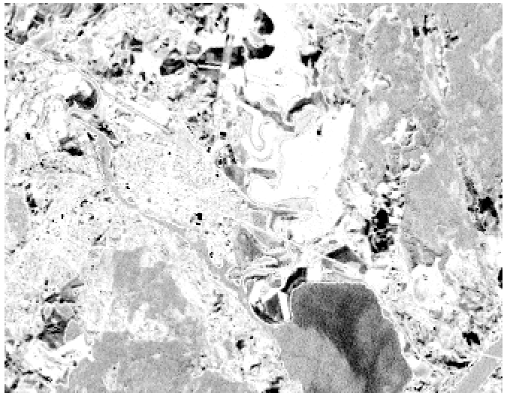
Posterior probability for class urban



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

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

Accuracy per class:

Class1: 81%

Class2: 96%

Class3: 100%

Class4: 90%

Averaged over all classes: 91.7%

Confusion matrix for the test set

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

Accuracy per class:

Class1: 85%

Class2: 81%

Class3: 98%

Class4: 86%

Averaged over all classes: 87.5%

Learning goals from this lecture

- Be able to **use and implement** Bayes rule with a l-dimensional Gaussian distribution.
- Know how μ_s and Σ_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.
- Be able to compute the discriminant function e.g. for case 1.
- Have a geometric interpretation of classification with 2 features.
- Be able to solve theoretical exercises on classification.


QUESTIONS
ANSWERS