What is "Pattern Recognition"?

The term "Pattern Recognition" ("Mustererkennung") is used for

Methods for classifying unknown objects based on feature vectors (narrow sense meaning of Pattern Recognition)

Methods or analyzing signals and recognizing interesting patterns (wide sense meaning of Pattern Recognition)

Pattern recognition can be applied to all kinds of signals, e.g.

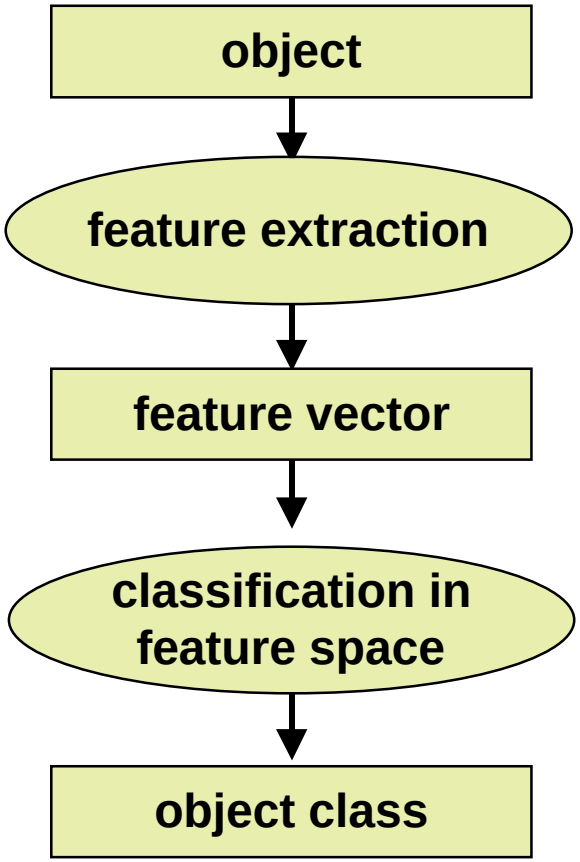
- **images**
- **acoustic signals**
- **seismographic signals**
- **tomographic data**

etc.

The following section deals with Pattern Recognition in the narrow sense.

(see Duda and Hart, Pattern Classification and Scene Analysis, Wiley 73)

Basic Terminology for Pattern Recognition



K classes $\omega_1 \ldots \omega_K$

N dimension of feature space

$\underline{x}^T = [x_1\ x_2 \ldots\ x_N]$ feature vector

$\underline{y}^T = [y_1\ y_2 \ldots\ y_N]$ prototype
(feature vector with known class membership)

$\underline{y}_i^{(k)}$ i-th prototyp of class k

M_k number of prototypes for class k

$g_k(\underline{x})$ discriminant function for class k

Problem:

Determine $g_k(\underline{x})$ such that

$g_k(\underline{x}) > g_j(\underline{x}), \quad \forall\ \underline{x} \in \omega_K \quad \forall\ k \neq j$

Example: Animal Footprints

What features can be used to distinguish the 3 footprint classes?

A Feature Space for Footprints

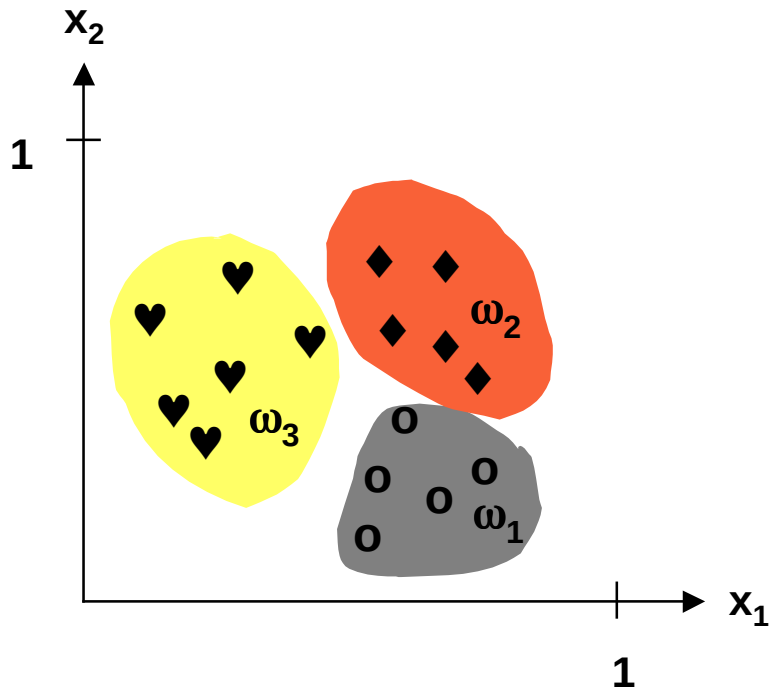
ω_1 = wolf

ω_2 = bear

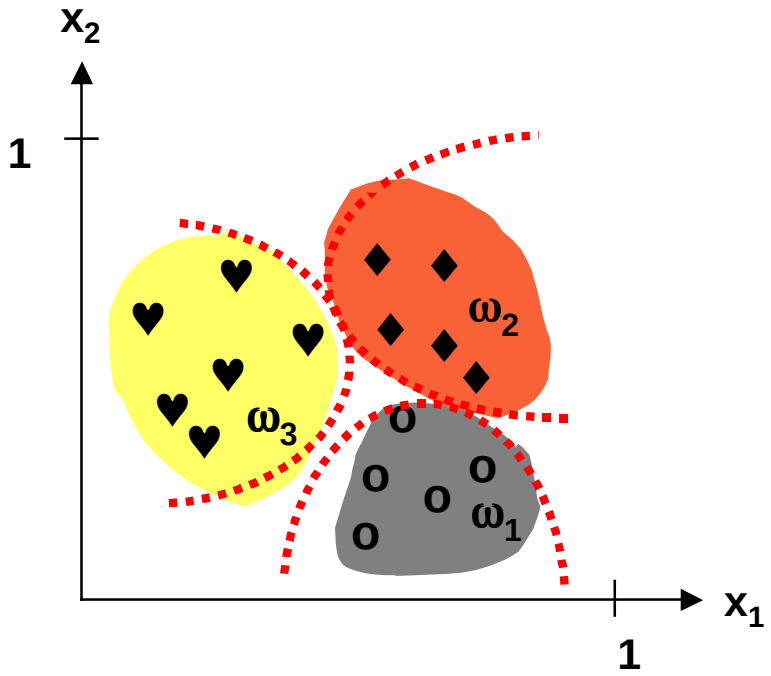
ω_3 = hare

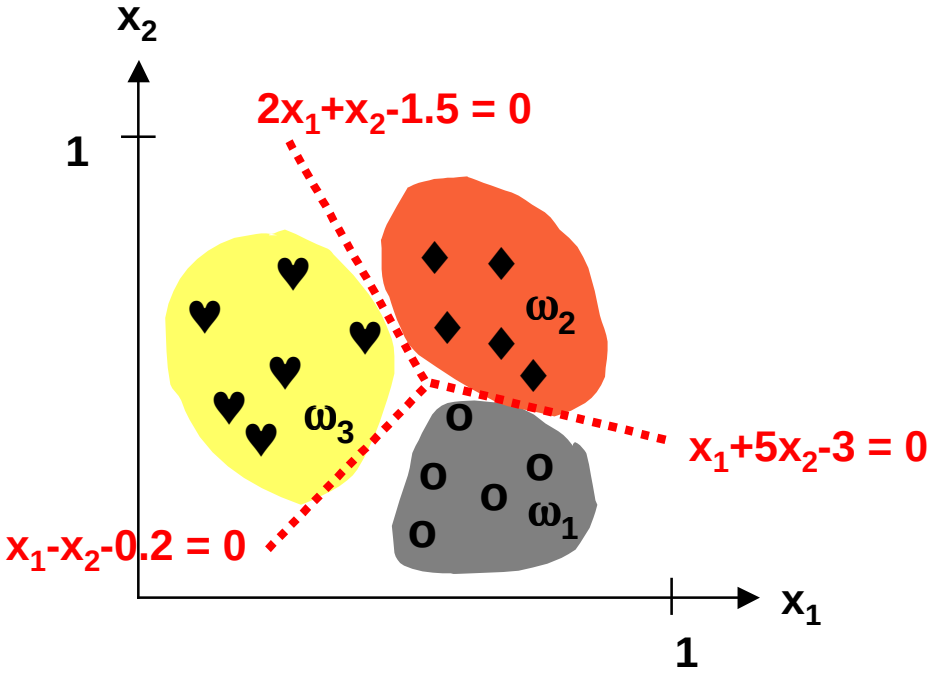
x_1 = "squareness" = $\dfrac{4bh}{(b+h)^2}$

x_2 = "solidness" = $\dfrac{\text{print area}}{bh}$



Discriminant Functions for Footprints





Quadratic discriminant functions:

$g_1 = -9x_1^2+10.8x_1-x_2-2.84$

$g_2 = x_1+20x_2^2-28x_2+9.4$

$g_3 = -x_1+5.6x_2^2-5.6x_2-1$

Piecewise linear discriminant functions:

$g_1 = (x_1-x_2-0.2 > 0) \wedge (x_1+5x_2-3 < 0)$

$g_2 = (x_1+5x_2-3 > 0) \wedge (2x_1+x_2-1.5 > 0)$

$g_3 = (2x_1+x_2-1.5 < 0) \wedge (x_1-x_2-0.2 < 0)$

Existence of Discriminant Functions

- For given prototypes, discriminant functions always exist as long as no two prototypes belonging to different classes are equal.

- If $g_i(\underline{x})$, i = 1 ... K, are discriminant functions for given prototypes, then

  $$g_i'(\underline{x}) = a(\underline{x})\, g_i(\underline{x}) + b(\underline{x}),\ \ a(\underline{x}) > 0,\ i = 1 \ldots K$$

  are also discriminant functions.

- If the classes of a 2-class problem are separable, then there always exists a function $g(\underline{x})$ such that

  $$g(\underline{x}) > 0 \qquad \forall \underline{x} \in \omega_1$$
  $$g(\underline{x}) < 0 \qquad \forall \underline{x} \in \omega_2$$

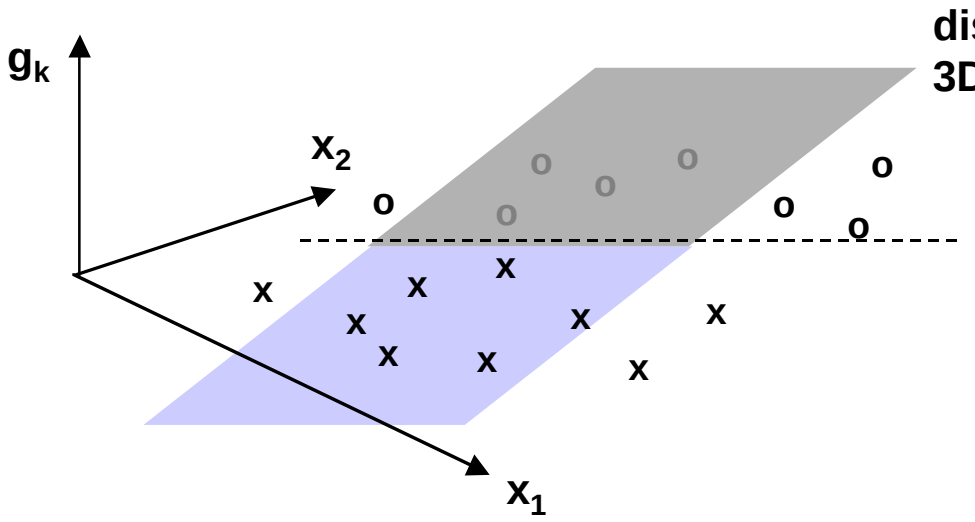
Linear Discriminant Functions

Linear discriminant functions are attractive because they can be

- **easily determined from prototypes**
- **easily analyzed**
- **easily evaluated**

Basic form of linear discriminant function:

$$g_k(\underline{x}) = \underline{w}_k^T \underline{x} + w_{k0}$$

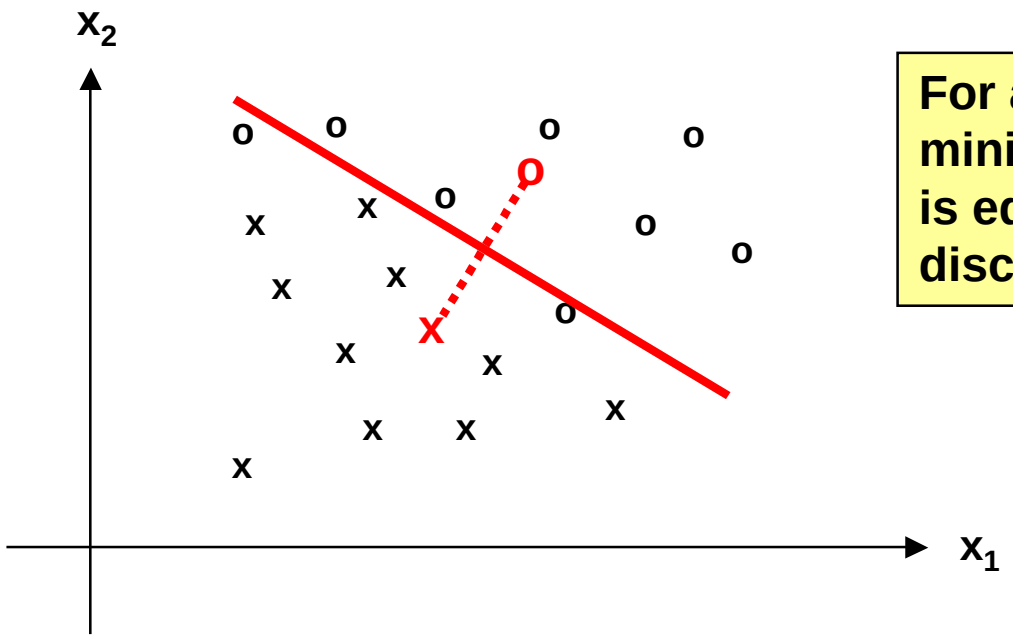


discriminant function is 3D plane

boundary line:
$\underline{w}_k^T \underline{x} + w_{k0} = 0$

Class Average
Minimal Distance Classification

- **Represent prototypes by class averages**
- **Assign object to class with minimum distance between object and class average**

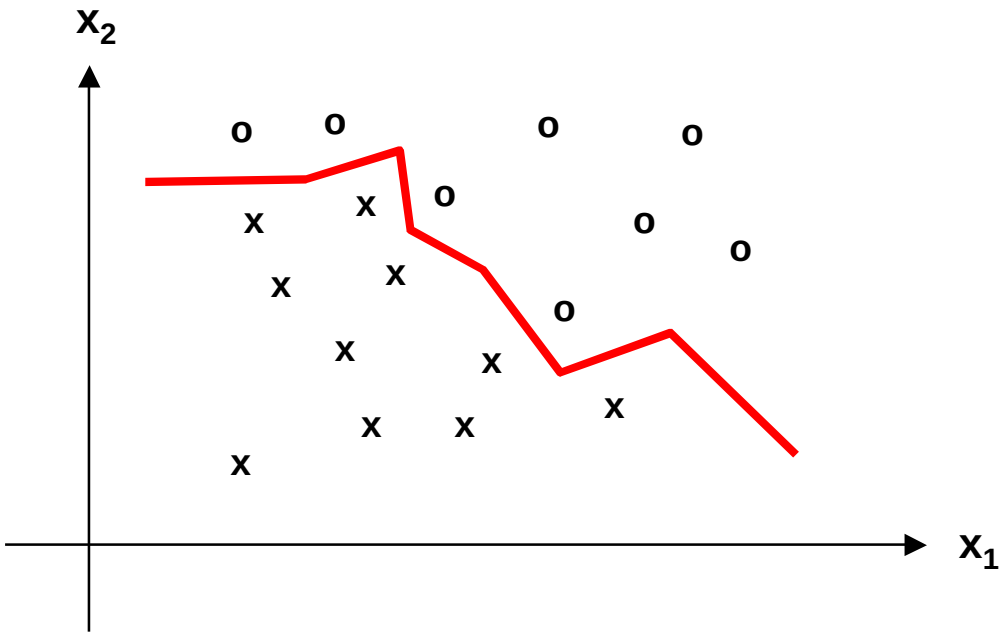


For a 2-class problem, the minimal distance criterion is equivalent to a linear discriminant function

Class average minimal distance classification may not separate prototypes even if they are linearly separable!

Nearest Neighbour Classification

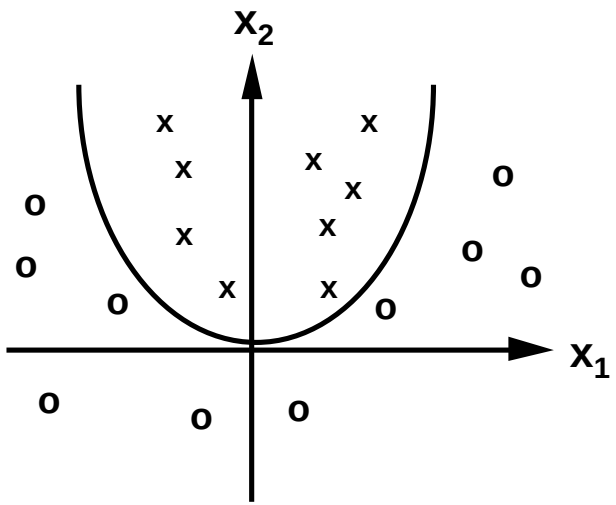
Assign object to class with nearest prototype



Piece-wise linear discriminant function

The nearest neighbour criterion classifies all prototypes correctly (except equal prototypes of different classes). The decision regions are not necessarily coherent.

Generalized Linear Discriminant Functions



Example:
Prototypes are not linearly separable

A quadratic discriminant function may work:

$$g(\underline{x}) = a_1x_1 + a_2x_2 + b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1x_2 + c$$

Transformation of prototypes into higher-dimensional feature space may allow linear discriminant functions.

Transformation for the example: $z_1 = x_1 \quad z_2 = x_2 \quad z_3 = x_1^2 \quad z_4 = x_2^2 \quad z_5 = x_1x_2$

Linear discriminant function in z-space:

$$g(\underline{z}) = a_1z_1 + a_2z_2 + b_{11}z_3 + b_{22}z_4 + b_{12}z_5 + c$$

Advantage: Linear separation algorithms may be applied
Disadvantage: Dimensionality of feature space is drastically increased

Linear Discriminant Functions for 2-Class Problems

Normalize prototypes such that

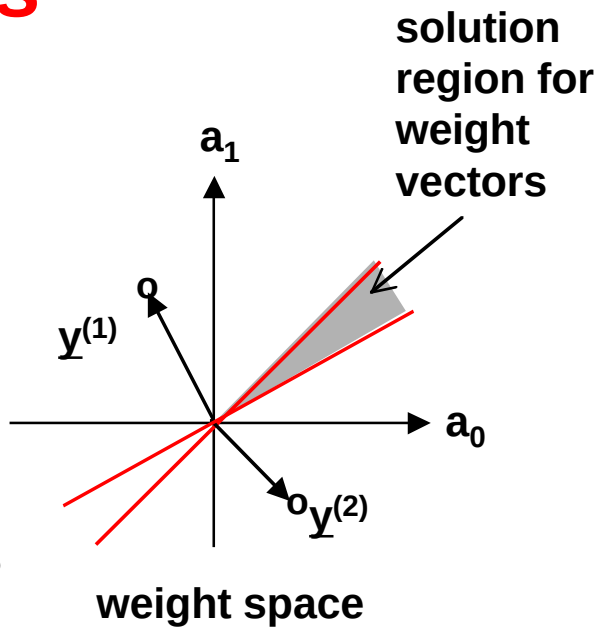
$\underline{y}^T = [1\ y_1\ y_2\ \ldots\ y_N]$

Discriminant function g can be expressed as

$g(\underline{x}) = \underline{a}^T\underline{x}$ with $\underline{a}^T = [a_0\ a_1\ \ldots\ a_N]$

Prototypes of class ω_2 are negated such that

$\underline{a}^T\underline{y} > 0$ => correct classification of both classes



Solution region in weight space (if it exists) is the space at the positive side of all hyperplanes $\underline{a}^T\underline{y} = 0$. Any weight vector $\underline{a}$ in this solution region gives a correct discriminant function.

Possible further constraints on solution vector $\underline{a}$:

$||\underline{a}|| = 1$ and $\underline{a}^T\underline{y} > b$ for all $\underline{y}$

b is "margin", i.e. minimal distance of a correctly classified point from the hyperplanes defined by the prototypes.

Perceptron Learning Rule

A solution vector a can be determined iteratively by minimizing a criterion function J(a) by gradient descent.

Perceptron criterion function:

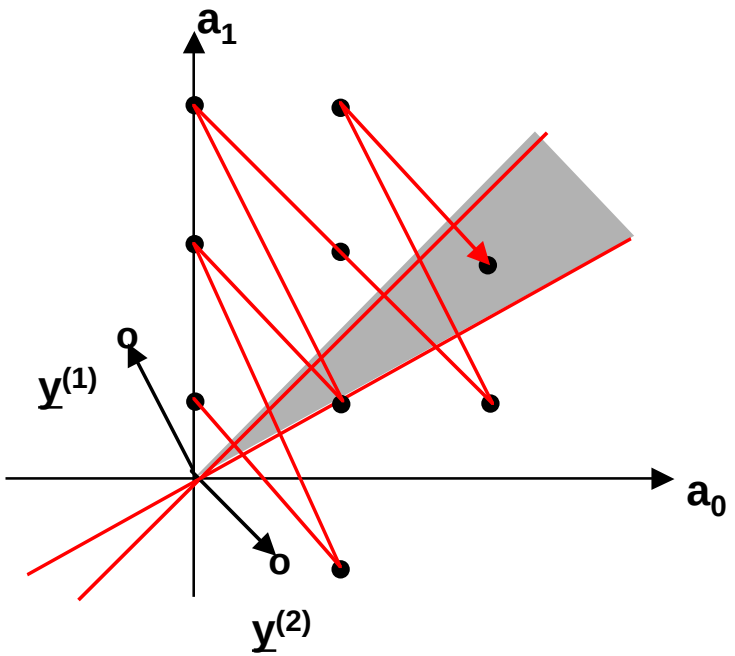
$$J_p(\underline{a}) = \sum_{y \in B} (-\underline{a}^T \underline{y})$$

with B = {all misclassified prototypes}

Gradient: $\nabla J_p(\underline{a}) = \sum_{y \in B} (-\underline{y})$

Basic gradient descent algorithm:

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \sum_{y \in B} (\underline{y})$$



iterations viewed in weight space

Example (see illustration) with
$y_1^T = [-1\ 2]$, $y_1^T = [1\ -1]$, $\rho = 2$:

k	0	1	2	3	4	5	6	7	8
$\underline{a}_k$	0 1	2 -1	0 3	2 1	0 5	2 3	4 1	2 5	4 3

solution

Minimizing the Discriminant Criterion

General form of gradient descent:

$$\underline{a}_{k+1} = \underline{a}_k - \rho_k \nabla J(\underline{a}_k) \qquad \text{with } \nabla J(\underline{a})^T = [\delta J/\delta a_0 \ \delta J/\delta a_1 \ \ldots \ \delta J/\delta a_N]$$

One can determine the optimal ρ_k which achieves the minimal $J(\underline{a}_{k+1})$ at the kth step by approximating $J(\underline{a})$ with a second-order Taylor series expansion:

$$J(\underline{a}) \approx J(\underline{a}_k) + \nabla^T J(\underline{a}_k)(\underline{a} - \underline{a}_k) + 0.5\,(\underline{a} - \underline{a}_k)^T D(\underline{a}_k)(\underline{a} - \underline{a}_k)$$

$D(\underline{a}_k)$ is the matrix of second derivatives $\delta^2 J/a_i \delta a_j$ evaluated at $\underline{a}_k$.

Using the iteration rule:

$$J(\underline{a}_{k+1}) \approx J(\underline{a}_k) - \rho_k \|\nabla J(\underline{a}_k)\|^2 + 0.5\,\rho_k^2\, \nabla J(\underline{a}_k)^T D(\underline{a}_k) \nabla J(\underline{a}_k)$$

The minimizing ρ_k is:

$$\rho_k = \frac{\|\nabla J(\underline{a}_k)\|^2}{\nabla J(\underline{a}_k)^T D(\underline{a}_k) \nabla J(\underline{a}_k)}$$

Newton´s algorithm is an alternative: Choose $\underline{a}_{k+1}$ which minimizes $J(\underline{a})$ in the Taylor series approximation.

$$\underline{a}_{k+1} = \underline{a}_k - D^{-1} \nabla J(\underline{a}_k)$$

Quadratic Criterion Function

Quadratic criterion function:

$$J_q(\underline{a}) = \sum_{y \ddagger B} (\underline{a}^T \underline{y})^2 \quad \text{with } B = \{\text{all samples where } \underline{a}^T \underline{y} \le 0\}$$

Problems:

- slow convergence close to boundaries $\underline{a}^T \underline{y} \approx 0$
- dominated by long sample vectors $\underline{y}$

Normalized quadratic criterion function:

$$J_r(\underline{a}) = \frac{1}{2} \sum_{y \in B} \frac{(\underline{a}^T \underline{y} - b)^2}{\|\underline{y}\|^2} \quad \text{with } B = \{\text{all samples where } \underline{a}^T \underline{y} \le b\}$$

Gradient:

$$\nabla J_r(\underline{a}) = \sum_{y \in B} \frac{\underline{a}^T \underline{y} - b}{\|\underline{y}\|^2} \underline{y}$$

Iteration rule:

$$\underline{a}_{k+1} = \underline{a}_k + \rho_k \sum_{y \in B_k} \frac{b - \underline{a}_k^T \underline{y}}{\|\underline{y}\|^2} \underline{y}$$

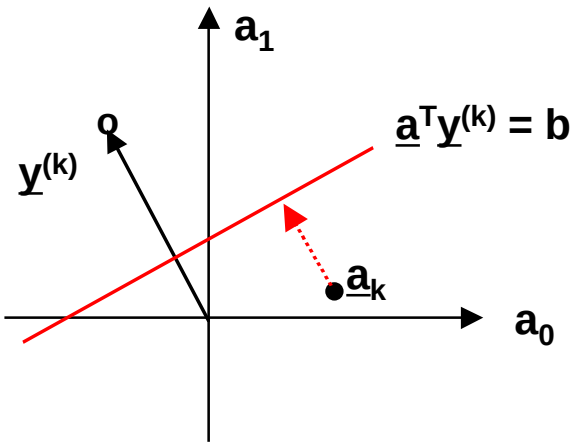
Relaxation Rule

If corrections based on the normalized quadratic criterion are performed for each single sample, one gets the "relaxation rule":

$$\underline{a}_{k+1} = \underline{a}_k + \rho \frac{b - \underline{a}_k^T \underline{y}^{(k)}}{\left\| \underline{y}^{(k)} \right\|^2} \underline{y}^{(k)} \qquad \text{where } \underline{a}^T \underline{y}^{(k)} \leq b \text{ for all } k$$

Distance from $\underline{a}_k$ to hyperplane $\underline{a}^T \underline{y}^{(k)} = b$ is $\dfrac{b - \underline{a}_k^T \underline{y}^{(k)}}{\left\| \underline{y}^{(k)} \right\|^2}$

For $\rho = 1$, the iteration rule calls for moving $\underline{a}_k$ directly to the hyperplane => "relaxation" of tension in inequality $\underline{a}^T \underline{y}^{(k)} \leq b$



Typical values: $0 < \rho < 2$

$\rho < 1$ "underrelaxation"

$\rho > 1$ "overrelaxation"

Minimum Squared Error

New criterion function for all samples:

Find $\underline{a}$ such that $\underline{a}^T\underline{y}_i = b_i$ with b_i = some positive constant

In matrix notation: $\underline{Ya} = \underline{b}$ **with** $Y = \begin{bmatrix} \underline{y}_1^T \\ \underline{y}_2^T \\ \dots \\ \underline{y}_M^T \end{bmatrix}$ **and** $\underline{y}_i^T = [\, y_{i1} \dots y_{iN} \,]$

In general, M >> N and Y^{-1} does not exist, hence $\underline{a} = Y^{-1}\underline{b}$ is no solution.

Classical solution technique: Minimize squared error criterion:

$$J_s(\underline{a}) = ||Y\underline{a} - \underline{b}||^2 = \Sigma\, (\underline{a}^T\underline{y}_i - b_i)^2$$

Closed-form solution by setting the gradient equal to 0.

$$\nabla J_s = 2Y^T(Y\underline{a} - \underline{b}) = \underline{0} \quad \Rightarrow \quad \underline{a} = (Y^TY)^{-1}Y^T\underline{b}$$

if $(Y^TY)^{-1}Y^T$ **is nonsingular**

pseudoinverse of Y

Ho-Kashyap Procedure

The MSE solution $\underline{a} = (Y^T Y)^{-1} Y^T \underline{b}$ does not necessarily provide a separating hyperplane $\underline{a}^T \underline{y} = 0$ if the classes are linearly separable, because $\underline{b}$ is chosen arbitrarily.

Ho-Kashyap algorithm searches for $\underline{a}$ and $\underline{b}$ such that $Y\underline{a} = \underline{b} > \underline{0}$ by minimizing $J_s(\underline{a}, \underline{b}) = \|Y\underline{a} - \underline{b}\|^2$ w.r.t. $\underline{a}$ and $\underline{b}$:

1. Iterate over $\underline{a}$ by choosing $\underline{a}_k = (Y^T Y)^{-1} Y^T \underline{b}_k$
2. Iterate over $\underline{b}$ by choosing $\underline{b}_1 > \underline{0}$

$$\underline{b}_{k+1} = \underline{b}_k + 2\rho \underline{e}_k^+ \qquad 0 < \rho < 1$$

with $\underline{e}_k = Y\underline{a}_k - \underline{b}_k$ error vector

$\underline{e}_k^+ = (\underline{e}_k + |\underline{e}_k|)/2$ positive part $\underline{e}_k$

Ho-Kashyap iteration over $\underline{b}$ generates sequence of margin vectors $\underline{b}$ which

- minimizes squared error criterion
- gives only positive margins $\underline{b} > \underline{0}$

For linearly separable classes and $0 < \rho < 1$, the Ho-Kashyap algorithm will converge in a finite number of steps.

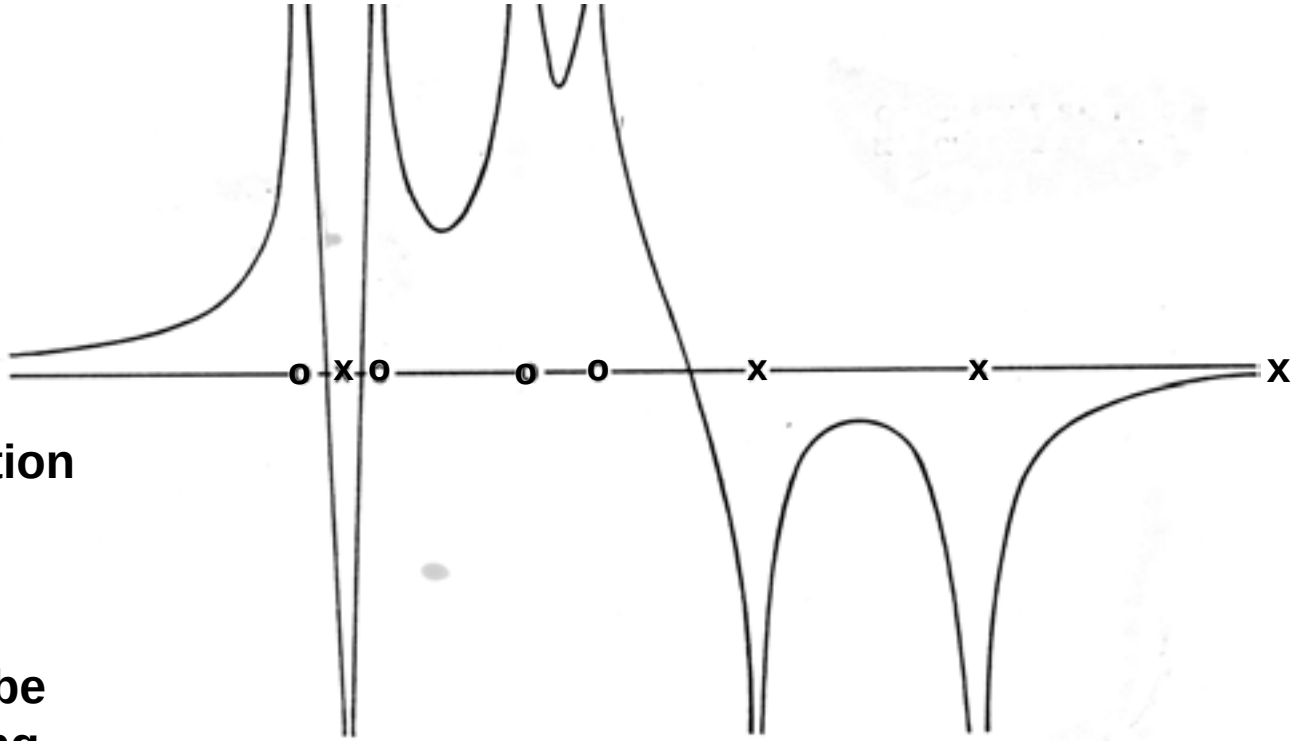
Discrimination with Potential Functions

Idea: Electrostatic potential centered at each prototype may sum up to a useful discriminant function

Example:

potential function

$K(\underline{x}, \underline{x}_i) = 1/||\underline{x}-\underline{x}_i||^2$



discriminant function

$g(\underline{x}) = \Sigma\, q_i\, K(\underline{x}, \underline{x}_i)$

"charges" q_i may be adjusted in learning procedure

Construction of Discriminant Functions Based on Potential Functions

Different choices for potential functions are possible, for example:

$$K(\underline{x}, \underline{x}_k) = \frac{\sigma^2}{\sigma^2 + \|\underline{x} - \underline{x}_k\|^2}$$

$$K(\underline{x}, \underline{x}_k) = \exp\left[-\frac{1}{2\sigma^2}\|\underline{x} - \underline{x}_k\|^2\right]$$

Potential functions must be tuned to provide the right kind of interpolation between samples

Iterative construction:

$$g'(\underline{x}) = \begin{cases} g(\underline{x}) + K(\underline{x}, \underline{x}_k) & \text{if } \underline{x}_k \text{ is of class 1 and } g(\underline{x}_k) \le 0 \\ g(\underline{x}) - K(\underline{x}, \underline{x}_k) & \text{if } \underline{x}_k \text{ is of class 2 and } g(\underline{x}_k) \ge 0 \\ g(\underline{x}) & \text{otherwise} \end{cases}$$

Statistical Decision Theory

Generating decision functions from a statistical characterization of classes (as opposed to a characterization by prototypes)

Advantages:

1. **The classification scheme may be designed to satisfy an objective optimality criterion:**

   Optimal decisions minimize the probability of error.

2. **Statistical descriptions may be much more compact than a collection of prototypes.**

3. **Some phenomena may only be adequately described using statistics, e.g. noise.**

Example: Medical Screening (1)

Health test based on some measurement x (e.g. ECG evaluation)

It is known that every 10th person is sick (prior probability):

ω_1 class of healthy people $P(\omega_1) = 9/10$
ω_2 class of sick people $P(\omega_2) = 1/10$

Task 1: Classify without taking any measurements (to save money)

Decision rule 1a: Classify every 10th person as sick

P(error) = P(decide sick if healthy) + P(decide healthy if sick)

= 1/10•9/10 + 9/10•1/10 = 0.18

Decision rule 1b: Classify all persons as healthy

P(error) = P(decide healthy if sick) = 1/10 = 0.1

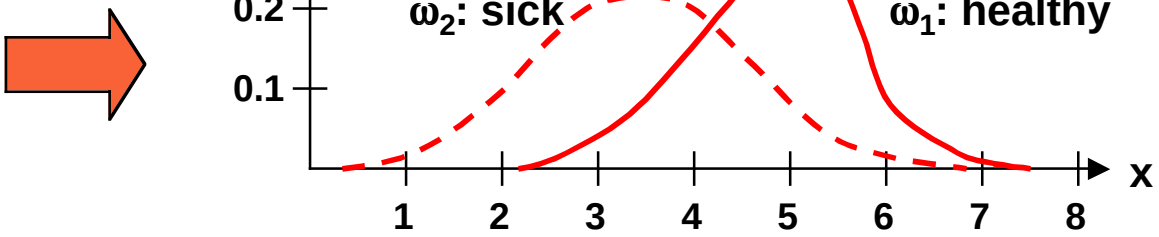
- **Decision rule 1b is better because it gives lower probability of error**
- **Decision rule 1b is optimal because no other decision rule can give a lower probability of error (try "every n-th" in 1a and minimize over n)**

Example: Medical Screening (2)

Task 2: Classify after taking a measurement x

Assume that the statistics of prototypes are given as $p(x|\omega_i)$, i = 1, 2

Person No.	x	indication
•	•	•
•	•	•
•	•	•
134	7.4	neg
135	6.8	neg
136	4.2	pos
137	5.6	neg
138	5.8	pos
139	7.2	neg
•	•	•
•	•	•
•	•	•



$P(e|x) = P(\text{error given } x) = P(\omega \neq \omega'|x) = 1 - P(\omega|x)$
where ω' is the class assigned to x by the decision rule.

$P(e|x)$ is minimized by choosing the class which maximizes $P(\omega|x)$.
Hence $g_i(x) = P(\omega_i|x)$ are discriminant functions.

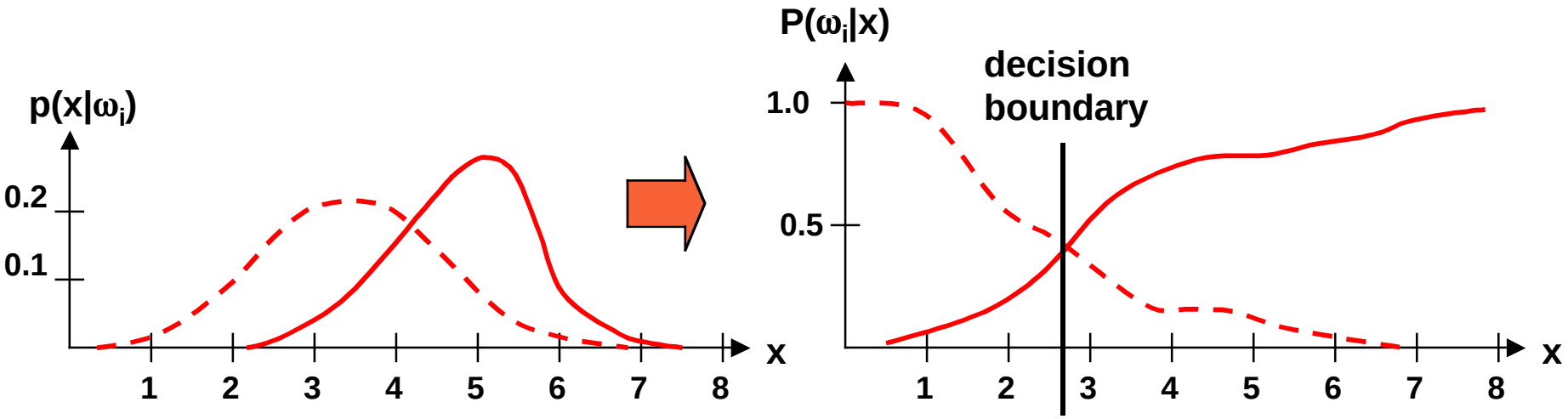
How do we get the "posterior" probabilities $P(\omega_i|x)$?

Example: Medical Screening (3)

The posterior probabilities $P(\omega_i|x)$ can be computed from the "likelihood" $p(x|\omega_i)$ using Bayes´ formula:

$$P(\omega_i|x) = \frac{p(x|\omega_i)\bullet P(\omega_i)}{p(x)} = \frac{p(x|\omega_i)\bullet P(\omega_i)}{\sum_i p(x|\omega_i)P(\omega_i)}$$

For the example, using Bayes´ Formula, one could get:



General Framework for Bayes Classification

Statistical decision theory which minimizes the probability of error for classifications based on uncertain evidence

$\omega_1 \ldots \omega_K$	K classes
$P(\omega_k)$	prior probability that an object of class k will be observed
$\underline{x} = [x_1 \ldots x_N]$	N-dimensional feature vector of an object
$p(\underline{x}\|\omega_k)$	conditional probability ("likelihood") of observing $\underline{x}$ given that the object belongs to class ω_K
$P(\omega_k\|\underline{x})$	conditional probability ("posterior probability") that an object belongs to class ω_K given $\underline{x}$ is observed

Bayes decision rule:

Classify given evidence $\underline{x}$ as class ω' such that ω' minimizes the probability of error $P(\omega \neq \omega' | \underline{x})$

=> Choose ω' which maximizes the posterior probability $P(\omega | \underline{x})$

$g_i(\underline{x}) = P(\omega_i|\underline{x})$ are discriminant functions.

Bayes 2-class Decisions

If the decision is between 2 classes ω_1 and ω_2, the decision rule can be simplified:

$$\text{Choose } \omega_1 \text{ if } \frac{p(\underline{x}|\omega_1)}{p(\underline{x}|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)}$$

Several alternative forms are possible for a discriminant function:

$$g(\underline{x}) = P(\omega_1|\underline{x}) - P(\omega_2|\underline{x}) \quad \Rightarrow \quad g(\underline{x}) = \log\frac{p(\underline{x}|\omega_1)}{p(\underline{x}|\omega_2)} + \log\frac{P(\omega_1)}{P(\omega_2)}$$

$$g(\underline{x}) = \frac{p(\underline{x}|\omega_1)}{p(\underline{x}|\omega_2)} - \frac{P(\omega_2)}{P(\omega_1)}$$

For exponential and Gaussian distributions it is useful to take the logarithm:

$$g(\underline{x}) = \log\frac{P(\omega_1|\underline{x})}{P(\omega_2|\underline{x})} = \log\frac{P(\underline{x}|\omega_1)P(\omega_1)}{P(\underline{x}|\omega_2)P(\omega_2)} = \log\frac{P(\underline{x}|\omega_1)}{P(\underline{x}|\omega_2)} - \log\frac{P(\omega_2)}{P(\omega_1)}$$

Normal Distributions

Gaussian ("normal") multivariate distribution:

$$p(\underline{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\underline{x}-\underline{\mu})^T \Sigma^{-1}(\underline{x}-\underline{\mu})\right]$$

$\Sigma = E[(\underline{x} - \underline{\mu})^T(\underline{x} - \underline{\mu})]$ N-by-N covariance matrix

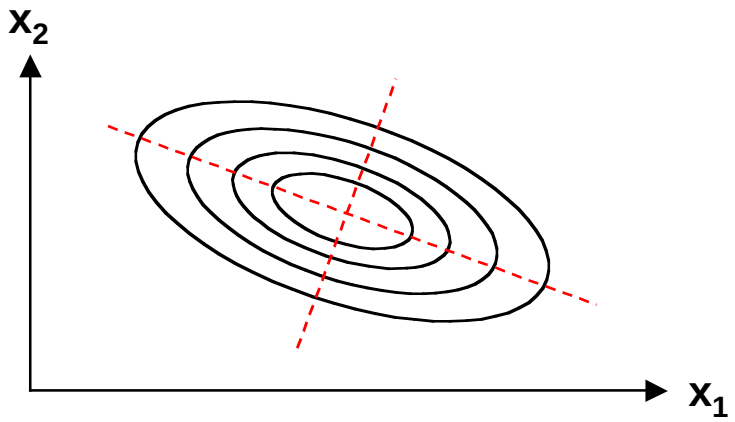
$\underline{\mu} = E[\underline{x}]$ mean vector

For decision problems, loci of points of constant density are interesting. For Gaussian multivariate distributions, these are hyperellipsoids:

$$(\underline{x} - \underline{\mu})^T\Sigma^{-1}(\underline{x} - \underline{\mu}) = \text{constant}$$

Eigenvectors of Σ determine directions of principal axes of the ellipsoids, eigenvalues determine lengths of the principal axes.

$d^2 = (\underline{x} - \underline{\mu})^T\Sigma^{-1}(\underline{x} - \underline{\mu})$ is called "squared Mahalanobis distance" of $\underline{x}$ from $\underline{\mu}$.



Discriminant Function for Normal Distributions

General form:

$$g_i(\underline{x}) = \log p(\underline{x}|\omega_i) + \log P(\omega_i)$$

For $p(\underline{x}|\omega_i) \sim N(\underline{\mu}_i, \Sigma_i)$:

$$g_i(\underline{x}) = -1/2\,(\underline{x} - \underline{\mu}_i)^T\Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) - N/2 \log 2\pi - 1/2 \log|\Sigma_i| + \log P(\omega_i)$$

$N/2 \log 2\pi$: irrelevant

We consider the discriminant functions for three interesting special cases:

- **univariate distribution N=1**
- **statistically independent, equal variance variables x_i**
- **equal covariance matrices $\Sigma_i = \Sigma$**

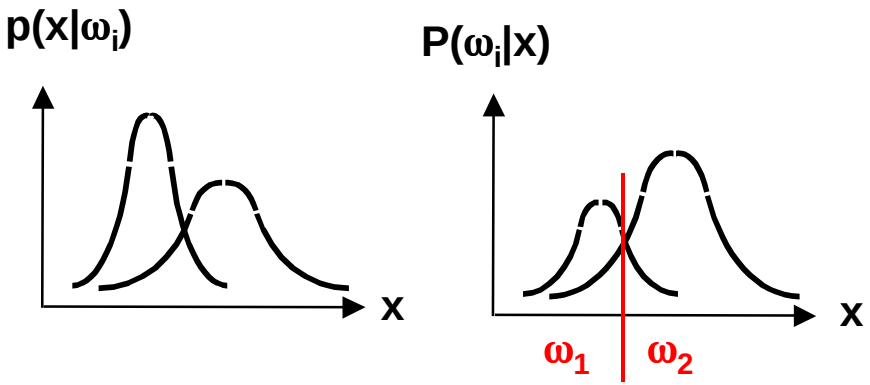
Univariate distribution

$p(x|\omega_i)$ are univariate Gaussian distributions.

Example: 2 classes ω_1 and ω_2

$$p(x|\omega_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right]$$

$$p(x|\omega_2) = \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right]$$



Decision rule:

$g_i(x) = \log P(\omega_i|x)$

$g_i(x) = -1/(2\sigma_i^2)\,(x - \mu_i)^2 - 1/2 \log \sigma_i + \log P(\omega_i)$

Statistically Independent, Equal Variance Variables

Because of sufficient statistical data, variables are sometimes assumed to be statistically independent and of equal variance.

$$\Sigma_i = \sigma^2 I$$

$$g_i(\underline{x}) = -1/(2\sigma^2) \, ||\underline{x} - \underline{\mu}_i||^2 + \log P(\omega_i)$$

If $P(\omega_i) = 1/N$, then the decision rule is equivalent to the minimum-distance classification rule.

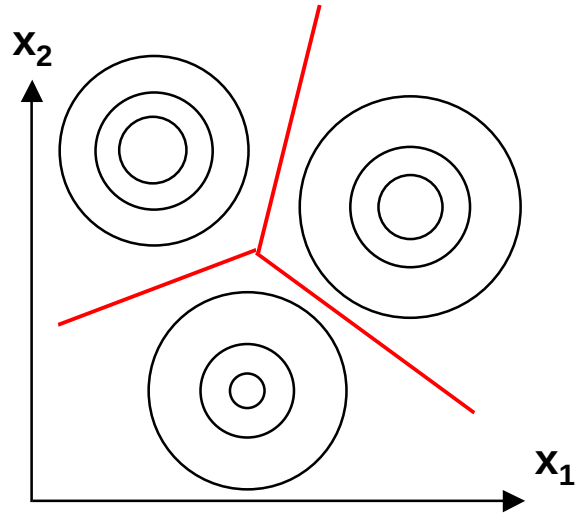
By expanding $g_i(\underline{x})$ and dropping the $\underline{x}^T\underline{x}$ term one gets the decision rule:

$$g_i(\underline{x}) = -1/(2\sigma^2)[-2\underline{\mu}_i^T\underline{x} + \underline{\mu}_i^T\underline{\mu}_i] + \log P(\omega_i)$$

which is linear in $\underline{x}$ and can be written

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

The decision surface is composed of hyperplanes.



Equal Covariance Matrices

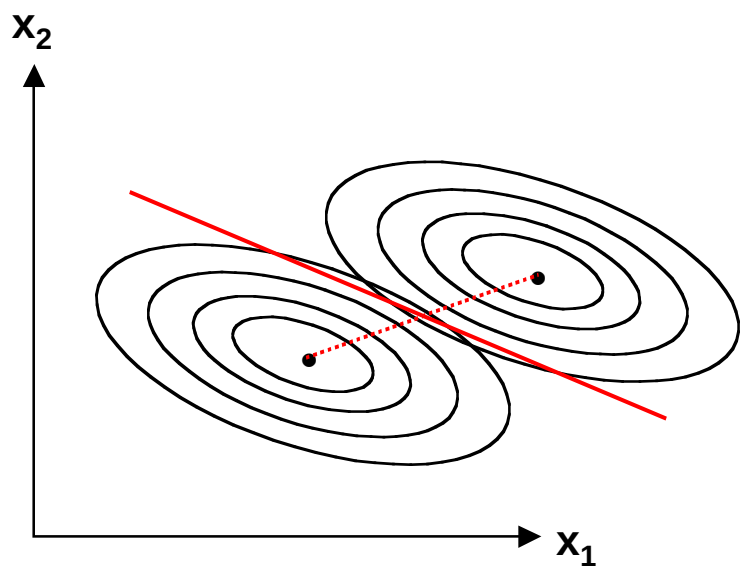
If $\Sigma_i = \Sigma$, the decision rule can be simplified:

$$g_i(\underline{x}) = -1/2\ (\underline{x} - \underline{\mu}_i)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_i) + \log P(\omega_i)$$

By expanding the quadratic form and dropping $\underline{x}^T \Sigma^{-1} \underline{x}$ one gets again a linear decision rule which can be written:

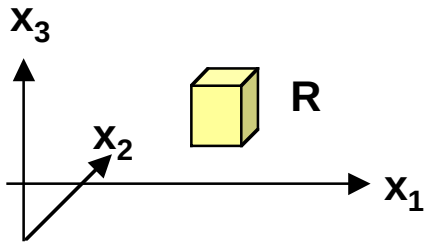
$$g_i(\underline{x}) = \underline{w}_i^T \underline{x} + w_{i0}$$

If the a-priori probabilities are equal, the decision rule assigns $\underline{x}$ to the class where the Mahalanobis distance to the mean $\underline{\mu}_i$ is minimal.



Estimating Probability Densities

Let R be a region in feature space with volume V.
Let k out of N samples lie in R.



$$\int_R p(\underline{x}')d\underline{x}' \approx \frac{k}{N} \approx p(\underline{x})V$$

$$p(\underline{x}) \approx \frac{k/N}{V}$$

relative frequency of samples per volume

A sequence of approximations $p_n(\underline{x})$ may be obtained by changing the volume V_n as the number of samples n increases.

Examples:

$V_n \sim 1/\sqrt{n}$ Parzen Windows

$k_n \sim \sqrt{n}$ adjust volume for k nearest neighbours

Conditions for a converging sequence of estimates $p_n(\underline{x})$:

1. $\lim_{n\to\infty} V_n = 0$
2. $\lim_{n\to\infty} k_n = \infty$
3. $\lim_{n\to\infty} \frac{k_n}{n} = 0$