Apprentissage et Reconnaissance des Formes

Machine Learning and Pattern Recognition

EP013M62

p.charbonnier@unistra.fr

pierre.charbonnier@cerema.fr

Telecom Physique Strasbourg - ISSD
MASTER IRIV – ID,
TOPO

Machine Learning and Pattern Recognition

- Introduction
- Statistical approaches
  - Bayesian decision theory
  - The Gaussian case / quadratic classifiers
  - Naive Bayes classification
- Probability density estimation
  - Parametric methods
  - Non-parametric methods
- Dimensionality reduction techniques
  - Feature selection / extraction
- Unsupervised classification
  - Mixture models and EM algorithm
  - Clustering methods
- Linear discriminant functions
- Neural approaches
  - Multilayer Perceptron
  - Radial Basis Functions
- Support Vector Machines
- Ensemble learning
- Evaluation
- Conclusion

Machine Learning and Pattern Recognition

p.charbonnier@unistra.fr

Telecom Physique Strasbourg - ISSD
MASTER IRIV – ID,
TOPO
2023-2024

Introduction

Pattern recognition – historical remarks – applications

A typical classification process – basic notions

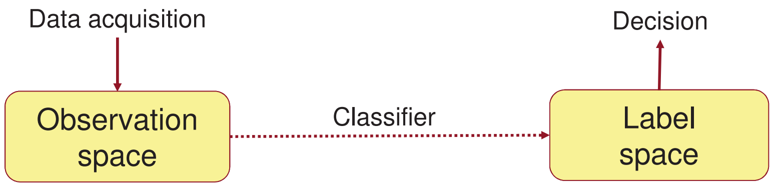
Main approaches of classification – an example

Outline of the course

Pattern Recognition (PR)

- **PR is a branch of Artificial Intelligence (AI)**
- **"The act of taking in raw data and making an action based on the "category" of the pattern" [Duda]**
- **In other words, the idea is:**
  - First, to characterize classes of patterns (machine learning)
  - Then to assign a label to the pattern (classification or recognition)

The classification task



Pattern Recognition (PR)

- **PR is a branch of Artificial Intelligence (AI)**
- **“The act of taking in raw data and making an action based on the “category” of the pattern” [Duda]**
- **In other words, the idea is:**
  - First, to characterize classes of patterns (machine learning)
  - Then to assign a label to the pattern (classification or recognition)
- **This might appear as trivial for a human being !**
  - A gift of the evolution process…
- **This is very complicated for a machine !**
  - Variability of patterns, observation noise and perturbation

Historically…

- **Quite a « long » story…**

- **Before the 1960’s: theoretical researches in the field of statistics**

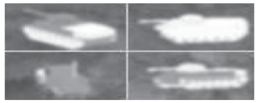
- **Development of computers: raising computational capacities more demands for practical applications**
- **Progress of signal/image processing, development of databases and the Internet: pattern recognition becomes one important element of a vaster signal/image/data interpretation process.**

  Keywords: *Big data, Deep Learning, Data Science…*

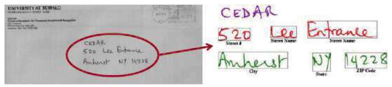
Some applications

[Gutierrez]

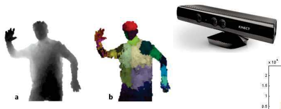
- **Machine vision**
  - Visual inspection
  - Target recognition



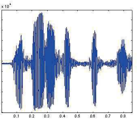
- **Character Recognition**
  - OCR (*Optical Character Recognition*)
  - Ex : addresses, bank checks…


CEDAR
520
Lee
Entrance
Amherst
NY
14228

- **Biometry, Computer Interface**
  - Fingerprints, faces…
  - Gesture recognition



[CRIMINISI 2011]



- **Signal analysis**
  - Computer aided diagnosis
  - Speech analysis

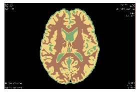


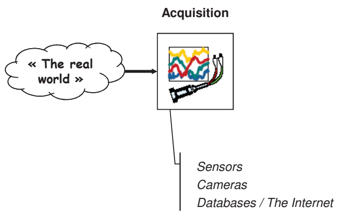
Image: F. Rousseau, iCube

- **Image database indexation**
- **Data mining, and many others…**

A typical classification system

[Gutierrez]

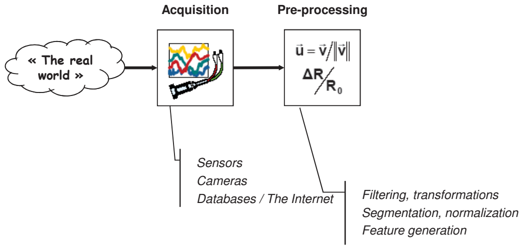
- **Data acquisition**
  - Take data in and prepare them for computerized processing
  - Data can be taken from various sources: sensors, databases…
  - We will focus on *images*



A typical classification system

[Gutierrez]

- **Pre-processing**
  - Suppress noise, useless information, normalize, re-sample, enhance contrast,…
  - Image processing (see course : ***Outils Fondamentaux en Traitement d'Images***)
- **Extract characteristics or features**
  - Make data representation compact and compatible with learning and decision tools
  - Feature engineering (this course) vs. feature learning (deep learning)



Sensors
Cameras
Databases / The Internet

Filtering, transformations
Segmentation, normalization
Feature generation

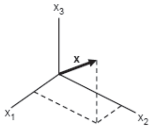
Characteristics or Features

- **A feature is an aspect, a property or a measure**
  - Symbolic, ex. color, « small », « medium », « regular »…
  - Numeric, e.g. dimension, gray level, signal magnitude…
- **A primitive**
  - Elementary component, non-decomposable of a shape e.g. a segment
- **Sample representation depends on the classification method…**
  - Organization of primitives (sequence, graph, composition rules)
  - Set of properties, logical description
  - Vector of characteristics or feature vector, of size d.
- **The feature vector, that will be denoted by x latter on,**
  - Is a concatenation of d numerical measurements.
  - Is one point in the d-dimensional feature space.

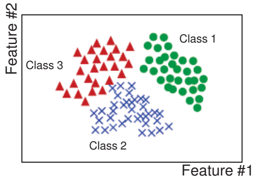
Characteristics or Features (continued)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Feature vector



Feature space (3D)



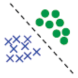
Scatter plot, 2D

[Gutierrez]

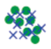
- **The feature space may be highly dimensional**
  - This increases algorithmic complexity
  - This makes learning more difficult (needs more sample to pave feature space)
- **One must then use dimensionality reduction techniques**

Characteristics or Features (finished)

- **A good feature vector must be discriminatory**
  - Samples from the same class must have similar values
  - Samples from different classes must have very different values

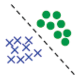


Good

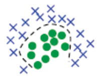


Bad

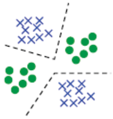
- **Other feature vector properties**



Linearly separable



Non-linearly separable



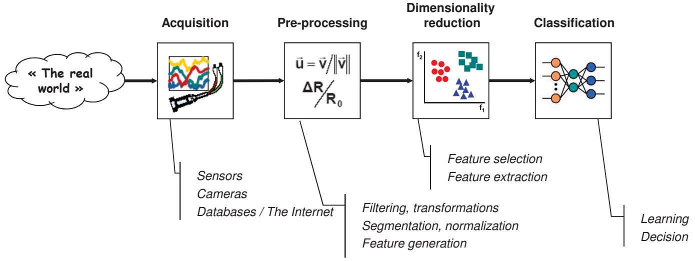
Multi-modal

[Gutierrez]

A typical classification system

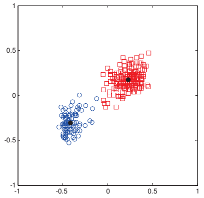
[Gutierrez]

- **Learning**
  - Using a *training sample*, build class representation that will lead to the best possible recognition performance
- **Classification**
  - Assign class label to a new pattern



Class representation

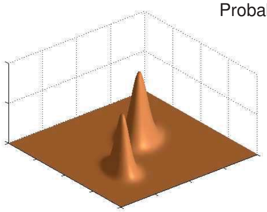
- **May be <u>parametric</u> or <u>non-parametric</u>, <u>deterministic</u> or <u>statistic</u>, e.g.**



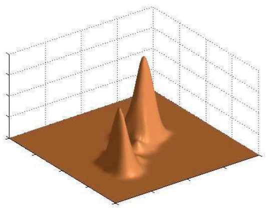
Prototypes (templates) = class means

$$\mu_i = \frac{1}{N}\sum_{x\in\omega_i} x$$

Probability Density Functions (PDF's) capture class variability



$$p(x|\omega_i) = f(x,\mu_i,\Sigma_i)$$

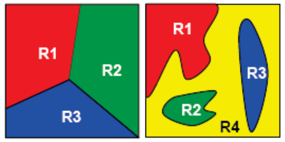


$$\mathrm{p}(\mathbf{x}|\omega_i) = \mathrm{f}\left(\mathbf{x},\left\{\mathrm{x}_{(1)},\mathrm{x}_{(2)},\dots,\mathrm{x}_{(N)}\right\}\right)$$

Classification

[Gutierrez]

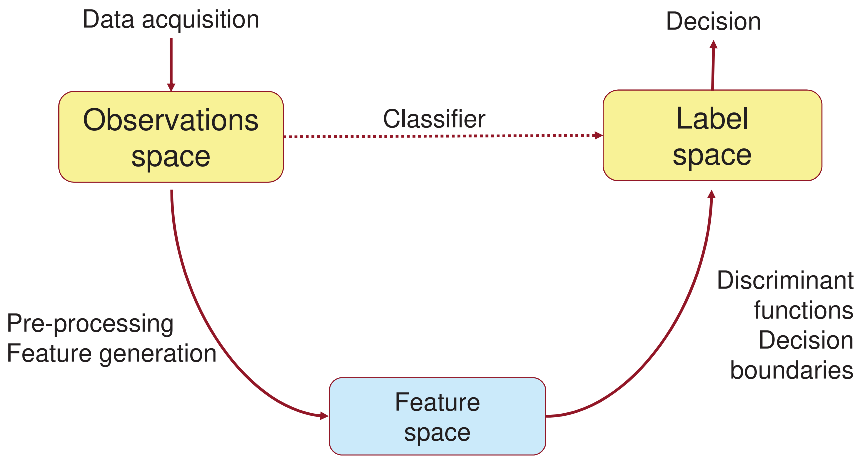
- **Group membership**
  - A classifier may rely on a set of <u>discriminant functions</u> $g_i(x)$,
  - One function per class (i=1…C)
  - Computes all $g_i(x)$ and assign x to the class that corresponds to the maximum
    - ✓ Exception: dichotomy (C=2). Decide according to the sign of g(x)=g1(x)-g2(x)
- **Partitioning of feature space**
  - One <u>decision region</u> per class
  - Boundaries between classes are called <u>decision boundaries</u>
  - Classification of x = determine which class it belongs to



Example in 2D

- **Max. discriminant functions => decision regions**
  - But decision functions may be set up directly

Features and the classification task



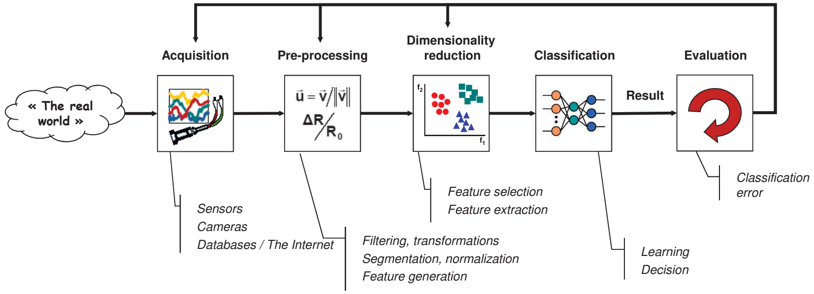
Learning: setting up the classifier

- **Supervised learning**
  - Needs a sample data set with labels $\{x, \omega_i\}$
  - Split the data sample into a training set and a validation/test set
- **Unsupervised learning**
  - When the sample data set comes without labels
  - It is necessary to group patterns according to some similarity measure
  - This is called clustering (one group = one cluster), or *coalescence* in French
- **Remark**
  - Clustering techniques can serve as data exploratory method, before a supervised learning is used.

A typical classification system

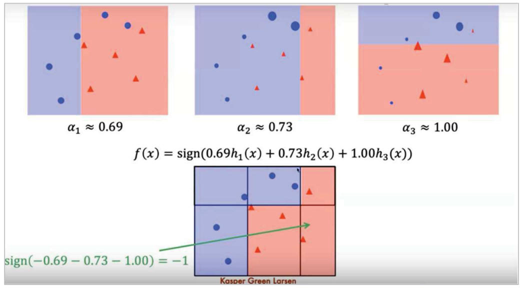
[Gutierrez]

- **Evaluation**
  - Process of assessing the performance of the system in order to identify possible weaknesses.



Remarks

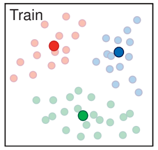
- **The typical classification chain is not always respected !**
  - Isolating a shape from its context (segmentation) is a classification problem (e.g. gray level thresholding == pixel classification)
  - Segmentation and recognition may be merged
- **Meta-classifiers combine weak classifiers to form strong ones (e.g. AdaBoost, random forests)**

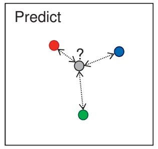

$\alpha_1 \approx 0.69$
$\alpha_2 \approx 0.73$
$\alpha_3 \approx 1.00$
$f(x) = \text{sign}(0.69h_1(x) + 0.73h_2(x) + 1.00h_3(x))$
$\text{sign}(-0.69 - 0.73 - 1.00) = -1$
Kasper Green Larsen

Source: https://www.youtube.com/watch?app=desktop&v=LhwnpMWIA90

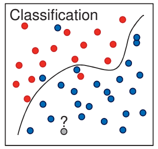
Remarks

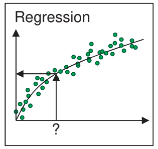
- **Template matching**
  - Particular case where each class is represented by a single individual (called prototype or *template*). Classification by maximizing some similarity measure





- **Regression**
  - Generalization of classification that uses a functional description of data to predict a continuous numerical value from an entry



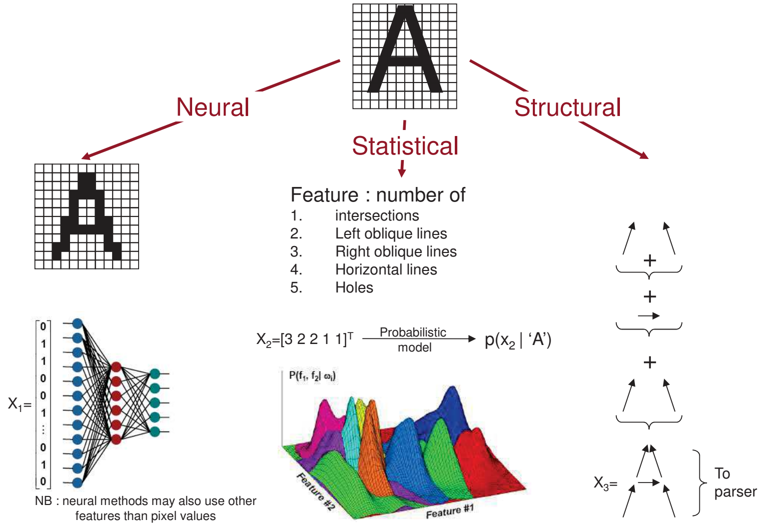


The three main approaches of classification

- **Statistical**
  - Uses statistical properties such as probability density functions (PDF), *a priori* probabilities P(ω_i) and the notion of cost to define decision boundaries.
  - E.g. class-conditional probabilities p(x|ω_i) = probability of x given the class label, ω_i. A natural way of encoding variability !
- **Neural (connectionist)**
  - Conceptually different approach: neuro-mimetic or connectionist. Can be seen as a kind of statistical approach in certain cases, anyway.
  - Classification = response of a network of connected elementary computation unit (the so-called "neurons"). "Knowledge" = way the neurons are connected and/or synaptic weight of each connection.
  - Does not need any a priori knowledge. Can handle arbitrarily complex decision boundaries. But some "black-box" flavor.
- **Syntactical (structural)**
  - Uses "non-metric" data and logical rules.
  - Classification = structural similarity measure. "Knowledge" = grammar rules or graphs.

Main classification approaches

from [Schalkoff]



Feature : number of
1. intersections
2. Left oblique lines
3. Right oblique lines
4. Horizontal lines
5. Holes

$X_2=[3\ 2\ 2\ 1\ 1]^T \xrightarrow{\text{Probabilistic model}} p(x_2 \mid \text{'A'})$

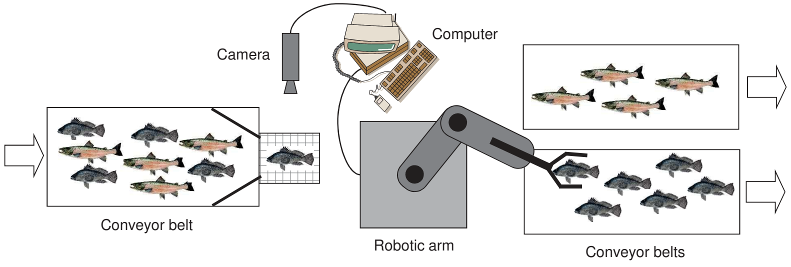
NB : neural methods may also use other features than pixel values

To parser

An example

[Duda]

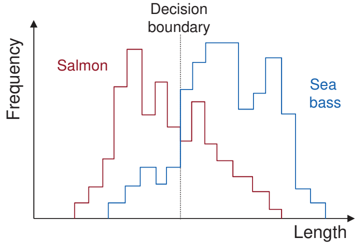
- **Automating a fish-packing plant**
  - Separate incoming fish: salmons from sea-bass
  - Tools: a camera, a computer, a robotic arm



- **Sample images ⇨ differences between species**
  - Size, color, number and shape of fins, position of the mouth…
  - Possible *features* for our *classifier*.

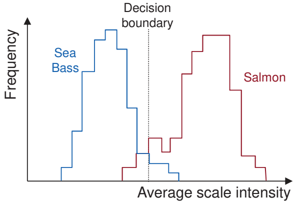
A first prototype

- **Image capture**
- **Pre-processing: image analysis**
  - Intensity level adjustment,
  - Segmentation to separate fishes from background, alignment.
- **Feature extraction**
  - Sea basses are, on average, larger than salmons
  - Estimate length of fishes from segmented images
- **Classification**
  - A decision threshold is determined from the histograms of fish lengths over a training set.
  - A fish is classified according to its length
  - 40 % classification error !

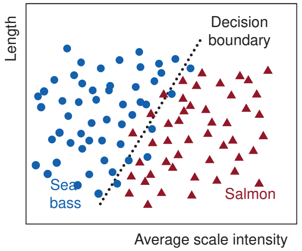


Improvement

- **Use different feature**
  - E.g. Average scale intensity
  - Better classification rate using the same method

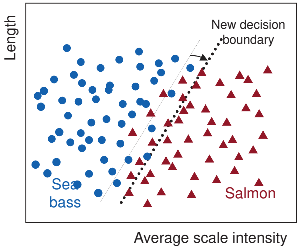


- **Combine features to improve class separability**
  - Compute a *linear discriminant function* to partition feature space.
  - Obtained classification rate: 95%
  - Is it worth it to incorporate more features ? If yes, to which point ? This is the so-called dimensionality issue.



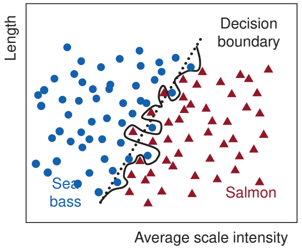
The notion of decision cost

- **So far, we tried to minimize classification error**
  - Is this symmetric cost the best figure-of-merit ?
  - i.e. are the consequences of our actions equally costly ?
  - This is the notion of decision cost
    - ✓ A consumer will be upset to find sea-bass in a can labeled "salmon"
    - ✓ A consumer will easily accept to find tasty salmon in a can labeled "sea bass"
- **To stay in the business**
  - We should adjust the decision boundary to avoid misclassifying sea bass ! (reduce the number of sea basses classified as salmons)
  - Minimize decision cost instead of classification error...



The issue of generalization

- **Suppose that incorporating more features provides too little improvement to our system.**
- **We might try to set up a much more complex decision boundary.**
  - The linear boundary is replaced by a complex one, which perfectly separates training patterns.
  - This is not a good idea ! What classification is about is to suggest actions when presented with *novel* patterns !
  - This is called the *generalization* issue.



Outline of the course

- **The ideal case: all statistical properties known**
  - Bayesian decision theory
- **Given a set of training data with class labels**
  - What can we do when only the parametric form of the PDFs is known ?
    - ✓ Parametric probability density estimation
  - What can we do when nothing is known about the PDFs ?
    - ✓ Non-parametric probability density estimation
  - How can we select features and/or reduce dimensionality ?
    - ✓ Feature extraction an selection
- **When the labels of training samples are unknown**
  - Unsupervised classification
- **Back to the supervised case**
  - Estimate linear discriminant functions when the PDFs are unknown
  - Non-linear extensions: neural networks and Support Vector Machines
- **Combining classifiers**
  - Ensemble learning
- **Evaluation**

Bayesian theory and classification

Minimizing the probability of classification error

Minimizing the Bayesian risk

Bayesian classification and discriminant functions

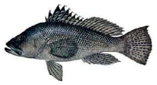
Bayesian classification and normal distributions

Summary of episode 1

- **PR: "The act of taking in raw data and making an action based on the "category" of the pattern" [Duda]**
  - Extract characteristics or features
  - Characterize classes (learning) → discriminant functions
  - Make a decision: perform classification
- **Three main approaches (among which we study the first 2)**
  - Statistical (natural for encoding variability, account for noise)
  - Connectionist or neural
  - Syntactical or structural
- **Let us consider now the ideal case (all probabilities known)**

Introduction

- **Let us consider our "fish-packing plant" example [Duda]…**
  - This is a C=2 classes problem (or dichotomy)
- **What kind of fish will appear next ?**



Sea bass ?

…or salmon ?

- **Let ω be the random variable that denotes the *state of nature***
- **with $\omega = \omega 1$: sea bass, $\omega = \omega 2$: salmon**

Notations and hypotheses

- **The *prior* probabilities, $P(\omega_i)$, i = 1…C supposed to be known**

$$\sum_{i=1}^{C} P(\omega_i) = 1$$

  - If not, we may estimate them from a sample set
  - If we have N samples, N_1 from class ω_1 and N_2 from class ω_2,

$$P(\omega_1) \approx N_1 / N \quad \text{and} \quad P(\omega_2) \approx N_2 / N$$

- **We note x the *feature vector***
  - x is a continuous random variable whose distribution depends on the state of nature
- **The *class-conditional* probability density functions, $p(x \mid \omega_i)$, i = 1, 2 are also known**

$$\int_{-\infty}^{+\infty} p(x|\omega_i)dx = 1$$

  - They are also called *likelihood* (of x for class ω_i)
  - We may also estimate them if necessary (supervised learning, see next chap.)

The MAP decision rule

- **Bayes theorem ⇨ *posterior* probability P(ω$_i$ | x)** $\sum_{i=1}^{C} P(\omega_i|x) = 1$
  - Probability that the state of nature is ω$_i$ given observation x

- **Intuitive decision (or classification) rule:**

« Choose the most probable state given an observed feature vector, x »

$$\begin{cases} \text{if} \quad P(\omega_1|x) > P(\omega_2|x) \quad \text{choose} \quad \omega_1 \\ \text{if} \quad P(\omega_1|x) < P(\omega_2|x) \quad \text{choose} \quad \omega_2 \end{cases}$$

  - We choose the class whose *a posteriori* probability is maximal (hence the name of the rule: MAP=maximum *a posteriori*)
  - In a more compact form :

$$\boxed{P(\omega_1|x) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} P(\omega_2|x)}$$

Likelihood test

- **Using Bayes theorem:** $\boxed{P(\omega_i|x) = \dfrac{p(x|\omega_i).P(\omega_i)}{p(x)}}$

$p(x) = \sum_{i=1}^{C} p(x|\omega_i)P(\omega_i)$

- **The decision rule becomes** $\dfrac{p(x|\omega_1).P(\omega_1)}{p(x)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \dfrac{p(x|\omega_2).P(\omega_2)}{p(x)}$

- **Since *p(x)* does not affect the decision rule**

$$\boxed{\Lambda(x) = \frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)}}$$

- **where Λ(x) is a *likelihood ratio*.**
  This kind of decision rule is also called a *likelihood test*.

Interpretation

- **If, moreover, the *a priori* probabilities are equal (to ½):**

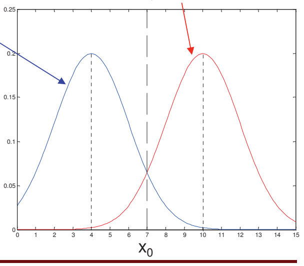
$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 1$$

  - The choice only depends on class-conditional PDFs: Maximum Likelihood (ML) test

- **Example (equal *priors*):**

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 1 \longrightarrow x \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} x_0 = 7$$

$p(x|\omega_1) = \dfrac{1}{2\sqrt{2\pi}} \exp\left(-(x-4)^2/8\right)$ $p(x|\omega2) = \dfrac{1}{2\sqrt{2\pi}} \exp\left(-(x-10)^2/8\right)$



- **In any case, one can make decision errors !**

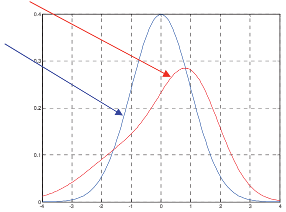
➔ How can we optimize x$_0$ in order to minimize the Probability of error ?

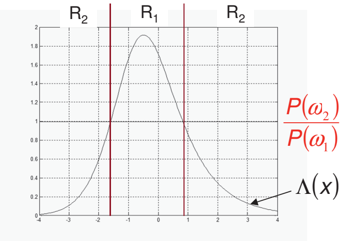
Remark

- **The decision regions R$_i$ may not be simply connected**
  - Example (from [Webb])
  - *Equal priors*

$p(x|\omega_2) = 0{,}6.\mathcal{N}(1{,}1) + 0{,}4.\mathcal{N}(-1{,}2)$

$p(x|\omega_1) = \mathcal{N}(0{,}1)$





- **We may have several *decision boundaries***
- **Problem: how can we optimize these boundaries to minimize classification error ?**

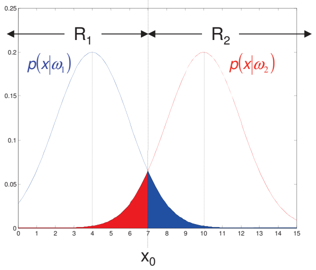
Minimizing the probability of error (1/2)

- **The probability of classification error is:**

$$P_e = P(x\in R_2,\omega_1) + P(x\in R_1,\omega_2)$$

$$= \underbrace{P(x\in R_2|\omega_1)}.P(\omega_1) + \underbrace{P(x\in R_1|\omega_2)}.P(\omega_2)$$

$$= \boxed{\int_{R_2} p(x|\omega_1)dx}.P(\omega_1) + \boxed{\int_{R_1} p(x|\omega_2)dx}.P(\omega_2)$$



- **But, by definition:** $\int_{R_1} p(x|\omega_i)dx + \int_{R_2} p(x|\omega_i)dx = \int_{R_1\cup R_2} p(x|\omega_i)dx = 1$

- **So:**

$$P_e = P(\omega_1).\left(1 - \int_{R_1} p(x|\omega_1)dx\right) + P(\omega_2)\int_{R_1} p(x|\omega_2)dx$$

Minimizing the probability of error (2/2)

- **We have just written P_e as a function of R_1 only:**

$$P_e = P(\omega_1) - \int_{R_1} p(x|\omega_1).P(\omega_1)dx + \int_{R_1} p(x|\omega_2).P(\omega_2)dx$$

- **Using Bayes formula:** $P(\omega_i|x).p(x) = p(x|\omega_i).P(\omega_i)$

$$P_e = P(\omega_1) - \int_{R_1} P(\omega_1|x).p(x)dx + \int_{R_1} P(\omega_2|x).p(x)dx$$

$$= P(\omega_1) - \int_{R_1} [P(\omega_1|x) - P(\omega_2|x)].p(x)dx$$

- **The error is minimized if R_1 is the locus of points x such that:**

$$g(x) = P(\omega_1|x) - P(\omega_2|x) > 0 \quad \Rightarrow \quad \boxed{P(\omega_1|x) > P(\omega_2|x)}$$

The MAP decision rule minimizes the probability of classification error.
This error P_e is also called *Bayesian error rate.*

The notion of Bayesian risk

- **All errors are not equal in terms of practical impact!**
  - Finding tasty salmon in a cheap fish can labeled “Sea Bass” is a better surprise than finding sea bass in a costly “Salmon” can (both being due to problems with our automatic fish-packing plant)
  - It is a more serious problem to say that a patient suffering from cancer is healthy than to diagnose a cancer to a healthy patient (both being diagnostic errors)
- **Let us suppose that a feature vector x from ω_k appears in R_i.**
  - Then x is assigned to class ω_i
  - This is a classification error iff k ≠ i
  - We can put a weight (cost, or *loss*) on our decision: λ_{ki}
- **The cost matrix, or *loss matrix* is defined as $L = [\lambda_{ki}]_{k,i=1,2}$**

Conditional risk

- **λ_{ki} = cost of assigning x to class ω_i while it belongs to ω_k**
- **What is the cost of assigning x to ω_i?**

  → λ_{1i} if x belongs in fact to class ω_1

  → λ_{2i} if x belongs in fact to class ω_2
- **The *conditional risk* is the risk of assigning x to class ω_i, i.e. the average (over *k*) of costs**
  - knowing that the probability that the “state of nature” is in fact ω_k is $P(\omega_k|x)$

$$\boxed{r_i(x) = \sum_{k=1}^{2} \lambda_{ki} P(\omega_k|x)}$$

- **Note that, in this formalism, good decisions also have a cost !**

Average risk

- **The average of risk over region R_i is:**

$$\bar{r}_i = \int_{R_i} \sum_{k=1}^{2} \lambda_{ki} P(\omega_k|x) p(x) dx$$

- **And the overall risk (over the whole set of regions)**

$$r = \sum_{i=1}^{2} \bar{r}_i = \sum_{i=1}^{2} \int_{R_i} \sum_{k=1}^{2} \lambda_{ki} P(\omega_k|x).p(x) dx = \sum_{i=1}^{2} \sum_{k=1}^{2} \lambda_{ki}.P(\omega_k).\int_{R_i} p(x|\omega_k) dx$$

- **Special case: “0 – 1” loss function**

$$\lambda_{ki} = 1 - \delta_{ki} = \begin{cases} 0 & if \quad k = i \\ 1 & if \quad k \neq i \end{cases}$$

➔ **Leads to** $r = P_e = \int_{R_2} p(x|\omega_1) dx.P(\omega_1) + \int_{R_1} p(x|\omega_2) dx.P(\omega_2)$

Minimizing the overall risk (1/2)

- **Let us develop** $r = \sum_{i=1}^{2} \sum_{k=1}^{2} \lambda_{ki}.P(\omega_k).\int_{R_i} p(x|\omega_k) dx$

- **We obtain:**

$$\begin{aligned} r = {} & \lambda_{11}.P(\omega_1).\int_{R_1} p(x|\omega_1) dx + \lambda_{21}.P(\omega_2).\int_{R_1} p(x|\omega_2) dx \\ & + \lambda_{12}.P(\omega_1).\int_{R_2} p(x|\omega_1) dx + \lambda_{22}.P(\omega_2).\int_{R_2} p(x|\omega_2) dx \\ & + \lambda_{12}.P(\omega_1).\int_{R_1} p(x|\omega_1) dx + \lambda_{22}.P(\omega_2).\int_{R_1} p(x|\omega_2) dx \\ & - \lambda_{12}.P(\omega_1).\int_{R_1} p(x|\omega_1) dx - \lambda_{22}.P(\omega_2).\int_{R_1} p(x|\omega_2) dx \end{aligned}$$

- **Which simplifies !**

$$\begin{aligned} r = {} & [\lambda_{21} - \lambda_{22}].P(\omega_2).\int_{R_1} p(x|\omega_2) dx - [\lambda_{12} - \lambda_{11}].P(\omega_1).\int_{R_1} p(x|\omega_1) dx \\ & + \lambda_{12}.P(\omega_1) + \lambda_{22} P(\omega_2) \end{aligned}$$

Constants !

Minimizing the overall risk (2/2)

- **The remaining expression:**

$$\int_{R_1} \{ [\lambda_{21} - \lambda_{22}].p(x|\omega_2).P(\omega_2) - [\lambda_{12} - \lambda_{11}].p(x|\omega_1).P(\omega_1) \} dx$$

or

$$\int_{R_1} \{ [\lambda_{21} - \lambda_{22}].P(\omega_2|x) - [\lambda_{12} - \lambda_{11}].P(\omega_1|x) \} p(x) dx$$

- **…is minimized if R_1 is the locus of points x such that:**

$$[\lambda_{12} - \lambda_{11}].P(\omega_1|x) > [\lambda_{21} - \lambda_{22}].P(\omega_2|x)$$

  - This is a **weighted** Bayesian decision rule

- **Or, under the form of a likelihood test:**

$$\boxed{\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}}$$

Recall that λ_{ki} = cost of assigning x to class ω_i while x belongs to ω_k

Exercise 1

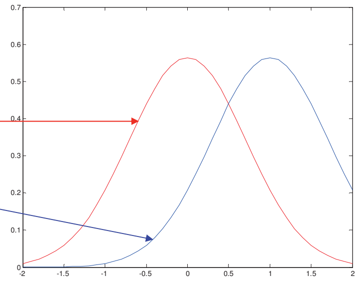
- **1-dimensional problem, 2 classes with equal priors**
- **Class-conditional probabilities:** $\mathcal{N}(0, ½)$ **and** $\mathcal{N}(1, ½)$
- **Calculate:**
  - The threshold x_0 that minimizes the probability of classification error
  - The threshold x_1 that minimizes the overall risk when the loss matrix is:

$$L = \begin{pmatrix} 0 & 0{,}5 \\ 1 & 0 \end{pmatrix}$$

$$p(x|\omega_1) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$$

$$p(x|\omega 2) = \frac{1}{\sqrt{\pi}} \exp(-(x-1)^2)$$

$$NB: \frac{\ln(2)}{2} \approx 0.35$$



- **Same exercise with 0-1 losses, and** $P(\omega_1) = 2/3$, $P(\omega_2) = 1/3$

Summary of decision rules (1/2)

- **Minimizing the overall risk allows**
  - Introducing supplementary *prior* information using decision costs (or losses)
  - Obtaining the most general form of decision rule

$$\Lambda(x)=\frac{p(x|\omega_1)}{p(x|\omega_2)}\underset{\omega_2}{\overset{\omega_1}{\gtrless}}\frac{\lambda_{21}-\lambda_{22}}{\lambda_{12}-\lambda_{11}}\frac{P(\omega_2)}{P(\omega_1)}\Leftrightarrow\frac{P(\omega_1|x)}{P(\omega_2|x)}\underset{\omega_2}{\overset{\omega_1}{\gtrless}}\frac{\lambda_{21}-\lambda_{22}}{\lambda_{12}-\lambda_{11}}$$

Bayes rule

- **Minimizing the classification error**
  - Is a particular case of Bayes rule based on “0-1” loss function
  - Amounts to maximizing the *posterior* distribution, P(ω_i | x)

$$\lambda_{ki}=\begin{cases}0 & k=i\\ 1 & k\neq i\end{cases}\qquad\Lambda(x)=\frac{p(x|\omega_1)}{p(x|\omega_2)}\underset{\omega_2}{\overset{\omega_1}{\gtrless}}\frac{P(\omega_2)}{P(\omega_1)}\Leftrightarrow\frac{P(\omega_1|x)}{P(\omega_2|x)}\underset{\omega_2}{\overset{\omega_1}{\gtrless}}1$$

MAP rule

Summary of decision rules(2/2)

- **The particular case where**
  - The “0 – 1” loss function is used and the priors are equal leads to maximizing the likelihood, p(x | ω_i)

$$\lambda_{ki}=\begin{cases}0 & k=i\\ 1 & k\neq i\end{cases}\quad P(\omega_i)=\frac{1}{C}\;\forall i\quad\Lambda(x)=\frac{p(x|\omega_1)}{p(x|\omega_2)}\underset{\omega_2}{\overset{\omega_1}{\gtrless}}1$$

ML rule

- **Other decision rules exist, see e.g. [Webb]…**
  - The minimax criterion
    - ✓ Does not need knowledge of *prior* probabilities.
  - The Neyman-Pearson decision rule
    - ✓ Fixes the probability of classification error for one class and seeks to minimize the probability of classification error for the other one
    - ✓ Does not need neither knowing *prior* probabilities, nor losses.

Structure of decision rules

- **Bayes rule**

$$\left[-\sum_{k=1}^{2}\lambda_{k1}.P(\omega_k|x)\right]-\left[-\sum_{k=1}^{2}\lambda_{k2}.P(\omega_k|x)\right]\underset{\omega_2}{\overset{\omega_1}{\gtrless}}0$$

$$(-r_1)-(-r_2)\underset{\omega_2}{\overset{\omega_1}{\gtrless}}0$$

- **MAP rule**

$$P(\omega_1|x)-P(\omega_2|x)\underset{\omega_2}{\overset{\omega_1}{\gtrless}}0$$

- **ML rule**

$$p(x|\omega_1)-p(x|\omega_2)\underset{\omega_2}{\overset{\omega_1}{\gtrless}}0$$

➔ are all of the form $g(x)=g_1(x)-g_2(x)\underset{\omega_2}{\overset{\omega_1}{\gtrless}}0$ **or** $g_1(x)\underset{\omega_2}{\overset{\omega_1}{\gtrless}}g_2(x)$

Generalization to C classes

- **The Bayes classification rule minimizes the overall risk**

$$r=\sum_{i=1}^{C}\int_{R_i}\sum_{k=1}^{C}\lambda_{ki}P(\omega_k|x).p(x)dx$$

  - For each x, choose class ω_i such that $g_i(x)$ is maximal, where

$$g_i(x)=-\sum_{k=1}^{C}\lambda_{ki}.P(\omega_k|x)=-r_i$$

- **The MAP rule minimizes the probability of classification error**

$$P_e=1-\sum_{i=1}^{C}\int_{R_i}P(\omega_i|x).p(x)dx$$

  - For each x, maximize

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

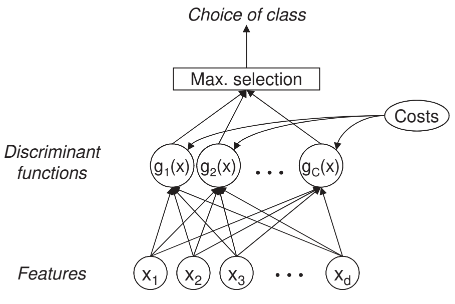
Discriminant Functions

- **A classification algorithm may be interpreted as a machine capable of evaluating a set of C discriminant functions, g_i and to select the category that corresponds to the highest obtained value**

Assign x to class ω_i iff
$g_i(x) > g_j(x) \quad \forall j \neq i$

- **Remark**

The choice of a discriminant function is not unique. Any strictly increasing function of a discriminant function is also a discriminant function



from [Duda]

Bayesian decision and discriminant functions

- **Bayesian decision rules naturally "fit" to the formalism of discriminant functions**

Rules	Discriminant funct.	Minimize
Bayes	$g_i(x) = -\sum_{k=1}^{C} \lambda_{ki}.P(\omega_k \vert x) = -r_i(x)$	Overall risk
MAP	$g_i(x) = P(\omega_i \vert x)$	Prob. of error
ML	$g_i(x) = p(x \vert \omega_i)$	

- **However, this approach dos not directly apply to all classification problems**
  - PDFs might be complicated and/or difficult to estimate
  - In some cases, it might be preferable to estimate decision boundaries directly, using other forms of cost functions, as we will see later.

- **In the Gaussian case, discriminant functions can be very simple…**

Bayesian classifiers & normal distributions

- **Recall (1)**
  - Any strictly increasing function of a discriminant function is also a discriminant function, in particular :

$$g_i(x) = \ln p(x|\omega_i) + \ln P(\omega_i) \quad - \ln p(x)$$

is a discriminant function

- **Recall (2) d-dimensional PDF normal distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$**

$$p(x|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right)$$

- **Gaussian MAP discriminant function**

$$\boxed{g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + \ln P(\omega_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{d}{2}\ln(2\pi)}$$

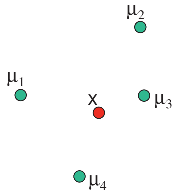
Case 1: $\Sigma_i = \sigma^2 I$

- **Occurs when features are independent, with equal variances**
  - The expression of the discriminant simplifies:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + \ln P(\omega_i) \underbrace{- \frac{1}{2}\ln|\Sigma_i| - \frac{d}{2}\ln(2\pi)}_{\downarrow}$$

$$g_i(x) = -\frac{1}{2\sigma^2}\underbrace{(x-\mu_i)^T (x-\mu_i)}_{\text{Quadratic distance to class mean}} + \ln P(\omega_i) \qquad + c_i$$

- **Particular case: equal priors**
  - (Euclidean) Minimum-distance classifier or nearest-mean classifier
  - Loci of constant distance are hyper-spheres



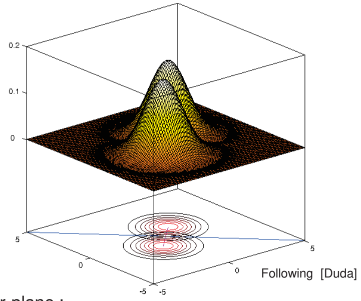
Case 1: discriminant function ($C = 2$)

- **Let us develop the expression of the discriminant function**

$$g_i(x) = -\frac{1}{2\sigma^2}\left(x^T x - 2\mu_i^T x + \mu_i^T \mu_i \right) + \ln P(\omega_i)$$

- Up to a constant (from the point of view of decision), we have:

$$g_i(x) = w_i^T x + w_{i0} \qquad \begin{cases} w_i = \dfrac{\mu_i}{\sigma^2} \\ w_{i0} = -\dfrac{1}{2\sigma^2}\mu_i^T\mu_i + \ln P(\omega_i) \end{cases}$$



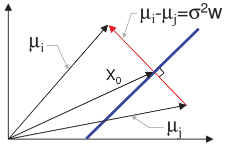
Following [Duda]

- The decision surface between ω_i and ω_j is a hyper-plane :

$$g_{ij}(x) \equiv g_i(x) - g_j(x) = w^T(x - x_0) = 0$$

$$\begin{cases} w = (\mu_i - \mu_j)/\sigma^2 \\ x_0 = \dfrac{1}{2}(\mu_i + \mu_j) - \sigma^2 \ln\left[\dfrac{P(\omega_i)}{P(\omega_j)}\right]\dfrac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2} \end{cases}$$

⇦ **(Check it !)**

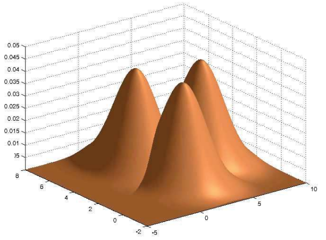


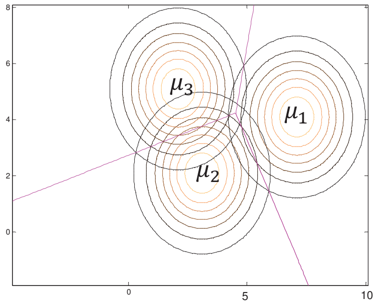
Following [Theodoridis]

Example, for 3 classes

$$\mu_1 = \begin{pmatrix}3\\2\end{pmatrix} \qquad \mu_2 = \begin{pmatrix}7\\4\end{pmatrix} \qquad \mu_3 = \begin{pmatrix}2\\5\end{pmatrix}$$

$$S_1 = S_2 = S_3 = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}$$





NB : classes with equal priors

$$\begin{cases} w = (\mu_i - \mu_j)/2 \\ x_0 = \dfrac{1}{2}(\mu_i + \mu_j) \end{cases}$$

following [Gutierrez]

Case 2: $\Sigma_i = \Sigma$

- **In this case:**
  - classes may be seen as ellipsoids with identical sizes and orientations, centered on the means, μ_i.
- **Why ?**
  - Surfaces of constant probability are such that: $(\mathbf{x} - \boldsymbol{\mu_i})^T\Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu_i}) = cte = c^2$
  - Since Σ is symmetrical and positive definite, it may be diagonalized by a unitary, orthogonal transformation (i.e. a rotation) : $\Sigma = \Phi\Lambda\Phi^T$ with $\Phi^T\Phi = \mathrm{I_d}$ and $\Lambda = diag\{\lambda_i\}_{i=1\ldots d}$
  - So : $(\mathbf{x} - \boldsymbol{\mu_i})^T\Phi\Lambda^{-1}\Phi^T(\mathbf{x} - \boldsymbol{\mu_i}) = c^2$
  - Taking $\mathbf{x}' = \Phi^T(\mathbf{x} - \boldsymbol{\mu_i})$, one obtains $\mathbf{x}'^T\Lambda^{-1}\mathbf{x}' = c^2$
  - Which is the equation of a hyper-ellipsoid $\boxed{\dfrac{x_1'^2}{\lambda_1} + \cdots + \dfrac{x_d'^2}{\lambda_d} = c^2}$

Cas 2: discriminant functions

- **The expression of the discriminant simplifies**

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i) + \ln P(\omega_i) \underbrace{- \frac{1}{2}\ln|\Sigma_i| - \frac{d}{2}\ln(2\pi)}_{+c_i}$$

$$g_i(x) = -\underbrace{\|x - \mu_i\|_{\Sigma^{-1}}^2}_{\textbf{Mahalanobis}\text{ distance}} + \ln P(\omega_i)$$

- **Particular case with classes of equal probabilities**
  - Mahalanobis minimum-distance classifier or nearest-mean classifier
- **Ignoring the $x^T\Sigma^{-1}x$ term (which is identical for all classes), the g_i's are linear**

$$g_i(x) = w_i^T x + w_{i0} \qquad \begin{cases} w_i = \Sigma^{-1}\mu_i \\ w_{i0} = -\dfrac{1}{2}\mu_i^T\Sigma^{-1}\mu_i + \ln P(\omega_i) \end{cases}$$

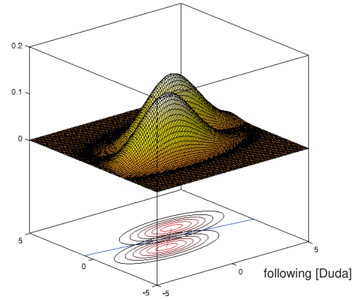
Case 2: separating hyper-plane ($C = 2$)

- **The decision surface between classes ω_i and ω_j is still a hyper-plane:**

$$g_{ij}(x) \equiv g_i(x) - g_j(x) = w^T(x - x_0) = 0$$

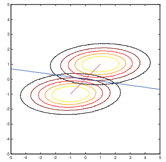
where

$$\begin{cases} w = \Sigma^{-1}(\mu_i - \mu_j) \\ x_0 = \frac{1}{2}(\mu_i + \mu_j) - \ln \frac{P(\omega_i)}{P(\omega_j)} \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2_{\Sigma^{-1}}} \end{cases}$$



following [Duda]

- **Beware:**
  - In this case, the separating hyper-plane may not be orthogonal to the segment connecting means...



- **When the priors are not equal**
  - Here again, the hyper-plane moves away from the mean of the most probable class

General case: $\Sigma_i \neq \Sigma_j$

- **The only term that can be discarded in the general equation:**

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + \ln P(\omega_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{d}{2}\ln(2\pi)$$

...is the last one.

- **The g_i functions remain quadratic**

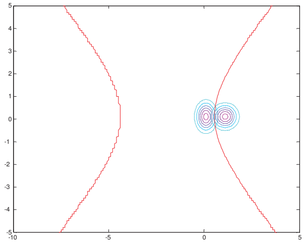
$$g_i(x) = x^T W_i x + w_i^T x + w_{i0} \qquad \begin{cases} W_i = -\frac{1}{2}\Sigma_i^{-1} \\ w_i = \Sigma_i^{-1}\mu_i \\ w_{i0} = -\frac{1}{2}\mu_i^T \Sigma_i^{-1}\mu_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i) \end{cases}$$

- **Decision surfaces, for 2 classes:**
  - hyper-planes, -spheres, -ellipsoids, -paraboloids, -hyperboloids...

Examples *following [Theodoridis]*

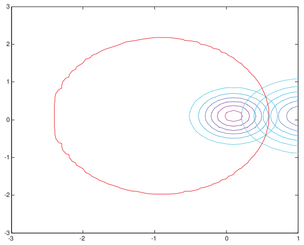
$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.15 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.15 & 0 \\ 0 & 0.1 \end{bmatrix}$$



$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.15 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.25 \end{bmatrix}$$



Examples *from [Duda]*

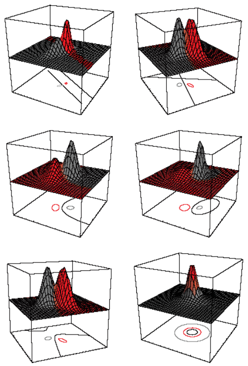


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

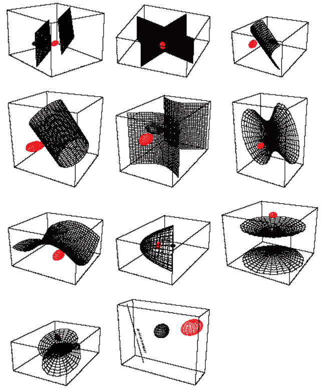


FIGURE 2.15. Arbitrary three-dimensional Gaussian distributions yield Bayes decision boundaries that are two-dimensional hyperquadrics. There are even degenerate cases in which the decision boundary is a line. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Take-away remarks…

- **When all probabilities are known (class-conditional PDF's + priors)**
  - The Bayesian framework allows the definition of optimal classifiers, in the sense of a certain cost function, e.g. classification error or overall risk.
  - These classifiers are implemented as likelihood tests in the 2-class case (dichotomy)
  - In the general, C-class case, a discriminant function has to be maximized.
- **Bayesian classifiers for Gaussian classes are**
  - Quadratic in general,
  - Linear when the covariance matrices are identical for all classes.
- **The linear minimal-distance classifier is**
  - Optimal for classes with equal priors, with identical Gaussian distributions.
  - Associated to the Euclidean distance when covariance matrices are diagonal, to the Mahalanobis distance, elsewhere.
- **These very well-known classifiers are obtained from the principles of Bayesian decision theory under mild simplifying hypotheses...**

Supplementary material

Discriminant function for Gaussian classifier with $\Sigma_i = \sigma^2 I$

- **The proof uses the following property**

$$(a-b)^T(a+b) = a^T a - b^T b$$

- **The discriminant is**

$$g_{ij}(x) = g_i(x) - g_j(x) = -\frac{1}{2\sigma^2}\left(-2\mu_i^T x + \mu_i^T\mu_i\right) + \ln P(\omega_i) + \frac{1}{2\sigma^2}\left(-2\mu_j^T x + \mu_j^T\mu_j\right) - \ln P(\omega_j)$$

$$g_{ij}(x) = \frac{1}{\sigma^2}(\mu_i - \mu_j)^T x - \frac{1}{2\sigma^2}\left(\mu_i^T\mu_i - \mu_j^T\mu_j\right) + \frac{\sigma^2}{\sigma^2}\ln\frac{P(\omega_i)}{P(\omega_j)}\frac{(\mu_i-\mu_j)^T(\mu_i-\mu_j)}{\|\mu_i-\mu_j\|^2}$$

- **Using the above equality**

$$g_{ij}(x) = \frac{1}{\sigma^2}(\mu_i - \mu_j)^T x - \frac{(\mu_i-\mu_j)^T(\mu_i+\mu_j)}{2\sigma^2} + \frac{\sigma^2}{\sigma^2}\ln\frac{P(\omega_i)}{P(\omega_j)}\frac{(\mu_i-\mu_j)^T(\mu_i-\mu_j)}{\|\mu_i-\mu_j\|^2}$$

$$g_{ij}(x) = w^T(x - x_0) \quad \text{with} \quad w = \frac{1}{\sigma^2}(\mu_i - \mu_j) \quad \text{and} \quad x_0 = \frac{\mu_i+\mu_j}{2} - \sigma^2 \ln\left[\frac{P(\omega_i)}{P(\omega_j)}\right]\frac{(\mu_i-\mu_j)}{\|\mu_i-\mu_j\|^2}$$

Solution of exercise 1

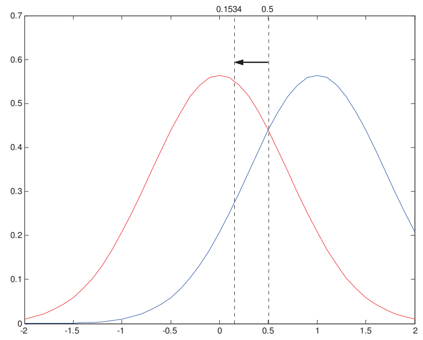
- **Minimization of classification error :**

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{1/2}{1/2} = 1 \longrightarrow x_0 = \frac{1}{2}$$

- **Minimization of the overall risk**

$$L = \begin{pmatrix} 0 & 0,5 \\ 1 & 0 \end{pmatrix}$$

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{1-0}{0.5-0} = 2 \longrightarrow x_1 = \frac{1-\ln(2)}{2}$$



- **The threshold shifts left ➔ ω_2 is favored (the risk is greater when classifying x in ω_1 while it belongs to ω_2 than in the reverse case)**

Solution of exercise 1 *(followed)*

- **Suppose now**
  - That decision losses are identical

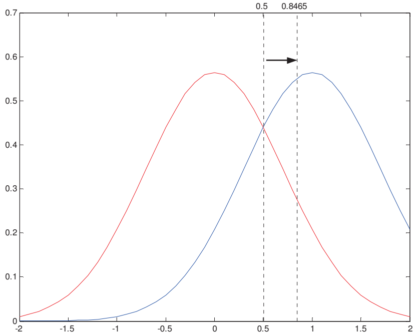
$$L=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

  - That ω_1 is more probable than ω_2 :

$$P(\omega_1)=2/3 \qquad P(\omega_2)=1/3$$

- **Minimization of the overall risk**

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{1/3}{2/3}=\frac{1}{2} \longrightarrow x_1=\frac{1+\ln(2)}{2}$$



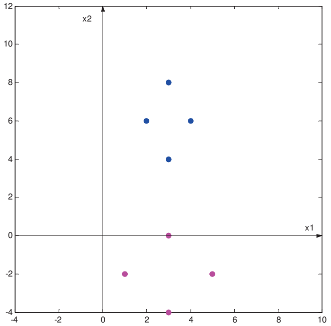
- **The threshold shifts right ➔ the most probable class, ω_1, is favored**

A transition exercise: Exercise 2 *[Duda]*

- **Consider 2 classes with <u>equal priors</u> and Gaussian densities**
- **Take the following samples**

D_1={ (3,4), (3,8), (2,6), (4,6) }

D_2={ (3,0), (1,-2), (5,-2), (3,-4) }



1. **Calculate the parameters of each class. Use:**

$$\mu=\frac{1}{N}\sum_{k=1}^{N}x_k \qquad \Sigma=\frac{1}{N}\sum_{k=1}^{N}(x_k-\mu)(x_k-\mu)^T$$

2. **Calculate the discriminant functions $g_1(x)$ and $g_2(x)$**
3. **What is the locus of points such that $g_1(x)$ = $g_2(x)$? (i.e. find the decision boundary)**

$NB:\ln(2)\approx 0.69$

Solution of exercise 2

x_1	x_2	$x_1-\mu_1$	$x_2-\mu_2$	$(x_1-\mu_1)^2$	$(x_1-\mu_1)(x_2-\mu_2)$	$(x_2-\mu_2)^2$
3	4	0	-2	0	0	4
3	8	0	2	0	0	4
2	6	-1	0	1	0	0
4	6	1	0	1	0	0
3	6			1/2	0	2

$$\Rightarrow \quad \mu_1=\begin{pmatrix}3\\6\end{pmatrix} \quad \text{et} \quad \Sigma_1=\begin{bmatrix}1/2 & 0\\ 0 & 2\end{bmatrix}$$

Similarly,

$$\mu_2=\begin{pmatrix}3\\-2\end{pmatrix} \qquad \Sigma_2=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$$

$$g_i(x)=x^TW_1x+w_1^Tx+w_{10}$$

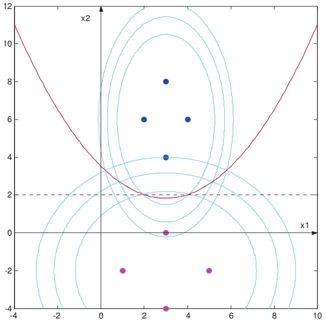
$$W_1=-\frac{1}{2}\Sigma_1^{-1}=\begin{bmatrix}-1 & 0\\ 0 & -1/4\end{bmatrix} \qquad w_1=\Sigma_1^{-1}\mu_1=\begin{pmatrix}6\\3\end{pmatrix}$$

$$w_{10}=-\mu_1^T\Sigma_1^{-1}\mu_1-\frac{1}{2}\ln|\Sigma_1|+\ln P(\omega_1)=-18-\ln(2)$$

$$W_2=\begin{bmatrix}-1/4 & 0\\ 0 & -1/4\end{bmatrix} \qquad w_2=\begin{pmatrix}3/2\\-1\end{pmatrix} \qquad w_{20}=-\frac{13}{4}-2\ln(2)$$

$$0=-\frac{3}{4}x_1^2+\frac{9}{2}x_1+4x_2-18+\frac{13}{4}+\ln(2)$$

$$\Rightarrow \quad x_2=0.1875x_1^2-1.125x_1+3.514$$


12
10
8
6
4
2
0
-2
-4
x2
x1
-4
-2
0
2
4
6
8
10