PA196: Pattern Recognition

1. Introduction
2. Bayesian decision theory

Dr. Vlad Popovici
popovici@iba.muni.cz

Institute of Biostatistics and Analyses
Masaryk University, Brno

Before starting

Organization:

- 14 weeks: 2h lecture + 2h exercise
- 1st half of the semester: business as usual
- 2nd half of the semester (in addition to "business as usual"): *project work*
- evaluation: 50% project, 40% written exam and 10% active participation

Bibliography

- [*DHS*] Duda, Hart, Stork: Pattern Classification. 2nd Ed. 2001
- [*EST*] Hastie, Tibshirani, Friedman: The Elements of Statistical Learning. 2nd Ed. (7th printing). 2013
  Available in PDF!
  `http://statweb.stanford.edu/~tibs/ElemStatLearn/`
- Webb, Copsey: Statistical Pattern Recognition. 3rd Ed. 2011
- Kuncheva: Combining Pattern Classifiers. 2004

Outline

Outline

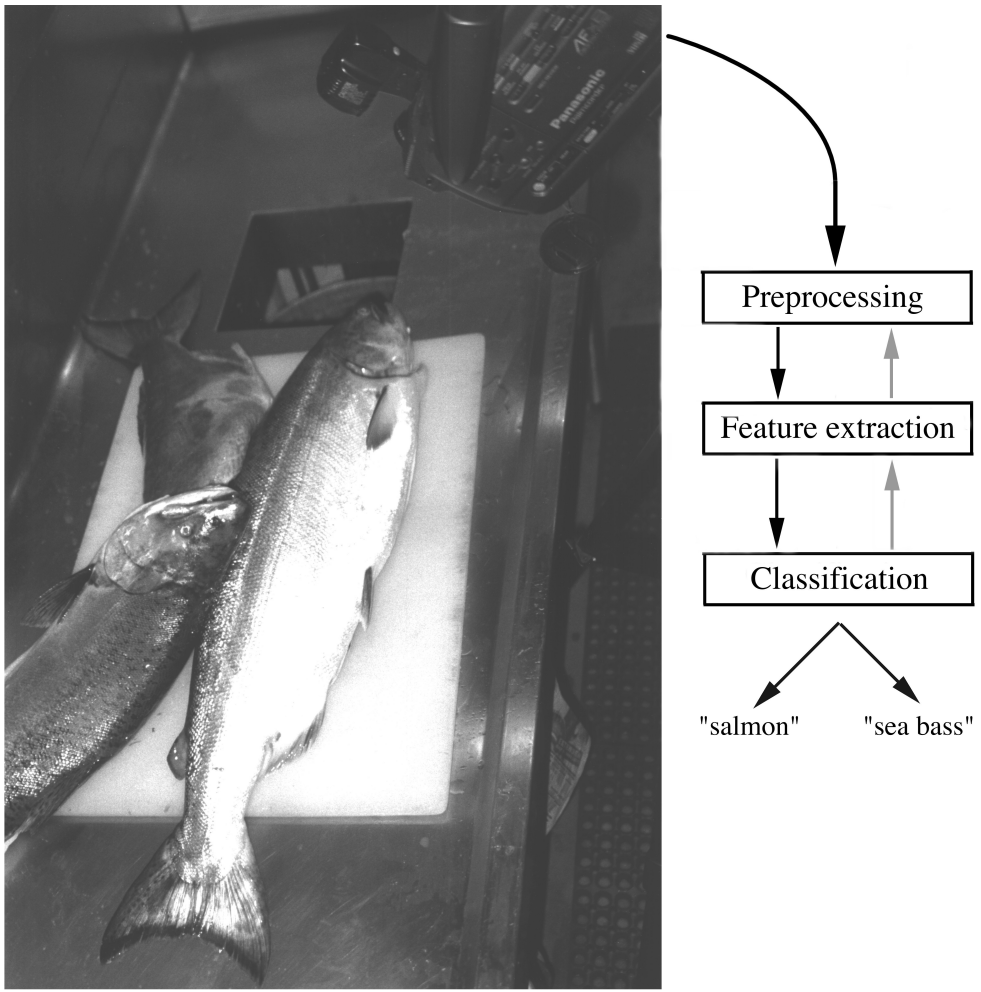
WHAT?

Webster dictionary:

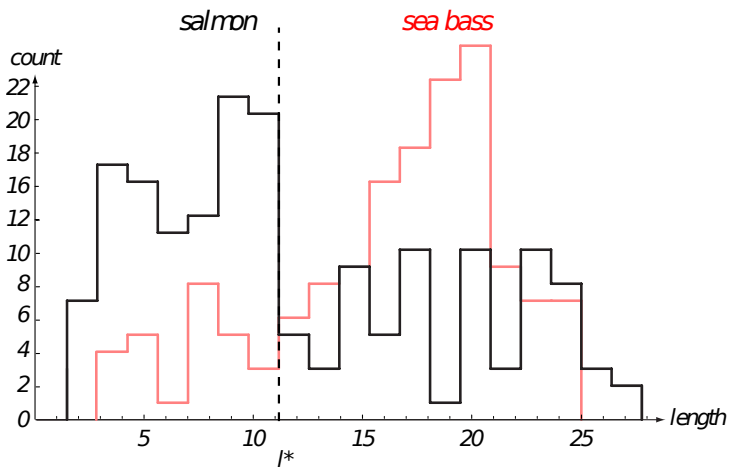
- Pattern: a combination of qualities, acts, tendencies, etc., forming a consistent or characteristic arrangement
- Recognition: the identification of something as having been previously seen, heard, known, etc.

- DHS: "the act of taking in raw data and taking an action based on the category of the pattern"
- *Wikipedia:* "Pattern recognition is nearly synonymous with machine learning. This branch of artificial intelligence focuses on the recognition of patterns and regularities in data. In many cases, these patterns are learned from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning)."

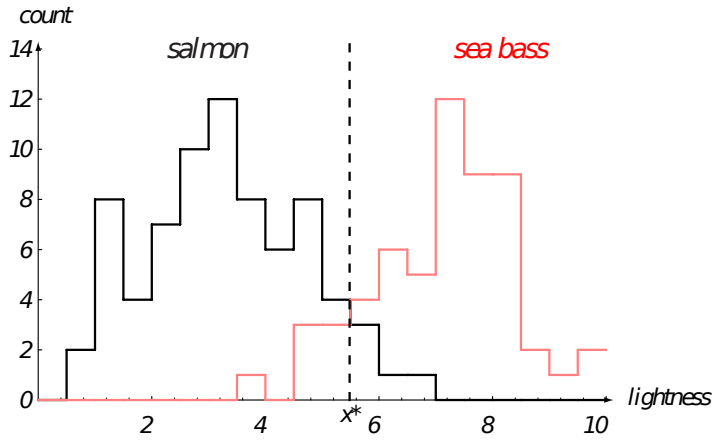
An example - from DHS (figs 1.1-1.4)



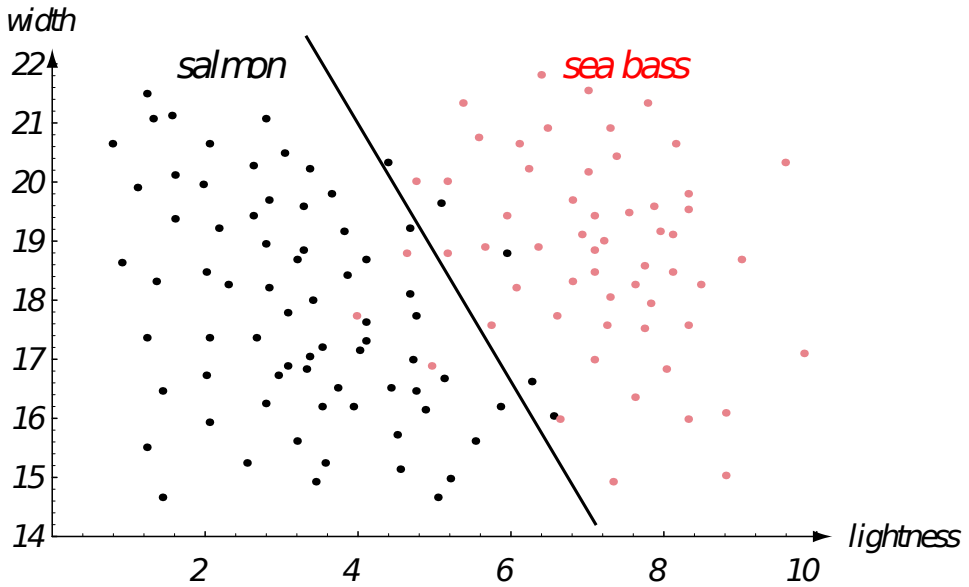
Characteristic/feature: width



Characteristic/feature: lightness



Combining width and lightness features:



Outline

Applications - in no particular order

Biometrics:

- face detection and recognition/detection
- gender and age recognition
- fingerprint recognition
- speaker recognition
- ...

Human-computer interfaces:

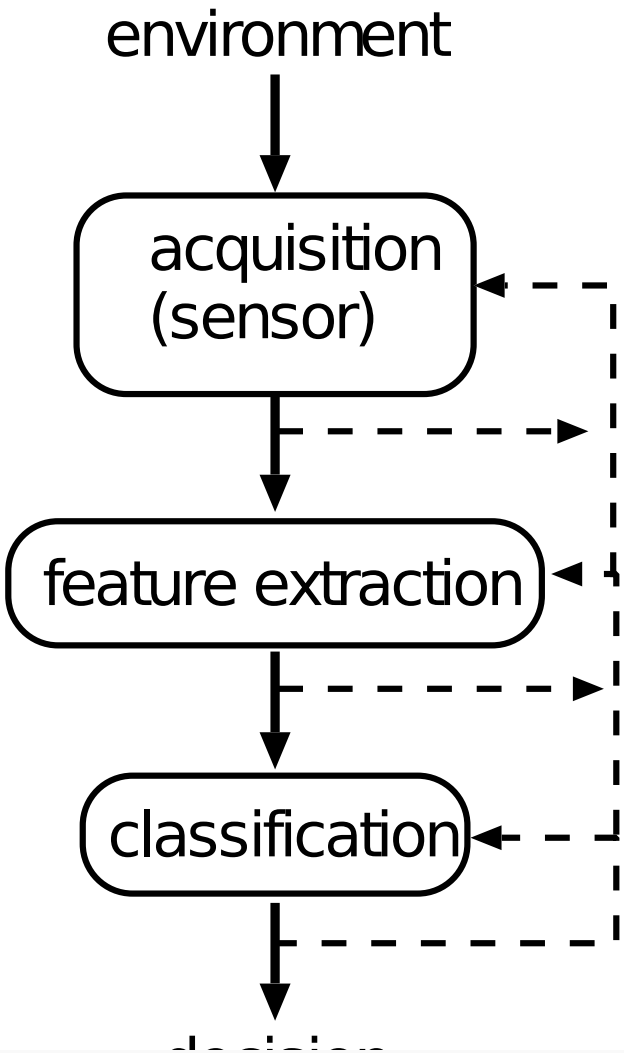
- user detection/recognition
- gait recognition
- gesture recognition
- brainwave categorization
- ...

Other:

- biomedical research: prediction of response, target genes, etc etc
- military/security: target detection, intrusion detection, etc etc
- spam filtering
- optical character recognition
- natural language processing
- remote sensing
- etc etc

Outline

Pattern recognition systems



Example:

- acquisition/sensor: CCD camera
- feature extraction: in all rectangular regions of size 30×30, compute the Gabor wavelet decomposition in corner regions
- given all the coefficients of Gabor w., classify the region as human face or "something else"

Some basic terminology

- learning: model fitting, optimization, training
- main types of problems:
  - CLASSIFICATION: the classes are known, the problem is to assign classes to the inputs; approached usually by supervised learning: samples and the corresponding labels are used in training the classifier
  - CLASS DISCOVERY: the classes are not known (maybe not even how many they are); approached usually by clustering: grouping similar inputs together
- other methods pf learning:
  - semi-supervised learning: some labeled and some unlabeled data
  - reinforcement learning: there is a "teacher" telling the system when it's right or wrong

Approaches:

- no clear cut separation between method types
- *statistical/Bayesian*: features are random variates; estimate the PDFs and use maximum a posterior for classification; minimize the "risk" of misclassification
- *geometric*: find boundaries between regions of the feature space
- *neural networks*
- *model-based*: reference pattern represent classes; "nearest pattern" rule is used for classification
- *syntactic*: classes are represented by grammars
- *structural*: classes represented by graphs (or similar)

Design cycle

- data collection $\longrightarrow$ sample size estimation
- feature selection
- classifier design $\longrightarrow$ selection of classifier(s), training, model selection
- performance estimation $\longrightarrow$ errors and costs, error estimates, variability of the estimates

Other issues

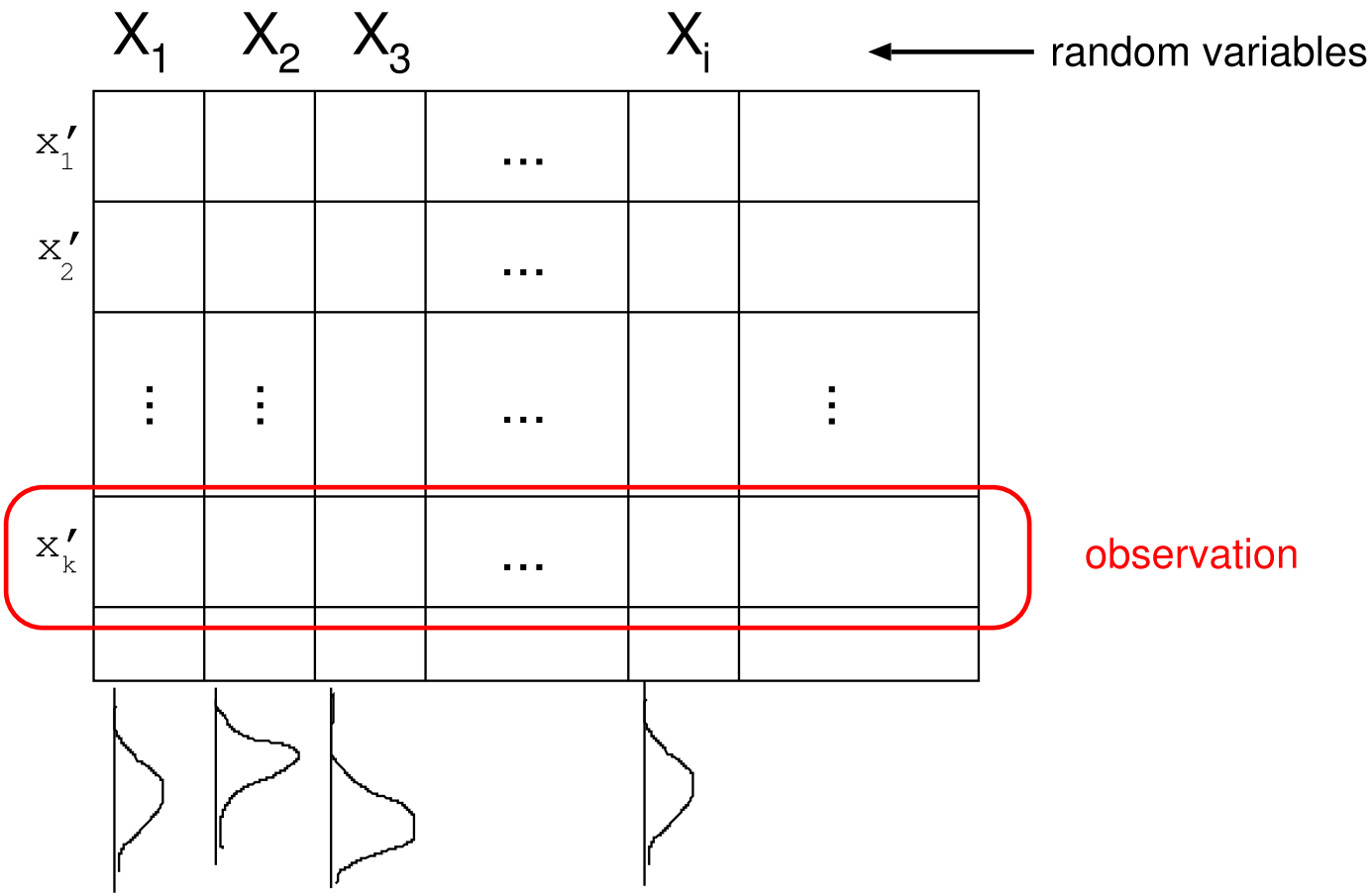
- pre-processing and normalization
  - improves stability of the models and convergence of the learning procedure
  - depend on classifier and application domain
  - feature standardization
- detection of outliers
- detection of errors in data

Goals of classifier design

- to build a classification model from a *finite* set of examples that minimizes some error measure and which *generalizes* well
- to *estimate* its future performance on unseen data

A bit of formalism

- a sample or a pattern is represented as a real-valued vector: $\mathbf{x} \in \mathbb{R}^d$
- $\mathbf{x} = [x_1, \ldots, x_i, \ldots, x_d]^t$, x_i is called variable or feature (actually, it is a realization - measurement - of a given variable)
- generally, there is a label $g \in \mathcal{G} = \{g_1, \ldots, g_m\}$ uniquely associated with each sample
- there is a probability $P(g)$ that each class would be observed - a priori probability (prior). Normally: $\sum_i P(g_i) = 1$.
- the probability of observing a sample $\mathbf{x}$, *given a class* g_i, is given by the class-conditional probability density $p(\mathbf{x}|g_i)$
- NOTE: $P(\cdot)$ is used for probability *mass* function (discrete variables) and $p(\cdot)$ for probability *density* function


X1
X2
X3
Xi
random variables
x'1
x'2
x'k
observation

Similarities, metrics, distances...

- most methods rely on an explicit or implicit distance between data points: $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$,
  - $d(\mathbf{x}, \mathbf{z}) > 0, \ \forall \mathbf{x} \neq \mathbf{z}$
  - $d(\mathbf{x}, \mathbf{z}) = 0 \iff \mathbf{x} = \mathbf{z}$
  - $d(\mathbf{x}, \mathbf{z}) = d(\mathbf{z}, \mathbf{x})$
- a metric is a distance which satisfies the *triangle inequality:* $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$, for $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$.
- a similarity measure is less formally defined; usually has large values for more alike objects
- examples: Euclidean metric; correlation coefficient as similarity measure

Outline

Outline

Bayes rule

Let $\mathcal{G} = \{g_i\}$ be a number of classes and let $\mathcal{X} = \{\mathbf{x}_i\}$ be a set of *observed* data points (i.e. training set).

- $p(\mathbf{x}|g)$ is called likelihood (function) with the variable g (the class label). Note: if g is fixed, but $\mathbf{x}$ are considered random, we have the class-conditional density model for generating the observations.
- $P(g)$ is the prior
- the goal is to find $P(g|\mathbf{x})$ i.e. the posterior probability that the label for $\mathbf{x}$ is g
- Bayes rule:

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

Bayes rule

$$\texttt{posterior} \propto \texttt{likelihood} \times \texttt{prior}$$

Tricky part: how to estimate the likelihood?!

Class-conditional density or likelihood function?

- ...it depends!
- consider a set of r.v. $X_1, \ldots, X_p$ *conditionally independent* given that $\Theta = \theta$
- $p_{X_i|\Theta}(\cdot|\theta)$ is the (postulated) density model for the variable X_i: for each possible value of Θ is the uncertainty about the values of X_i
- $\Pr\{X_1 \in A_1, \ldots, X_p \in A_p\} = \int_{A_1 \times \cdots \times A_p} \prod_{i=1}^{p} p_{X_i|\Theta}(x_i|\theta)\, dx_1 \ldots dx_p$
- once $[x_1, \ldots, x_p]$ are observed (hence they are fixed!), define the function

$$L_{x_1,\ldots,x_p} : \Omega \to \mathbb{R}; \quad L_{x_1,\ldots,x_p}(\theta) = \prod_{i=1}^{p} p_{X_i|\Theta}(x_i|\theta)$$

and call it likelihood function

Bayesian decision

- consider there are K classes $\{g_1, \ldots, g_K\}$
- there are a possible actions: $\{\alpha_1, \ldots, \alpha_a\}$
- let $\alpha(\mathbf{x})$ be the *decision rule/function*
- for each action-class pair there is a *loss* incurred: $\lambda(\alpha_k|g_i)$
- conditional risk:

$$R(\alpha_k|\mathbf{x}) = \sum_{i=1}^{K} \lambda(\alpha_k|g_i) P(g_i|\mathbf{x})$$

- the *expected risk* for the rule $\alpha(\mathbf{x})$

$$R = \int_x R(\alpha(\mathbf{x})|\mathbf{x}) p(\mathbf{x})\, d\mathbf{x}$$

Bayesian decision

$$\alpha^* = \arg\min_k R(\alpha_k|\mathbf{x})$$

- results in minimum expected risk
- is the best decision that one can take
- you can use this framework to build "test cases" for other classifiers

Example

- consider a classification problem (actions correspond to class assignment), with 0-1 loss function

$$\lambda(g_k|g_i) = \begin{cases} 0, & k = i \\ 1, & k \neq i \end{cases}$$

- the conditional risk becomes

$$R(g_k|\mathbf{x}) = \sum_{i=1}^{K} \lambda(g_k|g_i) P(g_i|\mathbf{x}) = \sum_{i \neq k} P(g_i|\mathbf{x}) = 1 - P(g_k|\mathbf{x})$$

- Bayesian decision rule becomes maximum a posteriori (MAP) rule

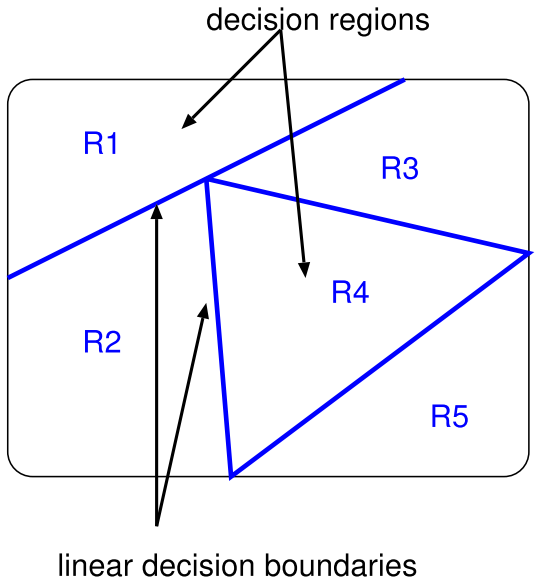
Bayesian classification - Maximum A Posteriori rule

Assign $\mathbf{x}$ to class g_k if $P(g_k|\mathbf{x}) > P(g_i|\mathbf{x})\ \forall i \neq k$

Outline

Discriminant functions

- d-functions: $h_i(\mathbf{x}), i = 1, \ldots, K$
- classifier: $\mathbf{x}$ is assigned to g_i if $h_i(\mathbf{x}) > h_k(\mathbf{x}), \forall k \neq i$
- ex.: $h_i(\mathbf{x}) = P(g_i|\mathbf{x})$ or $h_i(\mathbf{x}) = \ln p(\mathbf{x}|g_i) + \ln P(g_i)$
- different d-functions may give equivalent classifiers



Binary case

- if $K = 2$: binary classifier or dichotomizer
- multiclass problems can be decomposed in a series of 2-class problems
- a single discriminant function suffices:

$$h(\mathbf{x}) = h_1(\mathbf{x}) - h_2(\mathbf{x})$$

  with the decision rule: assign $\mathbf{x}$ to g_1 if $h(\mathbf{x}) > 0$
- this is the case we will study most of the time

Outline

The case of normal density

For $\mathbf{x} \in \mathbb{R}^p$ a column vector, $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$:

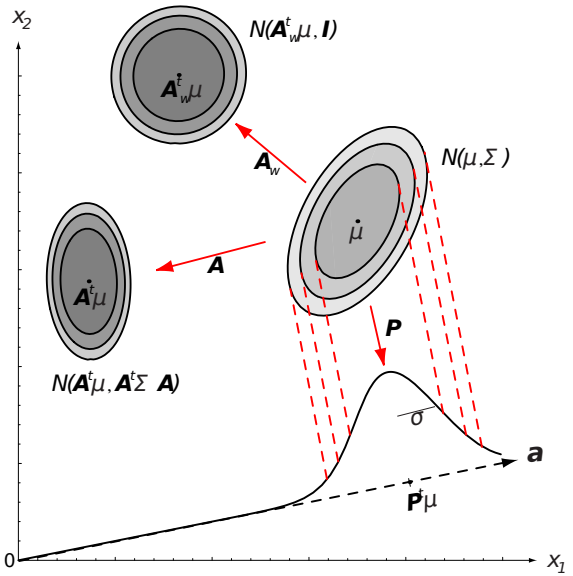
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)\right]$$

- μ is the mean vector and Σ is the covariance matrix (symmetric, positive definite)

$$\Sigma = \mathcal{E}[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^t] = \int (\mathbf{x}-\mu)(\mathbf{x}-\mu)^t p(\mathbf{x})\, d\mathbf{x}$$

- for an $p \times r$ matrix $\mathbf{A}$: $\mathbf{y} = \mathbf{A}^t\mathbf{x} \sim \mathcal{N}(\mathbf{A}^t\mu, \mathbf{A}^t\Sigma\mathbf{A})$
- *whitening:* $\mathbf{A}_w = \Phi\Lambda^{-1/2}$ where Φ is the matrix with eigenvectors of Σ as columns and Λ is a diagonal matrix with corresponding eigenvalues on diagonal

[DHS - Fig.2.8:]



Mahalanobis distance

$$r = \sqrt{(\mathbf{x} - \mu)^t \Sigma^{-1} (\mathbf{x} - \mu)}$$

- if the variables/features are independent and standardized, Σ becomes the identity matrix and the M-distance becomes Euclidean distance
- the volume of the hyperellipsoid corresponding to a distance r is

$$V = V_p |\Sigma|^{1/2} r^p,$$

where

$$V_p = \begin{cases} \pi^{p/2}/(d/2)! & \text{if } p \text{ is even} \\ 2^p \pi^{(p-1)/2} \left(\frac{d-1}{2}\right)!/p! & \text{if } p \text{ is odd} \end{cases}$$

Discriminant functions for Normal densities

Assume $p(\mathbf{x}|g_i) \sim \mathcal{N}(\mu_i, \Sigma_i)$.
Since

$$P(g_i|\mathbf{x}) \propto p(\mathbf{x}|g_i)P(g_i)$$

one can take a discriminant function of the form:

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

which leads to

$$h_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(g_i)$$

Special case: $\Sigma_i = \sigma^2 I$

$$h_i(\mathbf{x}) = -\frac{1}{2\sigma^2}\|\mathbf{x} - \mu_i\|^2 + \ln P(g_i)$$

which can be re-written as linear discriminant functions

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

with $\mathbf{w}_i \in \mathbb{R}^p$,

$$\mathbf{w}_i = \frac{1}{\sigma^2}\mu_i \qquad \leftarrow \text{coefficients}$$

$$w_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + \ln P(g_i) \qquad \leftarrow \text{threshold or bias}$$

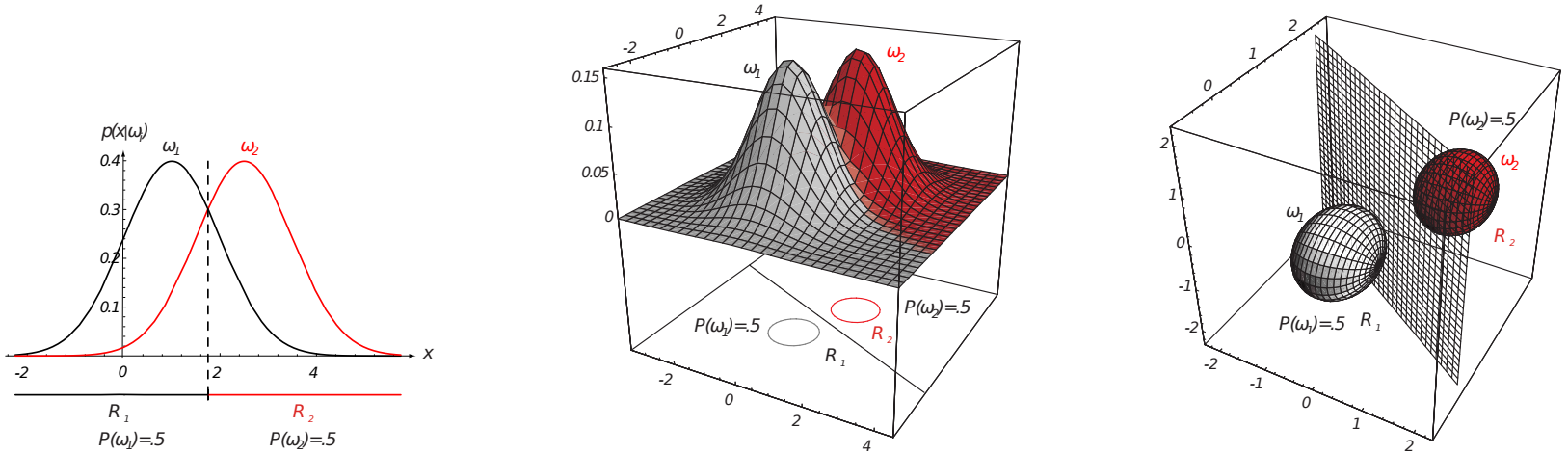
Decision surface: let *i* and *j* be the categories with highest posteriors. The eq. of the decision boundary is given by

$$h_i(\mathbf{x}) = h_j(\mathbf{x})$$

and can be written as

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

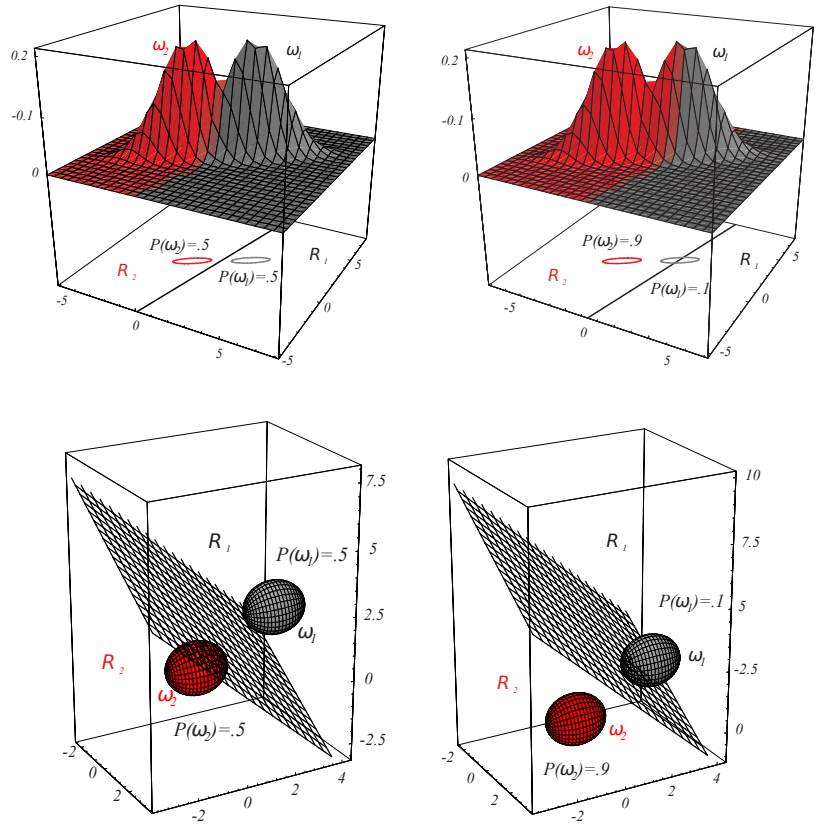
with $\mathbf{w} = \mu_i - \mu_j$ and $\mathbf{x}_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(g_i)}{P(g_j)}(\mu_i - \mu_j)$



[DHS - Fig.2.10]

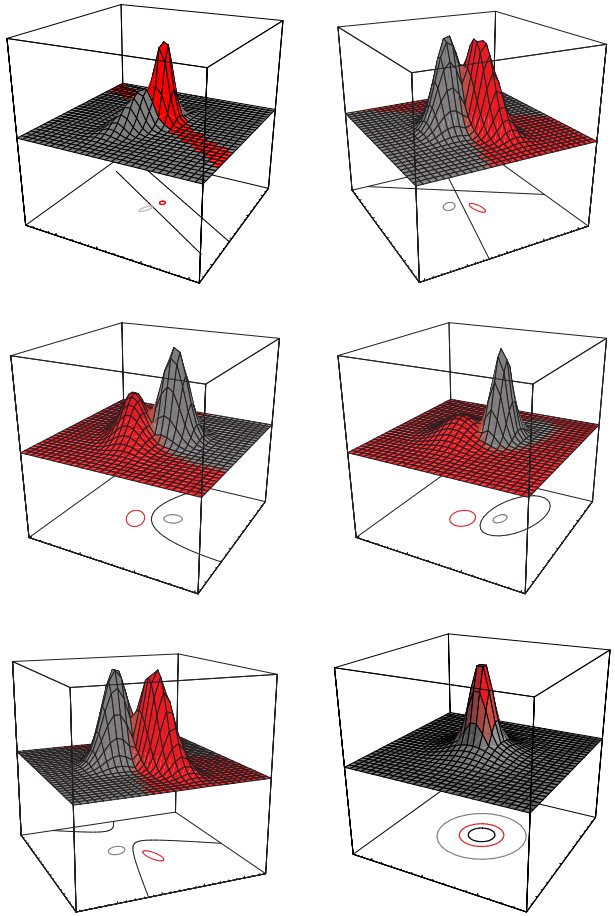
Special case: $\Sigma_i = \Sigma$

→ the separation hyperplane is no longer orthogonal on the line between Gaussian centers



[DHS - Fig.2.12]

General case: Σ_i arbitrary

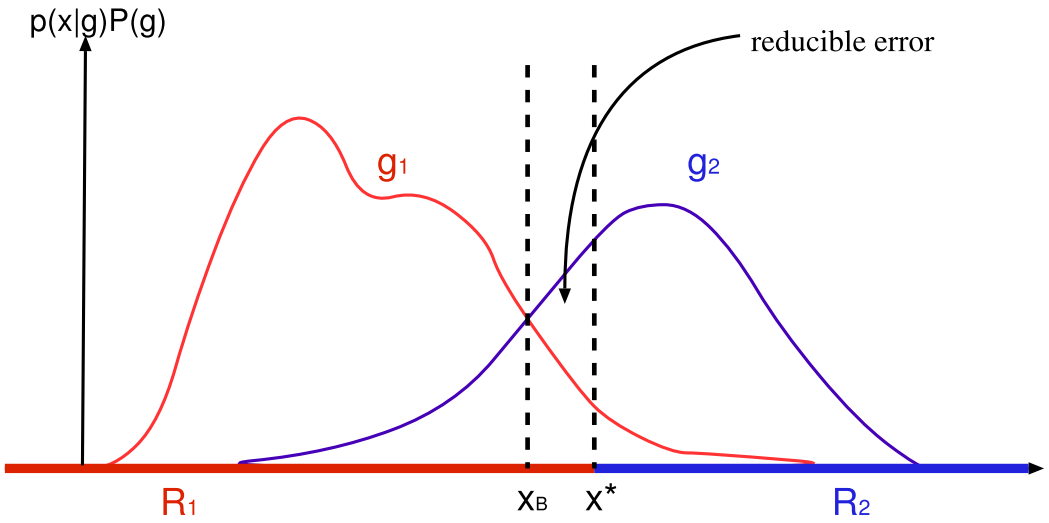


[DHS - Fig.2.14]

Outline

Errors

- *error*: predict the wrong class
- consider the binary classification problem
- $P_{err} = P(\mathbf{x} \in \mathcal{R}_2, g_1) + P(\mathbf{x} \in \mathcal{R}_1, g_2)$



$\mathbf{x}_B$: optimal (Bayes) decision; $\mathbf{x}^*$: another decision threshold

So

$$P_{err} = \int_{\mathcal{R}_2} p(\mathbf{x}|g_1)P(g_1)\, d\mathbf{x} + \int_{\mathcal{R}_1} p(\mathbf{x}|g_2)P(g_2)\, d\mathbf{x}$$

- minimum of P_{err} is obtained for $\mathbf{x}^* = \mathbf{x}_B$
- to compute P_{err} we need the class-conditional probabilities
- for Gaussian probabilities and binary classification, one can show that

  $$P_{err} \leq \exp\left(-\psi(\beta, \mu_{1,2}, \Sigma_{1,2})\right)$$

  where β can be optimized to minimize the rhs (Chernoff bound)
- for $\beta = \frac{1}{2}$ one obtains the Bhatacharyya bound

Wrap-up

- Bayesian theory offers a complete framework for building classfiers (among other applications)
- minimize the overall risk → choose the action the minimizes the conditional risk
- under normality assumption, exact formulation of the optimal classifier can be derived
- tight error bounds can also be computed
- ...but this assumption is rarely true in practice