Pattern Recognition Techniques

Bernardete Ribeiro
DEI-FCTUC, University of Coimbra

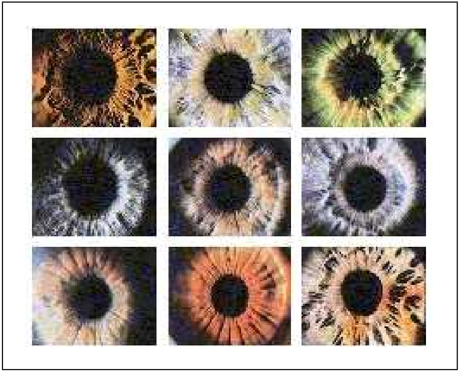
September 9, 2009

Pattern Recognition Techniques

Chapter 1: TRP Introduction

Chapter 1: Introduction to Pattern Recognition

TRP: 2009-2010



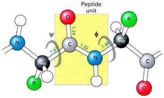
Pattern Recognition?

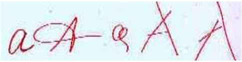
The assignment of a physical object or event to one of several pre-specified categories – Duda & Hart

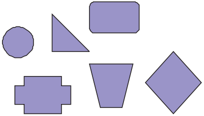
- A **pattern** is an object, process or event
- A **class** (or category) is a set of patterns that share common attribute (features) usually from the same information source
- During **recognition** (or classification) classes are assigned to the objects.
- A **classifier** is a machine that performs such task

What is a pattern?

A pattern is the opposite of a chaos; it is an entity vaguely defined, that could be given a name

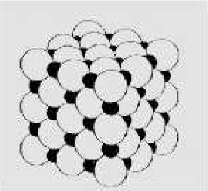


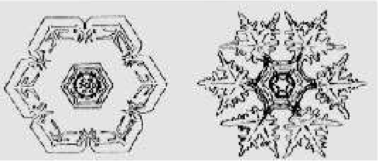




Examples of Patterns

- Cristal Patterns: atomic or molecular

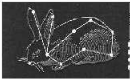




- Their structures are represented by 3D graphs and can be described by deterministic grammars or formal languages

Examples of Patterns

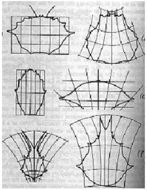
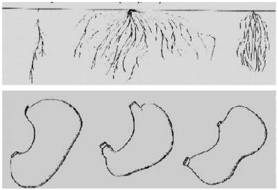
- Patterns of Constellations

- Patterns of constellations are represented by 2D planar graphs
- Human perception has strong tendency to find patterns from anything. We see patterns from even random noise — we are more likely to believe a hidden pattern than denying it when the risk (reward) for missing (discovering) a pattern is often high.

Examples of Patterns

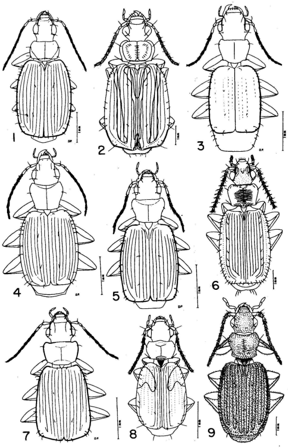
- Biological Patterns —morphology

- Landmarks are identified from biologic forms and these patterns are then represented by a list of points. But for other forms, like the root of plants, Points cannot be registered crossing instances.
- Applications: Biometrics, computacional anatomy, brain mapping, ...

Examples of Patterns

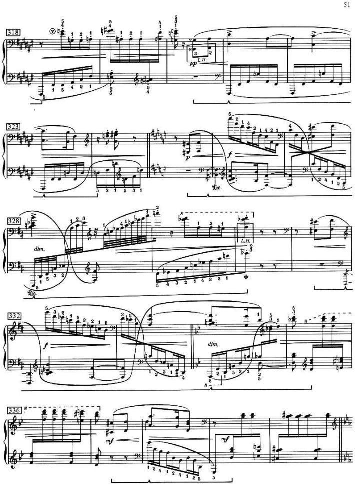
- Biological Patterns



- Landmarks are identified from biologic forms and these patterns are then represented by a list of points.

Examples of Patterns

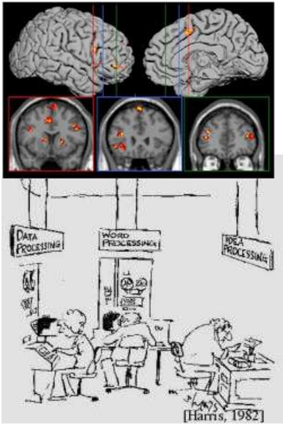
- Music Patterns



- Ravel Symphony?

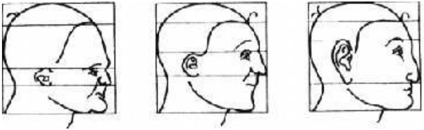
Examples of Patterns

- Patterns Behavior?



Examples of Patterns

- Discovery and Association of Patterns



- Statistics show connections between the shape of one's face (adults) and his/her Character. There is also evidence that the outline of children's face is related to alcohol abuse during pregnancy.

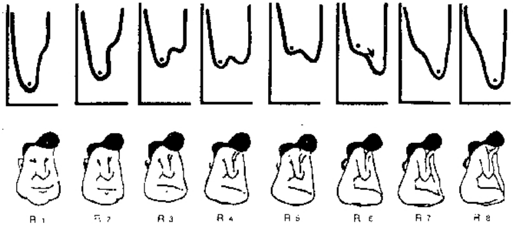
Examples of Patterns

- People Recognition

- Funny funny

Examples of Patterns

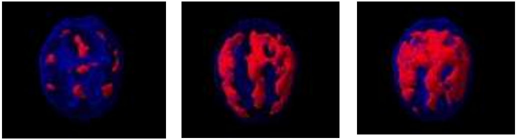
- Discovery and Association of Patterns



- What are the features?
  Statistics show connections between the shape of one' s face (adults) and his/her Character.

Examples of Patterns

- Patterns of Brain Activity



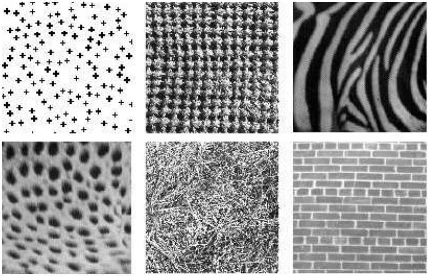
- We may understand patterns of brain activity and find relationships between brain activities, cognition, and behaviors

Examples of Patterns

- Variation Patterns:
  1. Expression: geometric deformation
  2. illumination: Photometric deformation
  3. Transformation: 3D pose 3D
  4. Noise and Occlusion

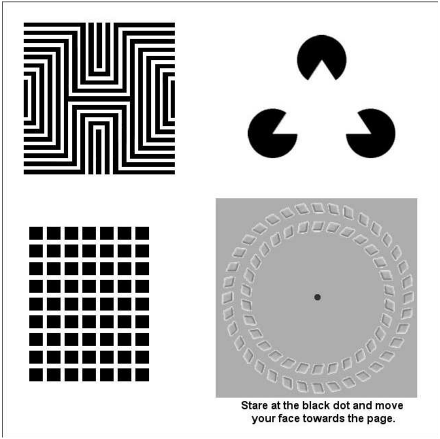
Examples of Patterns

- A broad range of texture patterns are generated by stochastic processes.



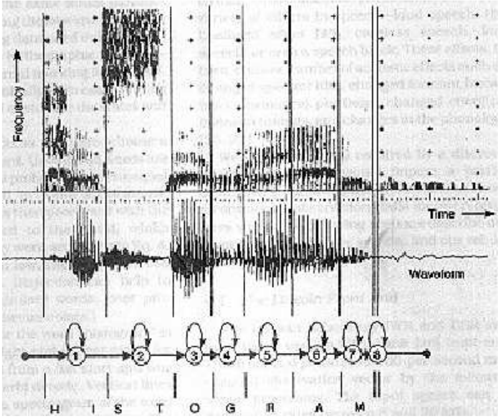
Examples of Patterns

- How are these patterns represented in human mind?



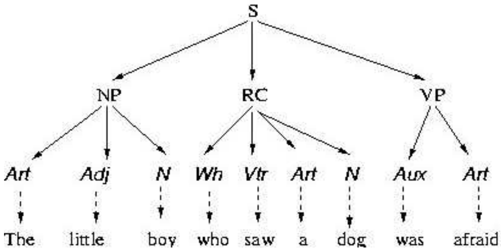
Examples of Patterns

- Speech signals and Hidden Markov models



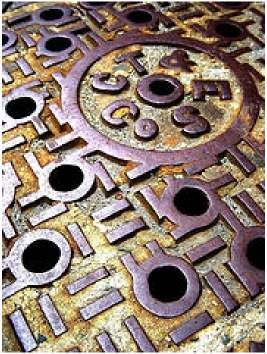
Examples of Patterns

- Natural Language and stochastic grammar.



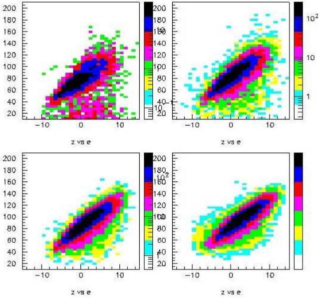
Examples of Patterns

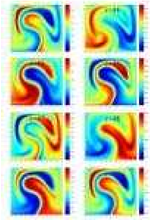
- Patterns everywhere?



Examples of Patterns

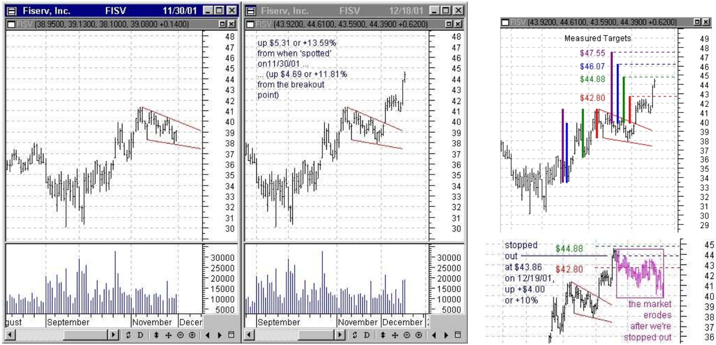
- Geographical Patterns





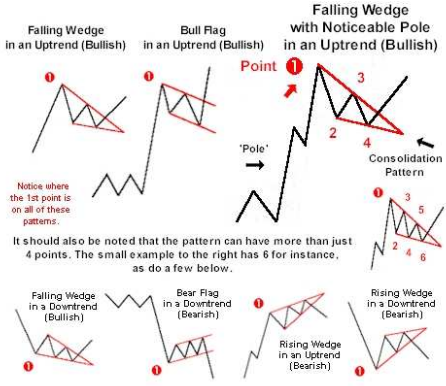
Examples of Patterns

- Financial Series Pattern Recognition



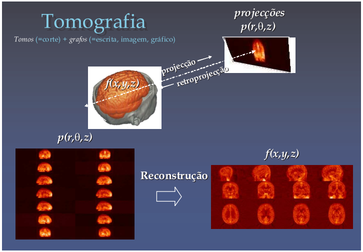
Examples of Patterns

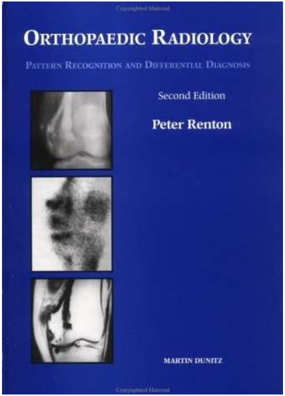
- How to Trade Chart Patterns ?



Examples of Patterns

- Pattern Recognition in Medical Diagnosis

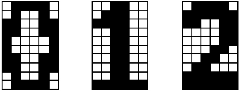




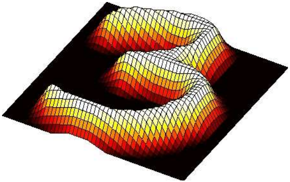
Examples of Patterns

- Optical Character Recognition

Padrões a memorizar

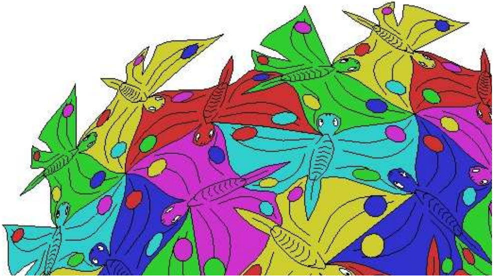


Padrão a ser reconhecido



Examples of Patterns

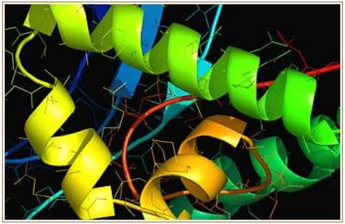
- Graphical arts



Examples of Patterns

Human Genom

Human Proteom



Examples of Applications (1)

- Optical Character Recognition (OCR)
  - Handwritten: sorting letters by postal code, input device for PDA's.
  - Printed texts: reading machines for blind people, digitalization of text documents
- Biometrics
  - Face recognition, verification, retrieval.
  - Finger prints recognition.
  - Speech recognition.

Examples of Applications (2)

- Diagnostic systems
  - Medical diagnosis: X-Ray, EKG analysis.
  - Machine diagnostics, waster detection
- Military applications
  - Automated Target Recognition (ATR).
  - Image segmentation and analysis (recognition from aerial or satelite photographs).

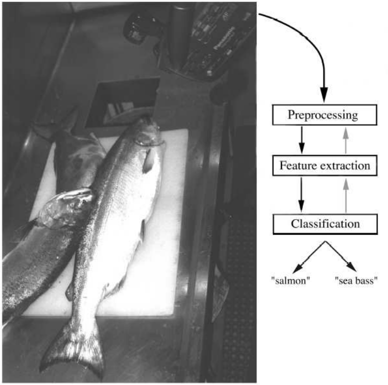
Pattern Recognition Approaches

- **Statistical** PR: based on underlying statistical model of patterns and pattern classes.
- **Neural Networks**: classifier is represented as a network of cells modeling neurons of the human brain (connectionist approach).
- **Support Vector Machines**: Global optimal for classification and regression problems
- **Structural (or syntactic)** PR: pattern classes represented by means of formal structures as grammars, automata, strings, etc.

An example of Pattern Recognition

Classification of fish into two classes: **Salmon** and **Sea Bass** by discriminative method

- Sorting incoming Fish on a conveyor according to species using optical sensing



Problem Analysis

- Set up a camera and take some sample images to extract features
  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth, etc.
- This is the set of all suggested features to explore for use in our classifier!

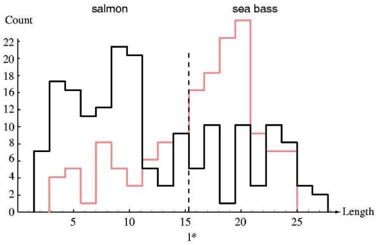
Pattern Recognition Phases

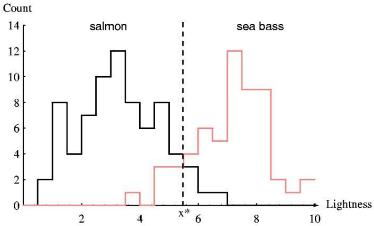
- Pre-process raw data from camera
- Segment isolated fish
- Extract features from each fish (length,width, brightness, etc.)
- Classify each fish

Pattern Recognition Phases

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
  - Information from a single fish is sent to a **feature extractor** whose purpose is to reduce the data by measuring certain features
- The features are passed to a **classifier**
- Classification
  - Select the length of the fish as a possible feature for discrimination

Features and Distributions



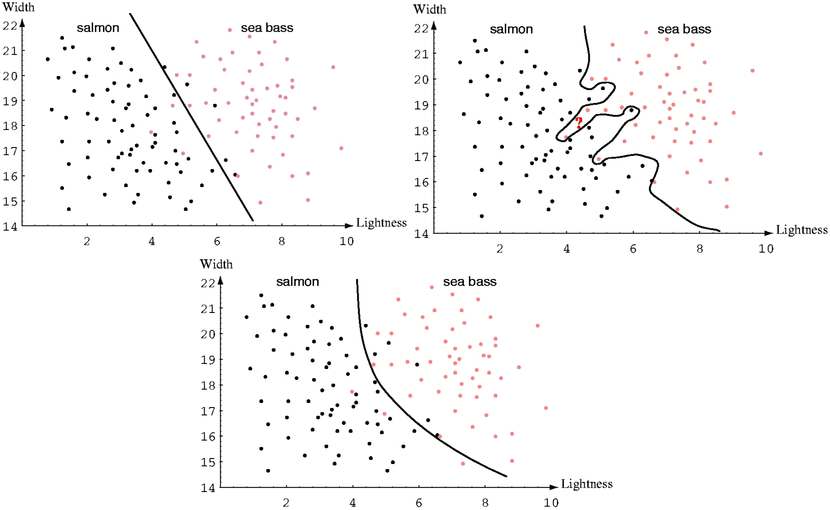


- The **length** is a poor feature alone!
- Select the **lightness** as a possible feature.

Decision Theory

- Customers do not want sea bass in their cans of salmon
  - Threshold decision boundary and cost relationship
- Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)
- Adopt the lightness and add the width of the fish
  Fish $\mapsto [x_1, x_2]$

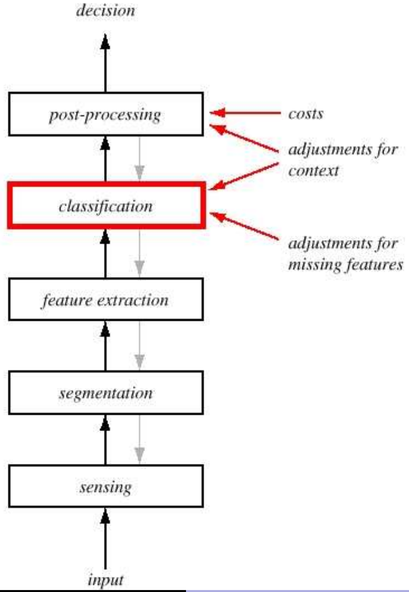
Decision/classification Boundaries (1)



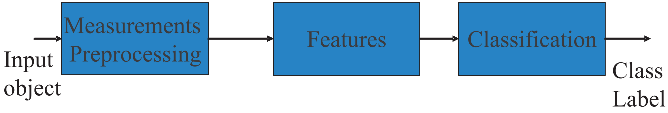
Decision/classification Boundaries (2)

- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input $\longrightarrow$ Issue of **generalization**!

A Complete PR System



Problem Formulation

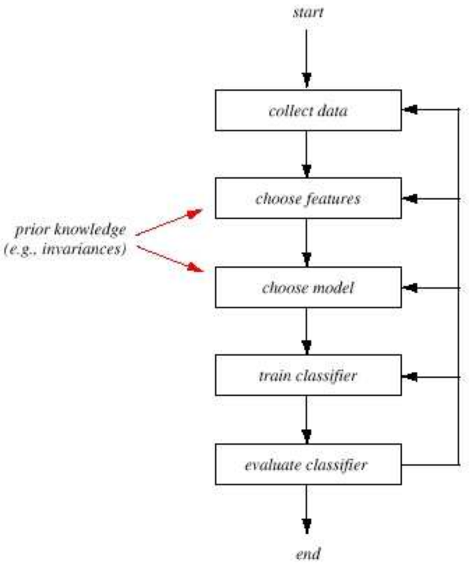


- Basic ingredients:
  - Measurement space (e.g., image intensity, pressure)
  - Features (e.g., corners, spectral energy)
  - Classifier - soft and hard
  - Decision boundary
  - Training sample
  - Probability of error

Design Cycle (1)

- Feature selection and extraction
  - — What are good discriminative features?
- Modeling and learning
  - Dimension reduction, model complexity
  - Decisions and risks
  - Error analysis and validation.
  - bounds and capacity.
  - Algorithms

Design Cycle (2)



Design Cycle (3)

- Data Collection
  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?
- Feature Choice
  - Depends on the characteristics of the problem domain. Simple to extract, invariant to irrelevant transformation, insensitive to noise
- Model Choice
  - Unsatisfied with the performance of our linear fish classifier and want to jump to another class of model

Design Cycle (4)

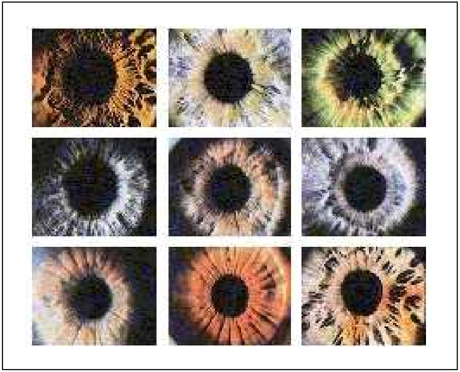
- Training
  - Use data to determine the classifier. Many different procedures for training classifiers and choosing models
- Evaluation
  - Measure the error rate (or performance) and switch from one set of features & models to another one.
- Computational Complexity
  - What is the trade off between computational ease and performance? (How an algorithm scales as a function of the number of features, number or training examples, number patterns or categories?)

Pattern Recognition Techniques

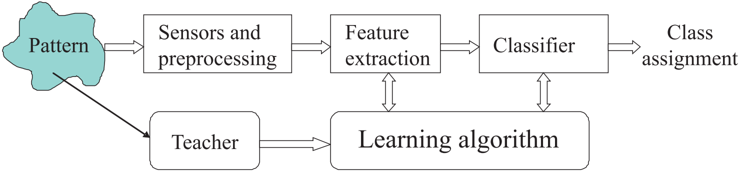
Chapter 2: TRP Pattern Discrimination

Chapter 2: Pattern Discrimination

TRP: 2009-2010

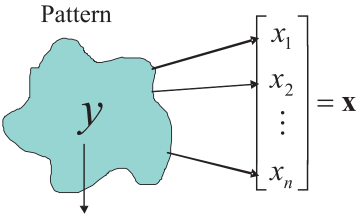


Components of PR system



- **Sensors and preprocessing.**
- **A feature extraction** aims to create discriminative features good for classification.
- **A classifier.**
- **A teacher** provides information about hidden state – supervised learning.
- **A learning algorithm** sets PR from training examples.

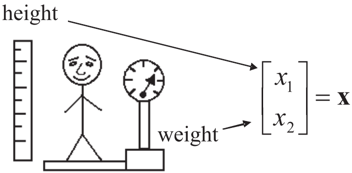
Components of PR system



- Feature vector $\mathbf{x} \in X$
  - A vector of observations (measurements). $\mathbf{x}$ is a point in feature space X

- Hidden state $y \in Y$
  - Cannot be directly measured.
  - Patterns with equal hidden state belong to the same class.
- Task
  - To design a classifer (decision rule) $d : X \longrightarrow Y$
  - which decides about the class of an observation.

Example

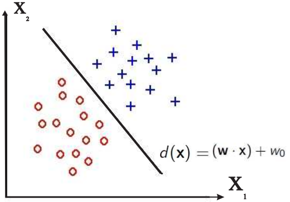


- Task: Jockey-Hoopster recognition
- The set of hidden state is $Y = \{H, J\}$
- The feature space is $X = \mathbb{R}^2$

- Linear Classifier:

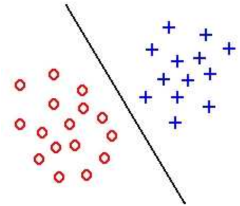
$$d(\mathbf{x}) = \begin{cases} H \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \geq 0 \\ J \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \leq 0 \end{cases}$$

Training Examples: $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\}$

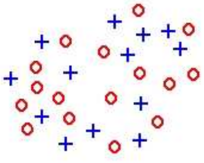


Feature extraction

- Task: to extract features which are good for classification. Good features:
  - Objects from the same class have similar feature values.
  - Objects from different classes have different values.

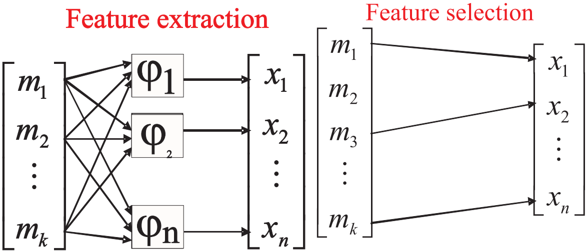


"Good" features



"Bad" features

Feature Extraction Methods

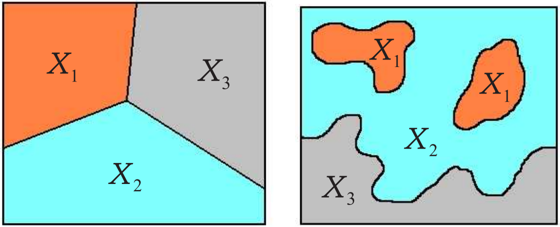


- Problem can be expressed as optimization of parameters of feature extractor .
- **Supervised methods**: objective function is a criterion of separability (discriminability) of labeled examples, e.g., linear discriminant analysis (LDA).
- **Unsupervised** methods: lower dimensional representation which preserves important characteristics of input data is sought for, e.g., principal component analysis (PCA).

Classifier

A classifier partitions feature space X into class-labeled regions such that

$$X = X_1 \cup X_2 \cup X_{|Y|} \text{ and } X = X_1 \cap X_2 \cap X_{|Y|}$$



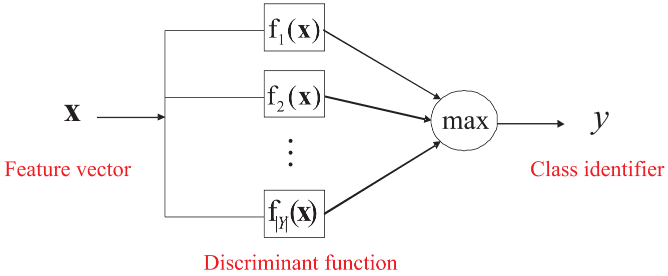
The classification consists of determining to which region a feature vector **x** belongs to. Borders between decision boundaries are called decision regions.

Representation of classifier

A classifier is typically represented as a set of discriminant functions

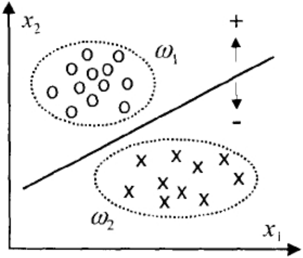
$$f(\mathbf{x}_i) : X \longrightarrow \mathbb{R}, i = 1 \cdots |Y|$$

The classifier assigns a feature vector **x** to the i-the class if $f(\mathbf{x}_i) < f(\mathbf{x}_j) \forall i \neq j$



Pattern Discrimination Revisited

- Decision Regions and Functions
  The feature space is $\mathbf{x} = [x_1, x_2] \in \mathbb{R}^2$



- Linear decision function:
  $d(\mathbf{x}) = x_1 w_1 + x_2 w_2 + w_0$
- Linear classifier:

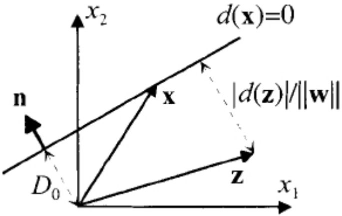
$$decision = \begin{cases} \omega_1 \ \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \geq 0 \\ \omega_2 \ \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \leq 0 \end{cases}$$

or

$$decision = \begin{cases} 1 \ \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \geq 0 \\ 0 \ \ if (\mathbf{w} \cdot \mathbf{x}) + w_0 \leq 0 \end{cases}$$

Decision Surface

Discriminant Function (Hyperplane) $d(\mathbf{x})$



- $$D_0 = \frac{w_0}{||\mathbf{w}||}$$
- $$\mathbf{n} = \frac{\mathbf{w}}{||\mathbf{w}||}$$
- where $||\mathbf{w}||$ represents the vector **w** length
- Distance of any point z to the hyperplane

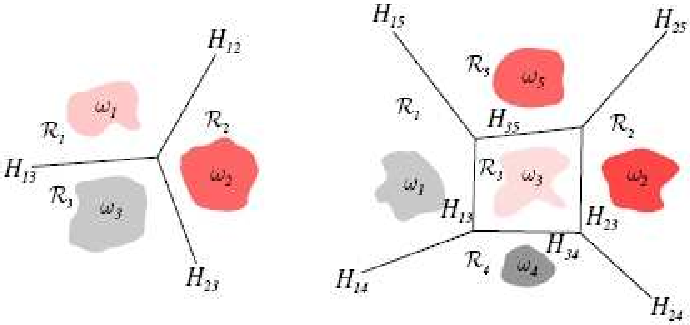
$$|d(\mathbf{z})|/||\mathbf{w}||$$

Generalised Decision Functions

- Working in $\mathbb{R}^d$ with d dimensions
- Generalised Decision Function

$$d_i(\mathbf{x}_i) > 0 \ \mathit{if} \ \mathbf{x} \in \omega_i; \ d_i(\mathbf{x}_i) < 0 \ \mathit{if} \ \mathbf{x} \in \omega_j \text{ with } j \neq i$$

Linear Discriminant Functions: multi-category case



Generalised Decision Functions

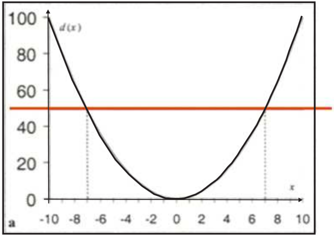
- Generalised Decision Function with Threshold Δ

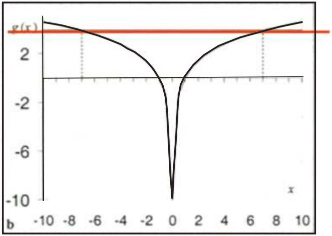
$$d_i(\mathbf{x}_i) > \Delta \; if \; \mathbf{x} \in \omega_i; \; d_i(\mathbf{x}_i) < \Delta \; if \; \mathbf{x} \in \omega_j \text{ with } j \neq i$$

Example

- Theshold $\Delta = 49$ with quadratic decision function:

$$if\ d(x) = x^2 > \Delta \text{ then } x \in \omega_1 \text{ else } x \in \omega_2$$





(a)Quadratic decision function (b) Logaritmic decision function

$d(x) = x^2$ $g(x) = ln(d(x))$

Generalised Decision Functions

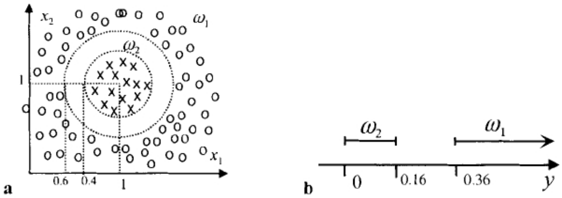
- Generalised Decision Function in a functional form:

$$d(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \cdots + w_k f_k(\mathbf{x}) + w_0 = \mathbf{w}'^* \cdot \mathbf{y}^*$$

$$\mathbf{y}^* = [1 \;\; f_1(\mathbf{x}) \, f_2(\mathbf{x}) \, \cdots \, f_k(\mathbf{x})]'$$

Two Class Discrimination Problem

- Original Feature Space

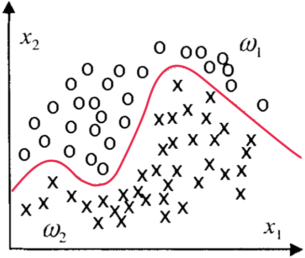


- Transformed one-dimensional Space

$$
\begin{aligned}
d(\mathbf{x}) &= (x_1 - 1)^2 + (x_2 - 1)^2 - 0.25 \\
\mathbf{y}^* &= [1 \quad y] \\
y &= f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2 \\
g(\mathbf{y}) &= [0.25 \quad 1]\mathbf{y}^* = y - 0.25
\end{aligned}
$$

Two Class Discrimination Problem

- Polynomial Decision Function

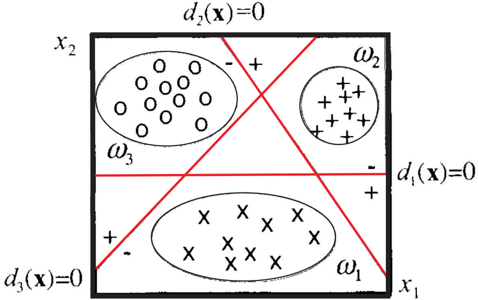


- Polynomial Degree = 4

$$d(\mathbf{x}) = w_{14}x_1^4 + w_{13}x_2^4 + w_{12}x_1^2x_2^2 + w_{11}x_1^3x_2 + w_{10}x_1x_2^3 + w_9x_1^3 + w_8x_2^3 + w_7x_1^2x_2 + w_6x_1x_2^3 + w_5x_1^2 + w_3x_1x_2 + w_2x_1 + w_1x_2 + w_0$$

Hyperplane Separability

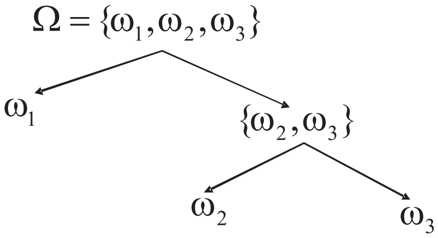
- Multiple class problem
  - Absolute separation (one class against the others)



$$R_i = \{\mathbf{x}; d_i(\mathbf{x}) > 0, d_j(\mathbf{x}) < 0,\ i, j = 1, \cdots, c,\ j \neq i\}$$

Hyperplane Separability

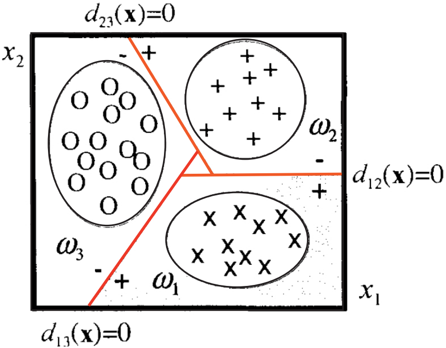
- Absolute separations corresponds to:
  - hierarchical classification



Hyperplane Separability

- Pairwise separation

$$\mathbf{d}_{ij}(\mathbf{x}) > 0, \forall \mathbf{x} \in \omega_i \text{ and } \mathbf{d}_{ij}(\mathbf{x}) < 0, \forall \mathbf{x} \in \omega_j \; (\mathbf{d}_{ij}(\mathbf{x}) = -\mathbf{d}_{ji}(\mathbf{x}))$$



$$R_i = \{\mathbf{x}; d_{ij}(\mathbf{x}) > 0, \; i, j = 1, \cdots, c, \; j \neq i\}$$

Feature Space Metrics

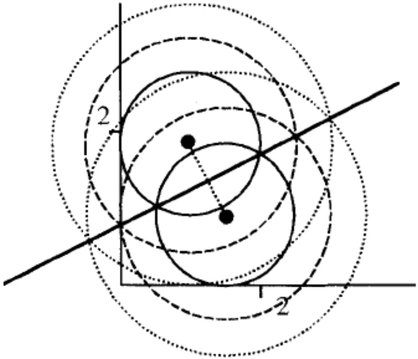
- Evaluation of patterns similarity:
  - Distance or norm:

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||$$

Euclidian norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \left(\sum_{i=1}^{d} (x_i - m_i)^2\right)^{1/2}$
Squared Euclidian norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \sum_{i=1}^{d} (x_i - m_i)^2$
City Block norm	$\|\|\mathbf{x} - \mathbf{m}\|\|_c = \sum_{i=1}^{d} \|x_i - m_i\|$
Chebychev norm	$\|\|\mathbf{x} - \mathbf{m}\|\|_c = \sum_{i=1}^{d} (x_i - m_i)$
Minkovsky norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \left(\sum_{i=1}^{d} (x_i - m_i)^2\right)^{1/p}$

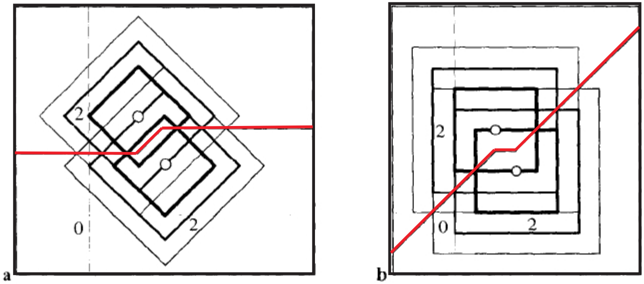
Equidistant surfaces for Euclidian metrics

- Decision Surface
  - Straight line is the set of equidistant points from the means $c = 2$ and $d = 2$



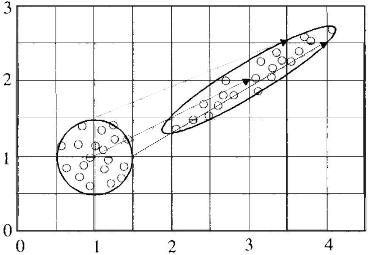
Equidistant Surfaces for City Block and Chebychev Metrics

- Decision surfaces Stepwise linear



- (a) CityBlock (b) Chebychev

Scaling by Linear Transformation



- Linear Transformation

$$\mathbf{y} = A\mathbf{x}$$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

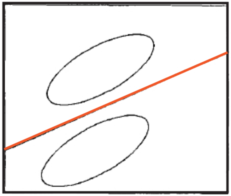
- Cluster structure changed form circular to ellipsoidal
  - circular class means: [1 1]
  - elliptic class means: [2 3]

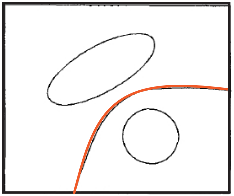
Generalisation do d-dimensional Space

- Mahalanobis distance

$$p(\mathbf{y}) = ||\mathbf{y} - \mathbf{m}||_m = \left((\mathbf{y} - \mathbf{m})' A \,(\mathbf{y} - \mathbf{m})\right)$$

linear decision surface quadratic decision surface

a


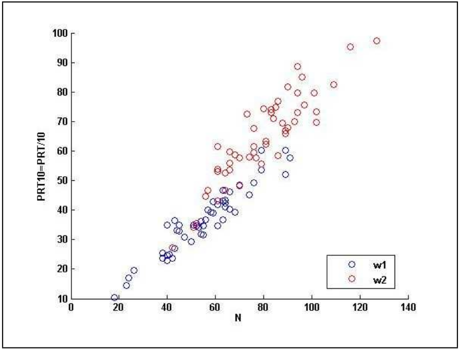
b


- **Hyperellispsoids** with Mahalanobis distance from the prototype

Data Scaling

- Data Set: *Cork Stoppers.xls*

Measurements made on binary images of cork stoppers defects in order to assess their quality

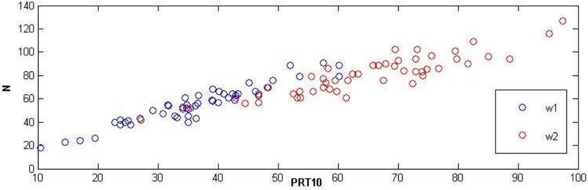


Cork Stop Features

Feature	Description
N	Total Number of Defects
PRT	Total perimeter of defects (in pixels)
ART	Total area of defects (in pixels)
PRM	Average perimeter of defects (in pixels) = PRT/N
ARM	Average area of defects (in pixels) = ART/N
NG	Number of bigger defects
PRTG	Total perimeter of big defects (in pixels)
ARTG	Total area of big defects (in pixels)
RAAR	Area Ratio of defects = ARTG/ART
RAN	Ratio of number of defects (NG/N)

Cork Stoppers Features

- The contribution of N to **class discrimination** seems negligible...
- Solution: equalising the features contribution



Feature Normalisation

- Normalisation

$$y_i = (x_i - m_i)/s_i$$

- Large variance features

$$s_i > 0$$

- Low variance features

$$s_i < 0$$

- Squared Euclidian distance

$$||\mathbf{y}|| = \sum_{i=1}^{d}(x_i - m_i)^2/s_i^2$$

Covariance Matrix

- Covariance between feature x_i and x_j estimated for n patterns:

$$c_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ki} - m_i)(x_{kj} - m_j)'$$

- Covariance Matrix

$$\mathbf{C} = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m}_k)(\mathbf{x}_k - \mathbf{m}_k)'$$

Feature Scaling

- Euclidian Distances

$$||\mathbf{x} - m_x|| \text{ and } ||\mathbf{y} - m_y||$$

are different before and after a linear transformation!

- Mahalanobis Distance

$$||\mathbf{x} - \mathbf{m}_m|| = (\mathbf{x} - \mathbf{m})\,\mathbf{C}^{-1}\,(\mathbf{x} - \mathbf{m})'$$

Mahalanobis Distance is **invariant** to scaling operations!

Orthonormal Transformation

- **Orthonormal transformation** is a linear transformation which allows to extract uncorrelated features from correlated ones
- We want to find the uncorrelated features $\mathbf{z}$ that maintain the same direction after transformation

$$\mathbf{y} = A\mathbf{z} = \lambda\mathbf{z}$$

- We have to solve:

$$(\lambda\mathbf{I} - A)\,\mathbf{z} = 0 \tag{1}$$

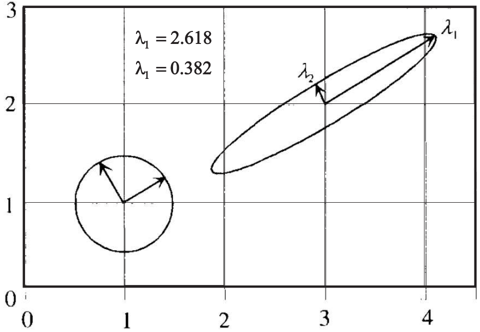
$$\Downarrow$$

λ *eigenvalues*

$\mathbf{z}$ *eigenvectors*

Principal Components

- Feature Selection



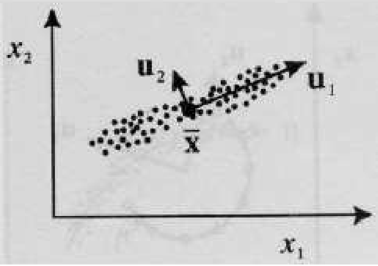
Eigenvectors of a linear transformation

- Principal component: each **eigenvector** corresponding to a **eigenvalue** with significant variance

$$\lambda_1^2/(\lambda_1^2 + \lambda_2^2) = 98\%$$

Geometric interpretation

- PCA projects the data along the directions where the data varies **the most**.
- These directions are determined by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues.
- The magnitude of the eigenvalues **corresponds to the variance** of the data along the eigenvector directions.

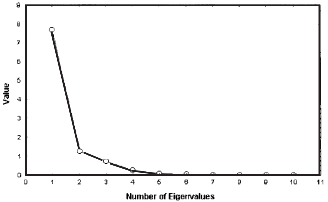


Principal Components

- Features that may exhibit high correlations among them and whose contribution to PR may vary substantially...

FACTOR ANALYSIS	Extraction: Principal components			
Value	Eigenval	% total Variance	Cumul. Eigenval	Cumul. %
1	7.672	76.72	7.67	76.72
2	1.236	12.36	8.91	89.08
3	.726	7.26	9.63	96.34
4	.235	2.35	9.87	98.68
5	.086	.86	9.95	99.54
6	.029	.29	9.98	99.83
7	.009	.09	9.99	99.91
8	.006	.06	10.00	99.98
9	.001	.01	10.00	99.99
10	.001	.01	10.00	100.00

Kaiser Criterion



Scree Test

Dimension Reduction: PCA

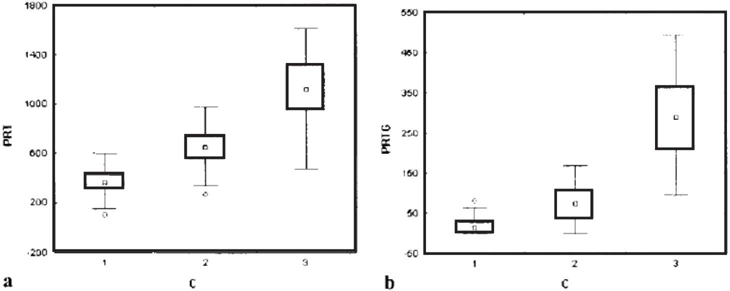
- Shortcomings/Observations:
  - Principal components are linear transformations of the original features (**problem of nonlinearity?**)
  - Principal components with negligible contribution to the overall variance may provide crucial contribution **pattern discrimination**
  - Difficult to attach any **semantic meaning** to principal components

Feature Assessment

- Assessing the **discriminative capability** of features:
  - Graphic Inspection
  - Distribution Model Assesement
  - Statistic Inference Tests

Graphic Model Inspection

- Allows to compare feature distributions for pattern classes

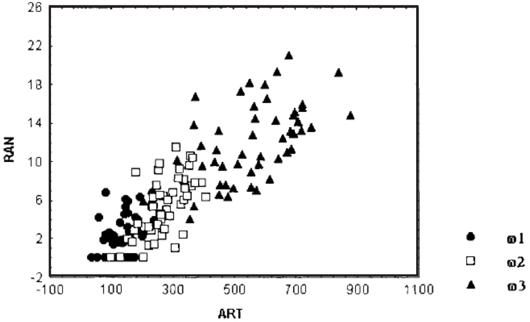


- Box Plots: **Statistical indicators**

  Median, Outliers, Extremes

Graphic Model Inspection

- **Topology** of classes and clusters

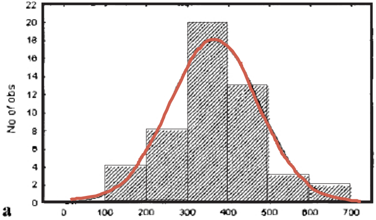


Scatter diagram

- A graphical model gives an indication of the amount of **overlapping of classes**

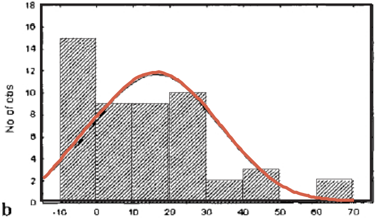
Distribution Model Assessement

PRT



a

ARTG



b

Statistical Inference

- Statistical tests (eg, **t-student, Anova**) for determining features discriminative power
- Feature Ranking: **Kruskal-Wallis** test (other?), sorts the feature values and assigns ordinal ranks
- Correlation matrix discard features **highly correlated**

Statistical Inference

- Cork Stoppers Problem : **Kruskal-Wallis** test

most discriminative feature ⟶ ART

less discriminative feature ⟶ N

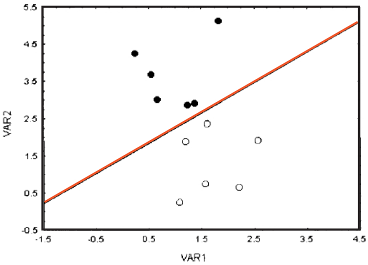
Feature	**H**
ART	121.6
PRTM	117.6
PRT	115.7
ARTG	115.2
ARTM	113.5
PRTG	113.3
RA	105.2
NG	104.4
RN	94.3
N	74.5

Statistical Inference

- The **Dimensional Ratio** Problem

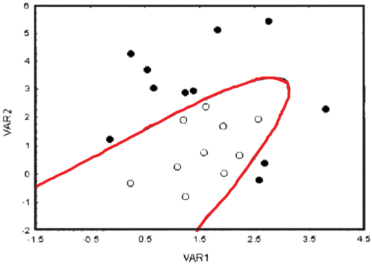
Linear discrimination

$n/d = 6$

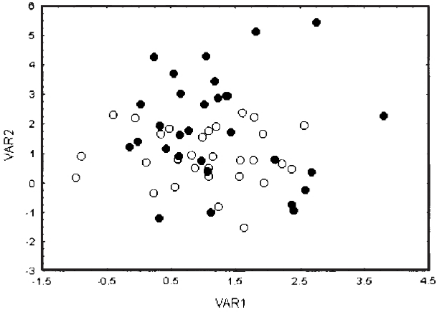


Quadratic discrimination

$n/d = 10$



Statistical Inference



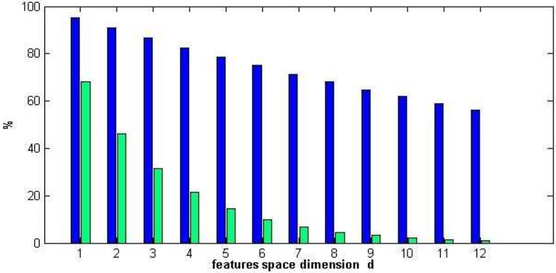
Scatter plot with **dimensionality ratio** $n/d = 30$

Curse of Dimensionality

- Use of low dimensionality ratio n/d can lead to total wrong conclusions about a classifier
- Problem of generalization on test data set?
- Consider each feature range divided into m intervals
- Each pattern location in m^d hypercubes
- Number of hypercubes grows exponentially with d - called **curse of dimensionality**

Curse of Dimensionality

- Percentage of normally distributed samples lying within **one standard deviation** neighborhood green bars and **two standard deviation** blue bars for several values of *d*

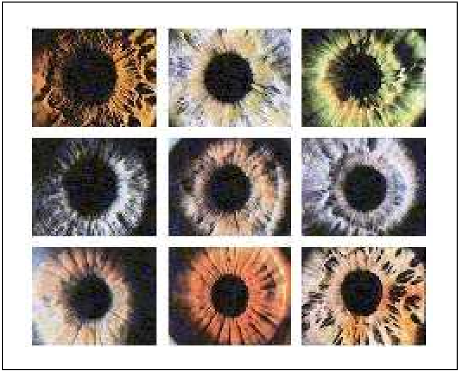


Pattern Recognition Techniques

Chapter 3: TRP Pattern Clustering

Chapter 3: Pattern Clustering

TRP: 2009-2010

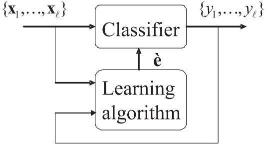


Data Clustering

The Reasons for Unsupervised Learning:

- Collecting and labelling a large amount of data may be pratically infeasible for some applications
- Given a small set of labeled samples initially, the system should generalize the models to a large data set, and track the change of the characteristics of the patterns over time
- In a general sense if we treat all the signals from vision, speech, smell, touch ...then the most of learning problems are unsupervised.

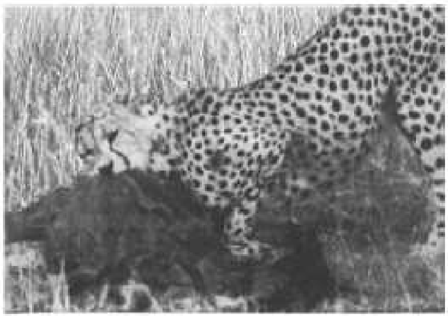
Unsupervised learning

- Input: training examples $\{\mathbf{x}_1 \cdots \mathbf{x}_l\}$without information about the hidden state.
- Clustering: goal is to find clusters of data sharing similar properties.
- A broad class of unsupervised learning algorithms:
- 

Classifier:$d : X \times \Theta \rightarrow Y$
Learning algorithm (supervised):
$(X \times Y)^l \rightarrow \Theta$

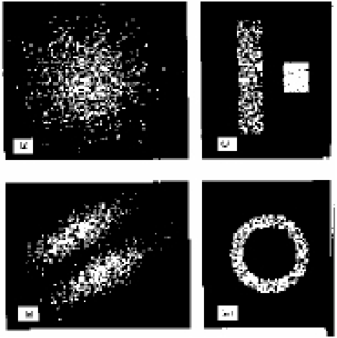
Example 1: Image Segmentation

- Given an image we are supposed to partition it into several classes and eachclass is a coherent pattern: Grass, Cheetah, Face, Bull and Ground in the sense that pixel intensities fit to a probabilistic model $c = 1, 2, 3, 4, 5$. Each model represent one type of pattern/concept.



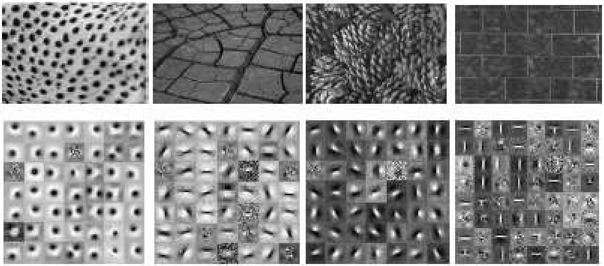
Example 2: Data Clustering

- There is no single crteria which is generally applicable. For isnstance some clusters are compact and round other are elongated and some may have holes. Therefore we need a broad range of models.



Example 3: Concept Discovery

- Sometimes we need to find concepts through data clustering. In the following, we find some basic elements for each type of texture by k-means clustering.



Data Clustering

What is Data Clustering?

- Given a set of n labelled examples $D = \{x_1, x_2, x_3 \cdots x_n\}$ in a d dimensional feature space we partition the data set D into a number of disjoint sets:

$$D = \cup_{j=1}^{K} D_j \qquad\qquad D_i \cap D_j = \emptyset \, \forall i \neq j$$

so that points in each set are coherent according to a criterion.

- We denote a partition by

$$\pi = (D_1, D_2, \cdots D_k)$$

- thus the problem is formulated as

$$\pi^* = \arg min\, f(\pi)$$

Data Clustering

- Hierarchical Algorithms
  use of linkage rules to produce hierarchical sequence of clustering solutions
- Centroid Adjustment Algorithms
  (eg, K-means clustering) adjust prototypes - centroids - describing the clusters

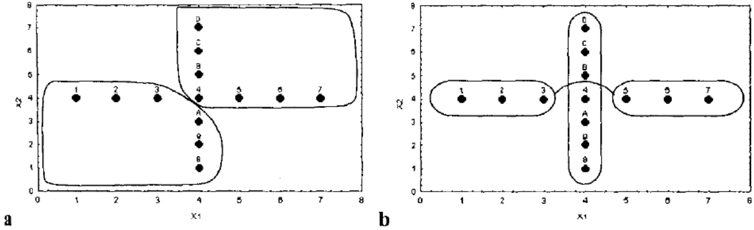
Hierarchical or Tree Clustering

- Metrics
- Standardization
- Linkage rules
- Dendrograms and Clustering Graphs
- Hierarchical sequence of clustering
- Examples

Different Metrics for Clustering?

Problem?

- Cluster.xls



Euclidian clustering   City-block clustering

Feature Space Metrics

- Evaluation of patterns similarity:
- Distance or norm $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||$

Euclidian norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \left(\sum_{i=1}^{d}(x_i - m_i)^2\right)^{1/2}$
Squared Euclidian norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \sum_{i=1}^{d}(x_i - m_i)^2$
City Block norm	$\|\|\mathbf{x} - \mathbf{m}\|\|_c = \sum_{i=1}^{d}\|x_i - m_i\|$
Chebychev norm	$\|\|\mathbf{x} - \mathbf{m}\|\|_c = \sum_{i=1}^{d}\max(x_i - m_i)$
Minkovsky norm	$\|\|\mathbf{x} - \mathbf{m}\|\| = \left(\sum_{i=1}^{d}(x_i - m_i)^p\right)^{1/r}$

Feature Space Metrics

- Pdist.m - Matlab

'euclidean' - Euclidean distance

'seuclidean' - Standardized Euclidean distance,

'cityblock' - City Block distance

'mahalanobis' - Mahalanobis distance

'minkowski' - Minkowski distance with exponent

'correlation' - One minus the sample correlation between

'hamming' - Hamming distance, percentage of coordinates

'chebychev' - Chebychev distance (maximum coordinate difference)

Error Minimization in Clustering

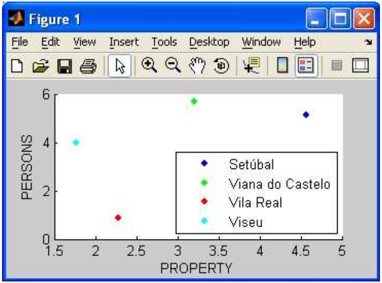
- Within-cluster average error
  - Error Minimization

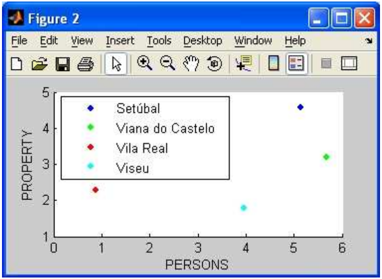
$$E = \sum_{i=1}^{c} \frac{1}{n} \sum_{\mathbf{x},\mathbf{y} \in \omega_i} distance(\mathbf{x}, \mathbf{y})$$

with n_i different patterns $\mathbf{x}, \mathbf{y}$ in cluster ω_i

Feature Standardization

- Visual Clustering of Crimes





- contradictory result?

Standardization methods

- In order to achieve scale invariance ...

$$
\begin{aligned}
y_i &= \frac{(x_i - m)}{s} \\
y_i &= \frac{(x_i - \min(x_i))}{(max(x_i) - min(x_i))} \\
y_i &= \frac{x_i}{(max(x_i) - \min(x_i))} \\
y_i &= \frac{x_i}{a}
\end{aligned}
$$

Standardization methods

- Disadavantages?
- Semantic information from the features can be lost
- Other?

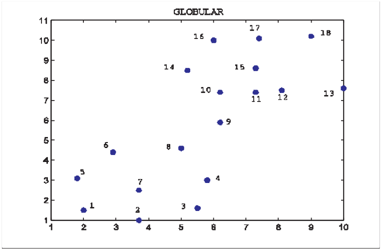
Tree Clustering

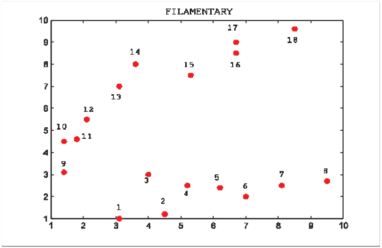
- Hierarchical or Tree clustering algorithms reveal the internal similarities of a given pattern set and structure these similarities hierarchically
  - Merging algorithm (bottom up)
  - Splitting algorithm (top down)

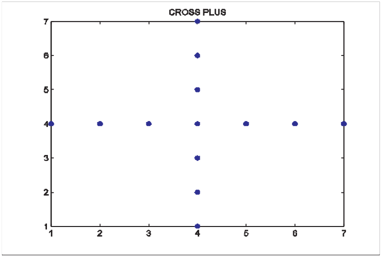
Tree Clustering

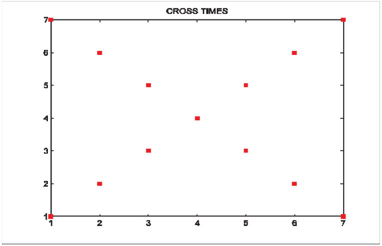
- Merging algorithm (bottom up)
  1. Given n patterns x_i consider $c = n$ singleton clusters $\omega_i = \{x_i\}$
  2. while $c \geq 1$
     - 2.1 Find the two nearest clusters ω_i and ω_j using a similarity measure rule
     - 2.2 Merge ω_i and ω_j: $\omega_{ij} = \{\omega_i, \omega_j\}$ obtaining $c - 1$ clusters
     - 2.3 Decrease c

Clusters Data Set



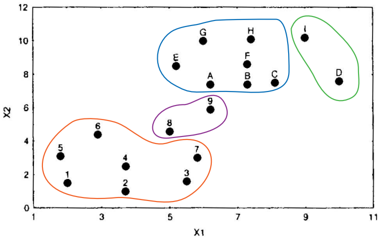


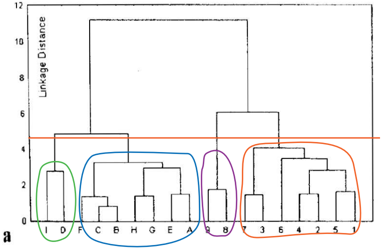




Clusters Tree Clustering

- Gobular data (Clusters.xls)





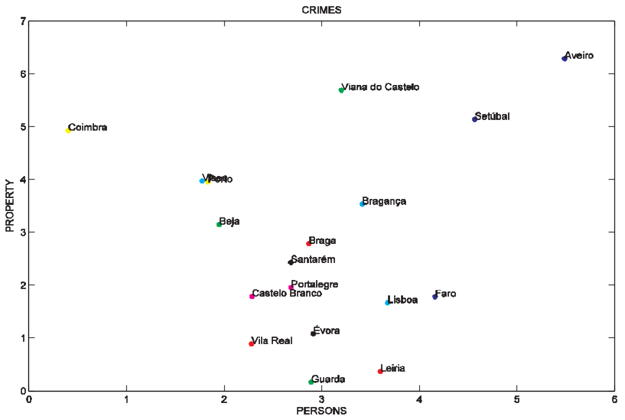
$\omega_1 = \{I, D\}$ $\omega_2 = \{A, B, C, D, E, G.H, F\}$

$\omega_3 = \{8, 9\}$ $\omega_4 = \{1, 2, 3, 4, 5, 6, 7\}$

Linkage distance 4 (red straight line)

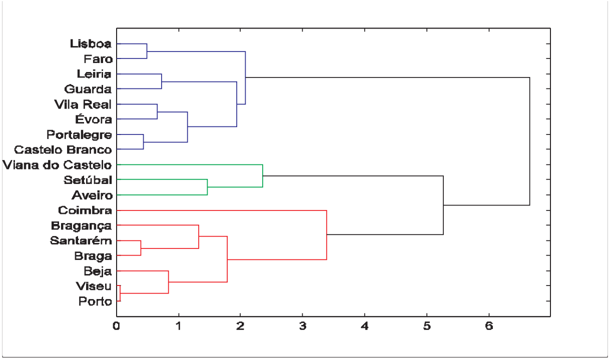
Crimes Data Set

- Crimes data set: Scatter plot



Crimes Tree Clustering

- Complete linkage
- Euclidian Distance



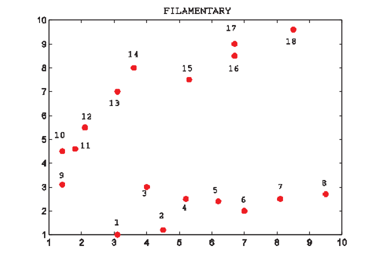
Cluster1= {Lisboa, Faro, Leiria, Guarda, Vila Real, Évora, Portalegre,Castelo Branco} Cluster2={Viana do Castelo, Setubal, Aveiro} Cluster3={Coimbra, Bragança, Santarém, Braga, Beja,Viseu,Porto }

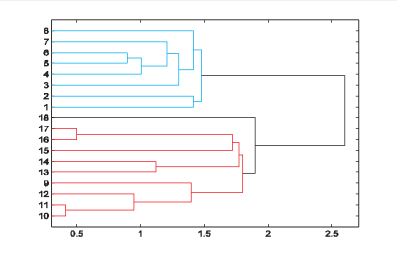
Linkage Rules

- Single linkage (NN - Nearest Neighbour)

$$d(\omega_i, \omega_j) = \min_{\mathbf{x}\in\omega_i, \mathbf{y}\in\omega_j} ||\mathbf{x} - \mathbf{y}||$$

- Filamentary data (Clusters.xls)



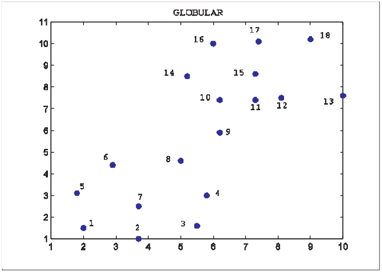


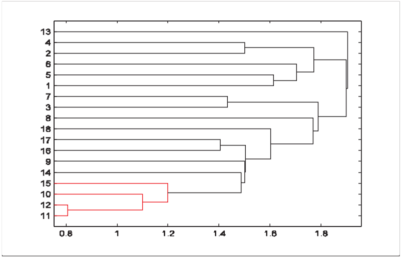
Linkage Rules

- Single linkage (NN - Nearest Neighbour)

$$d(\omega_i, \omega_j) = \min_{\mathbf{x} \in \omega_i, \mathbf{y} \in \omega_j} ||\mathbf{x} - \mathbf{y}||$$

- Filamentary data (Clusters.xls)



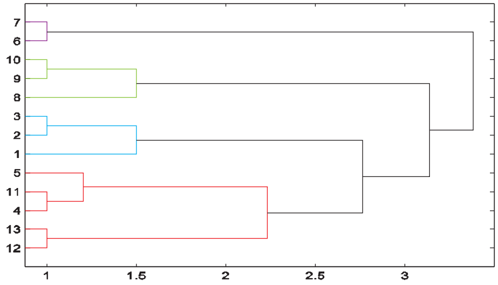


Linkage Rules

Single Linkage Rule (NN Neighbour)	$d(\omega_i, \omega_j) =_{\mathbf{x}\in\omega_i,\mathbf{y}\in\omega_j}^{\min} \|\|\mathbf{x}-\mathbf{y}\|\|$
Complete linkage (FN - Furthest Neighbour)	$d(\omega_i, \omega_j) =_{\mathbf{x}\in\omega_i,\mathbf{y}\in\omega_j}^{\min} \|\|\mathbf{x}-\mathbf{y}\|\|$
Unweighted average linkage between groups (UPGMA)	$d(\omega_i, \omega_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x}\in\omega_i} \sum_{\mathbf{x}\in\omega_j} \|\|\mathbf{x}-\mathbf{y}\|\|$
Weighted average linkage within groups (WPGMA)	$d(\omega_i, \omega_j) = \frac{1}{C(n_i+n_j,2)} \sum_{\mathbf{x},\mathbf{y}\in(\omega_i,\omega_j)} \|\|\mathbf{x}-\mathbf{y}\|\|$
Ward's method	$d(\omega_i, \omega_j) = \frac{1}{n_i+n_j} \sum_{\mathbf{x}\in(\omega_i,\omega_j)} \|\|\mathbf{x}-\mathbf{y}\|\|^2$

Dendrogram of +Cross Data

- Average linkage within groups (UPGMA)

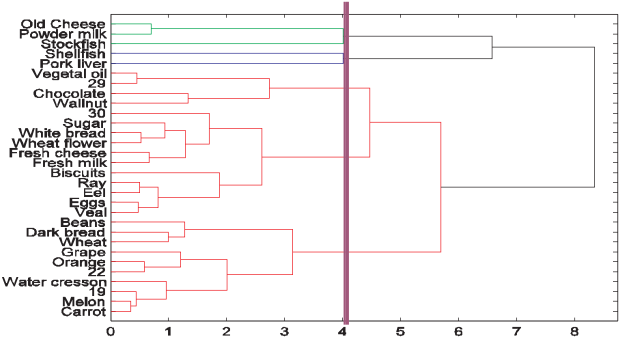


Tree Clustering Experiments

- Important to choose:
  - Appropriate metrics; and
  - Linkage rules

Tree Clustering Experiments

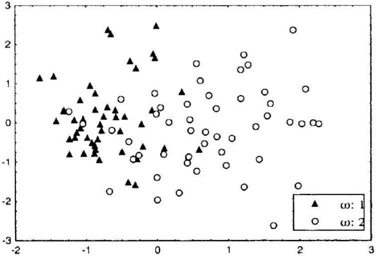
- Ward's method with Euclidian distance



Dimension Reduction

- Principal Component Analysis
  - Dimensionality reduction of the two first two classes of Cork stoppers (Cork_stoppers.xls) using two eigenvectors.
    - (a) Eigenvector Coefficients
    - (b) Eigenvector scatter plot

FACTOR ANALYSIS	Principal components	
Variable	Factor 1	Factor 2
ART	.121	-.242
N	.090	-.576
PRT	.113	-.383
ARM	.106	.296
PRM	.108	.246
ARTG	.123	.065
NG	.123	-.009
PRTG	.126	.037
RAAR	.110	.241
RAN	.116	.246

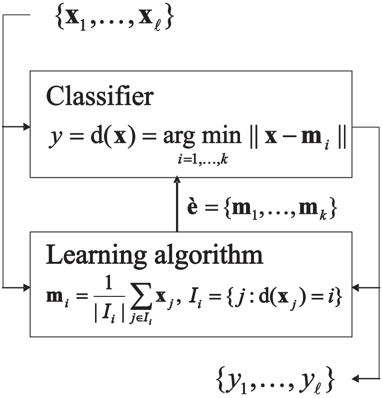


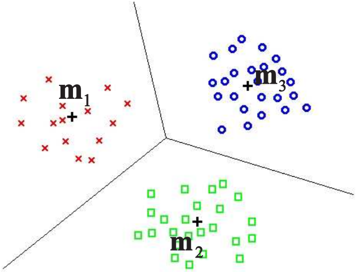
Data Clustering

- Centroid Adjustment Algorithms
  - (eg, K-means clustering)
  - adjust prototypes - centroids - describing the clusters
  - Performs iterative adjustment of c (previously defined)

K-Means Clustering

- Example of unsupervised learning algorithm: Goal is to minimize $E = \sum_{j=1}^{c} \sum_{\mathbf{x}_i \in \omega_j} ||\mathbf{x}_i - \mathbf{m}_j||^2$





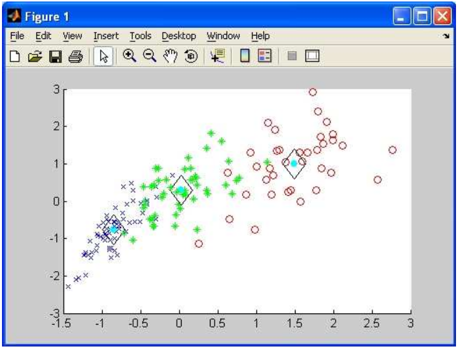
K-Means Clustering

- Algorithm k-means
  - Given c number of clusters, *maxiter*, max number of iterations, Δ threshold error. Assume c initial centroids $\mathbf{m}_j^{(k)}$ for the iteration$k = 1$
  - Assign each $\mathbf{x}_i$ to the cluster represented by the nearest $\mathbf{m}_j^{(k)}$
  - Compute for the previous partition the new centroid $\mathbf{m}_j^{(k+1)}$ and $E^{(k+1)}$
  - Repeat steps 2. and 3. until $k = maxiter$ or $|E^{(k+1)} - E^{(k)}| < \Delta$

$$E = \sum_{j=1}^{c} \sum_{\mathbf{x}_i \in \omega_j} ||\mathbf{x}_i - \mathbf{m}_j||^2$$

K-Means Clustering Cork

- centroids $\longrightarrow$ $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$
- classes: c_1, c_1, c_1

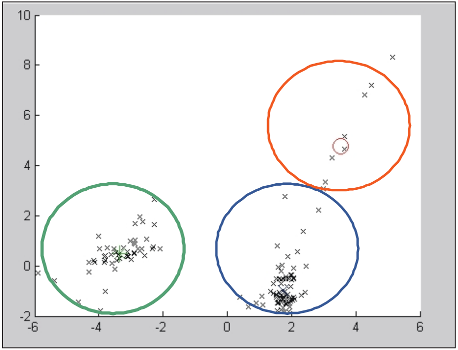


K-Means Clustering

- Variants of K-Means according to initial choice of **cluster centroids**
  - Choose patterns to be initial centroids
  - Choose the first c patterns to be the centroids
  - Sort distances between all patterns and choose uniformly the centroids
  - Choose patterns that maximize cluster distances

K-Means Clustering

- Rocks data set ($c = 3$ Clusters)
- Solution with two principal components



Cluster Validation

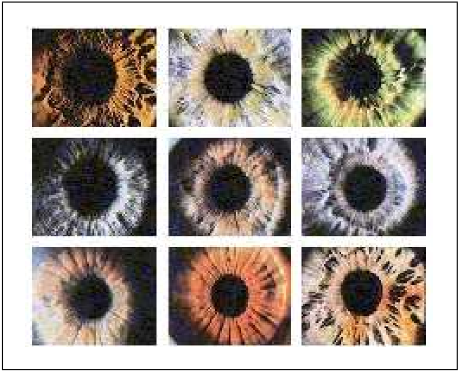
- Kruskal-Wallis test
  - Test the cluster solution and consider it acceptable if the corresponding test probability is below a certain confidence level
- Replication Analysis
  - Divide the original set into 2 data sets (randomly split ROCKS into $S1 = 66$ and $S2 = 68$ cases)
  - Cluster the 1st data set S1 and find centroids
  - Assign the data of the second data set $S2$ to the nearest centroids
  - Cluster the second data set $S2$
  - Compute a measure agreement between the clustering of S2 based on the nearest centroid of S1 and the direct clustering of $S2$.

Pattern Recognition Techniques

Chapter 4: TRP Statistical Linear Discriminants

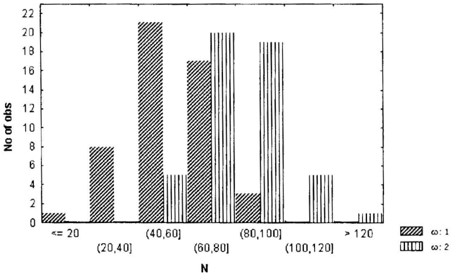
Chapter 4: Statistical Linear Discriminants

TRP: 2009-2010



Linear Discriminants

- Minimum Distance Classifier (Cork_stoppers.xls)
- Assume $\mathbf{x} = [\text{ N }]$, N - number of defects $d = 1$
- Prototypes
  1. $\mathbf{m}_1 \longrightarrow \omega_1$
  2. $\mathbf{m}_2 \longrightarrow \omega_2$
- Minimum distance classifier (template matching)



Rule: Assign each cork stopper to the nearest prototype!

Linear Discriminants

- Minimum Distance Classifier (Cork_stoppers.xls)
- Minimum distance classifier (template matching)

$$\textit{if}\, ||\mathbf{x} - [55.28]|| < ||\mathbf{x} - [79.54]|| \textbf{ then } \mathbf{x} \in \omega_1 \textbf{ else } \mathbf{x} \in \omega_2$$

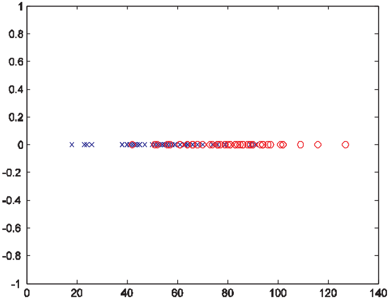
- using the value at half distance from the means

$$\textit{if}\, \mathbf{x} < 67.51 \textbf{ then } \mathbf{x} \in \omega_1 \textbf{ else } \mathbf{x} \in \omega_2$$

- The separating hyperplane is simply the point ($\mathbf{X} = 67.51$)

Linear Discriminant Classifier

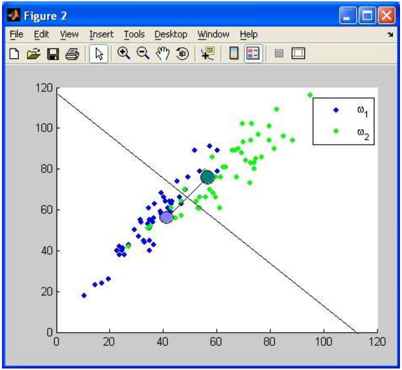
- Performance of the classifier:
  - compute the error rate in training set;
  - compare with predicted classifications



- Cork_stoppers data set
- Feature N
- Overall Error =0.23(23%)
- 18% ω_1; 28% ω_2
- CC =0.77(77%)

Linear Discriminant Classifier

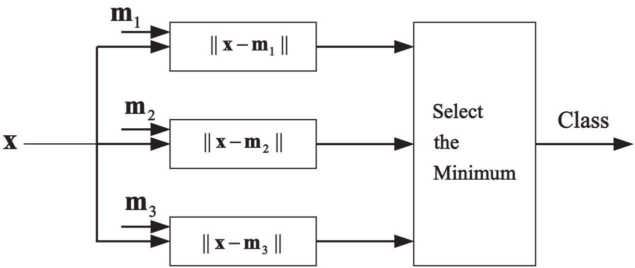
- dimensional space ($d = 2$) by adding feature PRT



- Draw the straight line (decision surface) equidistant from the means, perpendicular to the segment linking the means and passing at half distance
- Any pattern above straight line is ω_2, below ω_1. Assignement is arbitary in the boundary
- Overall Error $=0.18(18\%)$
- CC $=0.82(82\%)$

Linear Discriminant Classifier

- Minimum distance classifier (using any metric distance)



- feature vector $\mathbf{x}$, c classes $\omega_k(k = 1, 2, \cdots, c)$, $\mathbf{m}_k$ prototypes

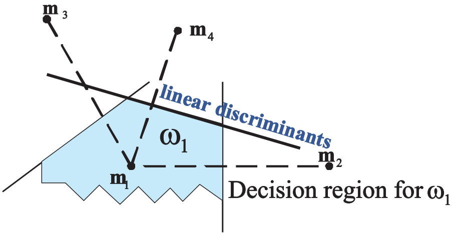
Euclidian Linear Discriminants

- Generalization for any d-dimensional space and any number of classes $\omega_k (k = 1, 2, ...c)$
- Squared Euclidian Distance

$$d_k^2(\mathbf{x}) = ||\mathbf{x} - \mathbf{m}_k||^2 = (\mathbf{x} - \mathbf{m}_k)'(\mathbf{x} - \mathbf{m}_k)$$

- Decision boundary (c=2)

$$(\mathbf{m}_1 - \mathbf{m}_2)'[\mathbf{x} - 0.5(\mathbf{m}_1 + \mathbf{m}_2)] = 0$$



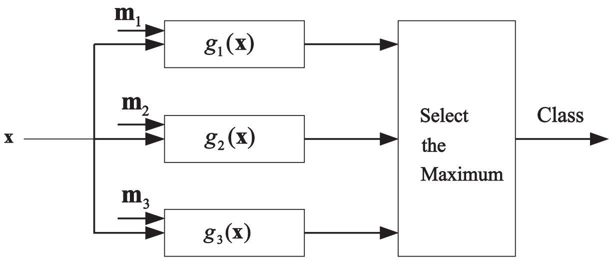
- Hyperplane perpendicular to the straight line joining $\mathbf{m}_1$ and $\mathbf{m}_2$ and intersects it in the middle point $0.5(\mathbf{m}_1 + \mathbf{m}_2)$

Euclidian Linear Discriminants

- Minimize $d_k(\mathbf{x})^2$ is equivalent to maximize discriminant function $g_k(\mathbf{x})$

$$g_k(\mathbf{x}) = \mathbf{m}'_k\mathbf{x} - 0.5||\mathbf{m}_k||^2 = \mathbf{w}'_k\mathbf{x} + w_{k,0}$$

with $\mathbf{w}_k = \mathbf{m}'_k$ and $w_{k,0} = ||\mathbf{m}_k||^2$



Mahalanobis Linear Discriminants

- Mahalanobis is a generalization of Euclidian distance

$$d_k^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_k)'\mathbf{C}^{-1}(\mathbf{x} - \mathbf{m}_k)$$

- As before, minimize $d_k(\mathbf{x})^2$ is equivalent to maximize discriminant function $g_k(\mathbf{x})$

$$g_k(\mathbf{x}) = \mathbf{w}_k'\mathbf{x} + w_{k,0}$$

$$\text{with } \mathbf{w}_k = \mathbf{C}^{-1}\mathbf{m}_k \text{ and } w_{k,0} = -0.5\mathbf{m}_k'\mathbf{C}^{-1}\mathbf{m}_k$$

Mahalanobis Linear Discriminants

- Considering again one Feature N in Cork Stoppers

$$\mathbf{m}_1 = [55.28] \qquad \mathbf{m}_2 = [79.54]$$

- Average variance $s^2 = 287.63$

$$\mathbf{w}_1 = \mathbf{m}_1/s^2 = [0.19219]; \qquad w_{k,0} = -0.5||\mathbf{m}_1||^2/s^2 = -6.00532$$

$$\mathbf{w}_2 = \mathbf{m}_2/s^2 = [0.27723]; \qquad w_{k,0} = -0.5||\mathbf{m}_2||^2/s^2 = -11.7464$$

$$g_k(\mathbf{x}) = \mathbf{w}_k'\mathbf{x} + w_{k,0} \quad \longleftarrow \quad \text{função discriminante } g_k(\mathbf{x})$$

- Suppose $\mathbf{x} = [N], N = 65$

$$g_1(\mathbf{x}) = 0.19219 \times 65 + (-6.00532) = 6.49 \text{ Decision is for class } \omega_1$$

$$\textbf{if} \quad g_1(\mathbf{x}) > g_2(\mathbf{x}) \quad \textbf{then} \quad \mathbf{x} \in \omega_1 \quad \textbf{else} \quad \mathbf{x} \in \omega_2$$

Mahalanobis Linear Discriminants

- Considering now two Features N and PRT10 in Cork Stoppers

$$\mathbf{C} = \begin{bmatrix} 287.6296 & 204.0698 \\ 204.0698 & 172.5529 \end{bmatrix} \qquad \mathbf{C}^{-1} = \begin{bmatrix} 0.0216 & -0.0255 \\ -0.0255 & 0.036 \end{bmatrix}$$

- We obtain the coefficients in the discriminant decision functions:

$$g_1(\mathbf{x}) = \mathbf{w}_1'\mathbf{x} + w_{1,0} = [0.2616 - 0.09783]\mathbf{x} - 6.1382$$

$$g_2(\mathbf{x}) = \mathbf{w}_2'\mathbf{x} + w_{2,0} = [0.2616 - 0.09783]\mathbf{x} - 6.1382$$

- Ex: suppose N=65 and PRT=520 pixels

$$g_1([6552]') = 5.78 < g_2([6552]) = 6.84 \longleftarrow \text{Decision is for class } \omega_2$$

Minimum Distance Classifier Types

Covariance	Classifier	Equipropability Surfaces	Discriminants
$\mathbf{C}_i = s^2\mathbf{I}$	Linear Euclidian	Hyperspheres	Hyperplanes orthogonal to line linking means
$\mathbf{C}_i = \mathbf{C}$	Linear Mahalanobis	Hyperellipsoids	Hyperplanes along regression line
$\mathbf{C}_i$	Quadratic Mahalanobis	Hyperellipsoids	Quadratic surfaces

Fisher's Linear Discriminant

- Class separability: two classes c_1 and c_2
- Within-class scatter matrix variance:

$$S_w = \sum_{k=1}^{2} \sum_{\mathbf{x} \in c_k} (\mathbf{x} - \mathbf{m}_k)(\mathbf{x} - \mathbf{m}_k)'$$

- In Between-class scatter matrix variance:

$$S_b = (m\_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)'$$

Goal: choose a direction (in the feature space) along which the distance of the means to S_w reaches a maximum

- Maximize the criterion:

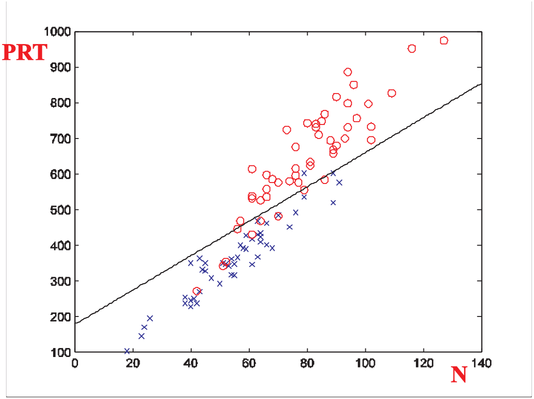
$$J(\mathbf{x}) = \frac{\mathbf{x} S_b \mathbf{x}'}{\mathbf{x} S_w \mathbf{x}'}$$

- Direction $\mathbf{x}$ that maximizes $J(\mathbf{x})$:

$$\mathbf{x} = S_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

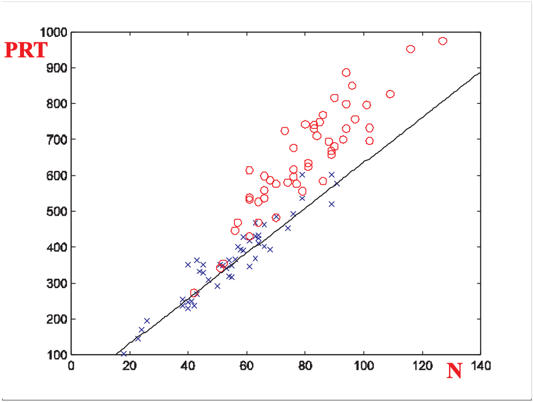
Fisher's Linear Discriminant

- Features N e PRT (Error =0.10 (10%))



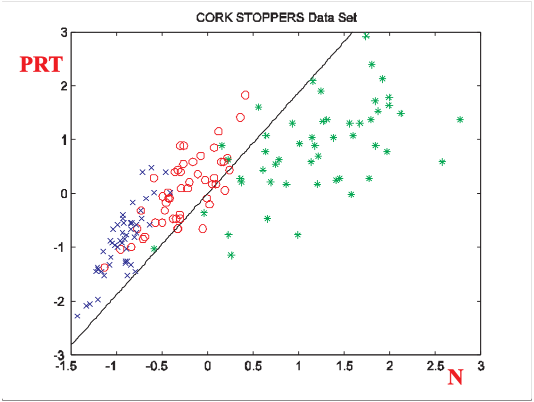
Perceptron

- Features N e PRT Error =0.16 (16%)



Multi-Perceptron

- Features N e PRT Normalised

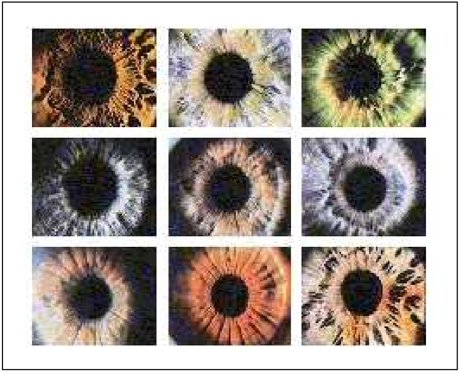


Pattern Recognition Techniques

Chapter 5: TRP Statistical Bayes Classification

Chapter 4: Bayesian Decision Theory

TRP: 2009-2010



Bayesian Decision Theory

- Fundamental statistical approach to problem classification.
- Quantifies the tradeoffs between various classification decisions using probabilities and the costs associated with such decisions.
  - Each action is associated with a cost or risk.
  - The simplest risk is the classification error.
  - Design classifiers to recommend actions that minimize some total expected risk.

Examples

- Ex 1: Cork Stoppers classification

$X = I$ is the image of cork stoppers,
$\mathbf{x}$ = (N (number of Defects), PRT (Perimeter of Defects),
ART (Area of Defects))
w is our belief what the cork stoppers type is
Ω^c={"Super Quality (S)", "Average Quality (A)", "Poor Quality (P)"}
α is a decision for the Cork Stoppers type, in this case $\Omega^c = W^\alpha$
$\Omega^\alpha = \{``S", ``A", ``P", \cdots\}$

Examples

- Ex 2: Fish classification

$X = I$ is the image of fish,
$\mathbf{x}$ =(brightness, length, # fins, $\cdots$)
w is our belief what the fish type is
Ω^c={"sea bass", "salmon", "trout", $\cdots$}
α is a decision for the fish type, in this case $\Omega^c = \Omega^\alpha$
Ω^α = {"sea bass", "salmon", "trout", $\cdots$}

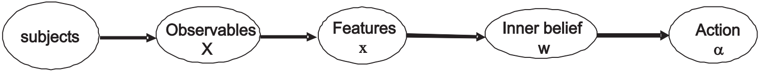
Examples

- Ex 2: Medical diagnosis

X= all the available medical tests, imaging scans that a doctor can order for a patient
$\mathbf{x}$= (blood pressure, glucose level, cough, x-ray, $\cdots$)
w is an illness type
Ω^c=“Flu”, “cold”, “TB”, “pneumonia”, “lung cancer”$\cdots$
α is a decision for treatment,
$\Omega^\alpha = \{$“Tylenol”, “Hospitalize”, $\cdots\}$

Diagram of pattern classification

Procedure of pattern recognition and decision making



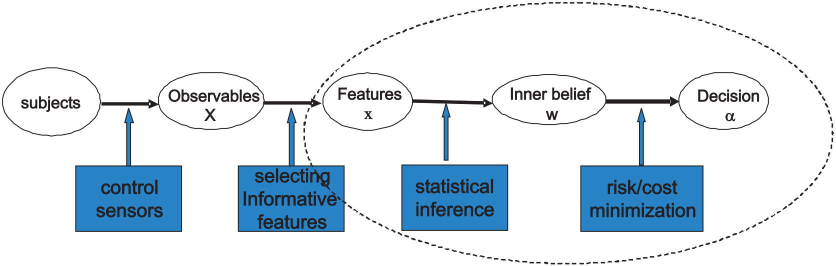
- X - all the observables using existing sensors and instruments
- x - is a set of features selected from components of X, or linear/non-linear functions of X.
- w - is our inner belief/perception about the subject class.
- α is the action that we take for x.

- We denote the three spaces by

$$x \in \Omega^d \qquad w \in \Omega^C \qquad \alpha \in \Omega^\alpha$$

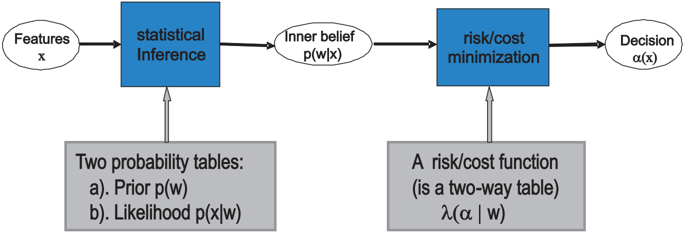
where $x = (x_1, x_2, \cdots, x_d)$ is a vector and w is the index of class, $\Omega^C = \{w_, w_2, w_k, \cdots\}$

Tasks



In Bayesian decision theory, we are concerned with the last three steps in the big ellipse assuming that the observables are given and features are selected.

Bayesian Decision Theory



- The belief on the class w is computed by the Bayes rule

$$p(w|x) = \frac{p(x|w)p(w)}{p(x)} \tag{1}$$

- The risk is computed by

$$R(\alpha_i/x) = \sum_{j=1}^{k} \lambda(\alpha_i|w_j)p(w_j|x) \tag{2}$$

Decision Rule

- A decision rule is a mapping function from feature space to the set of actions
$$\alpha(x) : \Omega^d \longrightarrow \Omega^\alpha \quad (3)$$
- We will show that randomized decisions are not optimal.
- A decision is made to minimize the average cost / risk,
$$R = \int R(\alpha(x)|x)p(x)dx \quad (4)$$
- It is minimized when our decision is made to minimize the cost / risk for each instance x.

$$\alpha(x) = \arg \underset{\Omega^\alpha}{\min} R(\alpha|x) = \arg \underset{\Omega^\alpha}{\min} \sum_{j=1}^{k} \lambda(\alpha|w_j)p(w_j|x) \quad (5)$$

Bayesian error

- In a special case, like cork stoppers classification, the action is classification, we assume a 0/1 error.

$$\lambda(\alpha_i|w_j) = 0 \quad \text{if} \quad \alpha_i = w_j \lambda(\alpha_i|w_j) = 1 \quad \text{if} \quad \alpha_i \neq w_j$$

- The risk for classifying x to class α_i is,

$$R(\alpha_i|x) = \sum_{w_j \neq \alpha_i} p(w_j|x) = 1 - p(\alpha_i|x) \tag{6}$$

- The optimal decision is to choose the class that has maximum posterior probability

$$\alpha(x) = \arg\min_{\Omega^\alpha}(1 - p(\alpha|x)) = \arg\max_{\Omega^\alpha} p(\alpha|x) \tag{7}$$

- The total risk for a decision rule, in this case, is called the Bayesian error

$$R = p(error) = \int p(error)|x)p(x)dx = \int (1 - p(\alpha(x)|x))p(x)dx \tag{8}$$

Discriminant functions

- To summarize, we take an action to maximize some discriminant functions

$$
\begin{aligned}
g_i(x) &= p(w_i|x) \\
g_i(x) &= p(x|w_i)p(w_i) \\
g_i(x) &= \log p(x|w_i) + \log p(w_i) \\
g_i(x) &= -R(\alpha_i|x)
\end{aligned}
$$

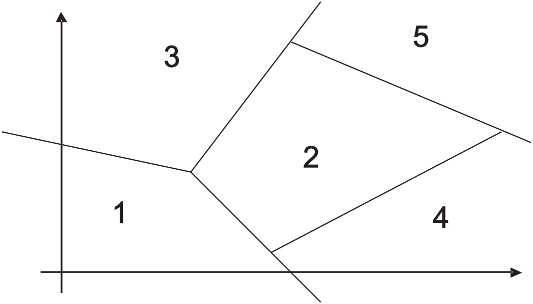
$$\alpha(x) = \arg\max\{g_1(x), g_2(x), \cdots, g_k(x)\} \quad (9)$$

Partition of feature space

$$\alpha(x) : \Omega^d \longrightarrow \Omega^\alpha$$

- The decision is a partition /coloring of the feature space into k subspaces

$$\Omega = \cup_{i=1}^{k} \Omega_i \qquad \Omega_i \cap \Omega_j = \emptyset, i \neq j$$



Example 1: Cork Stoppers

- Bayes Rule for Minimum Risk

  Super(S) ω_1

  Average(A) ω_2

  Number of cork stoppers of class ω_1: n_1

  Number of cork stoppers of class ω_2: n_2

  Total number of cork stoppers: $n = n_1 + n_2$
- Define the prior probabilities or prevalences

$$P(\omega_1) = n_1/n = 0.4 \qquad P(\omega_2) = n_2/n = 0.6$$

Conditional Probability

- $P(\omega_i|\mathbf{x})$ conditional probability of cork $\mathbf{x}$ belong to ω_i then

$$\begin{aligned} &\text{if} \quad P(\omega_i|\mathbf{x}) > P(\omega_2|\mathbf{x}) \quad \text{we decide } \mathbf{x} \in \omega_1 && (10) \\ &\text{if} \quad P(\omega_i|\mathbf{x}) < P(\omega_2|\mathbf{x}) \quad \text{we decide } \mathbf{x} \in \omega_2 && (11) \\ &\text{if} \quad P(\omega_i|\mathbf{x}) = P(\omega_2|\mathbf{x}) \quad \text{decision is arbitrary} && (12) \end{aligned}$$

simplifying

$$\text{if } P(\omega_i|\mathbf{x}) > P(\omega_2|\mathbf{x}) \text{ then } \mathbf{x} \in \omega_1 \text{ else } \mathbf{x} \in \omega_2 \qquad (13)$$

Bayes Rule

- Posterior Probabilities $P(\omega_i|\mathbf{x})$ computed by Bayes Law

$$P(\omega_i|\mathbf{x}) = \frac{\overbrace{p(\mathbf{x}|\omega_1)}^{likelihood}\overbrace{P(\omega_1)}^{prior}}{p(\mathbf{x})} \tag{14}$$

with $p(\mathbf{x}) = \sum_{i=1} cp(\mathbf{x}|\omega_1)P(\omega_1)$ total probability of $\mathbf{x}$.

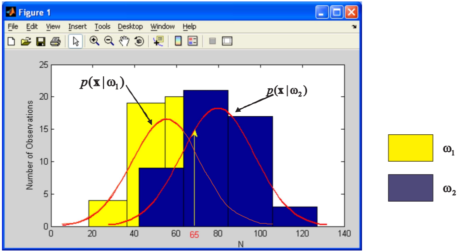
Discriminant Rule

$$\text{if } p(\mathbf{x}|\omega_1)P(\omega_1) > p(\mathbf{x}|\omega_2)P(\omega_2) \text{ then } \mathbf{x} \in \omega_1 \text{ else } \mathbf{x} \in \omega_2$$

$$\text{if } \nu(\mathbf{x}) = \underbrace{\frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x}|\omega_2)}}_{\text{Likelihood ratio}} > \underbrace{\frac{P(\omega_2)}{P\omega_1}}_{\text{Inverse of the priors}} \quad \text{then } \mathbf{x} \in \omega_1 \text{ else } \mathbf{x} \in \omega_2$$

- The decision depends then on how the likelihood ratio compares with the **inverse of prevalences (prior) ratio**

Likelihood Estimation



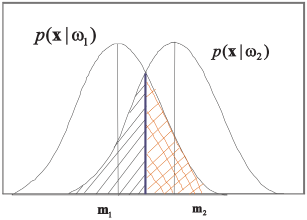
$$p(\mathbf{x}|\omega_1) = 0.833 \Rightarrow P(\omega_1)p(\mathbf{x}|\omega_1) = 0.333 \quad (15)$$

$$p(\mathbf{x}|\omega_1) = 0.696 \Rightarrow P(\omega_1)p(\mathbf{x}|\omega_1) = 0.418 \quad (16)$$

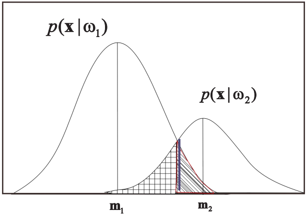
N=65 We decide class ω_2 although the likelihood of ω_1 is higher

Classification Risks

Equal prevalences



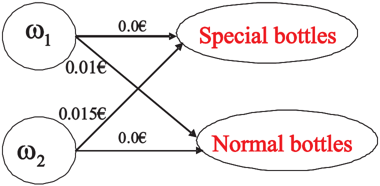
Unequal prevalences



- shaded **red** Errors class ω_1 shaded **black** Errors class ω_2

Bayes Rule for Minimum Risk

$$\begin{aligned} \text{cost of } \omega_1(super - S) &= 0.025E \\ \text{cost of } \omega_2(average - A) &= 0.015E \end{aligned}$$



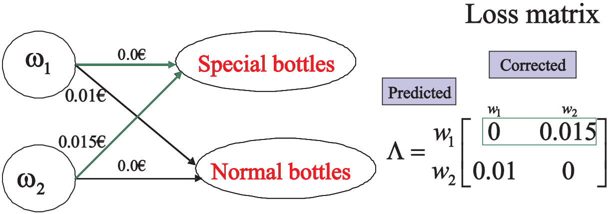
Loss matrix

$$\Lambda = \begin{bmatrix} 0 & 0.015 \\ 0.01 & 0 \end{bmatrix}$$

$\lambda_{ij} = \lambda(\alpha_i|\omega_j)$ loss associated with action α_i when the correct class is ω_j

Bayes Rule for Minimum Risk

$$\begin{aligned} \text{cost of } \omega_1(super - S) &= 0.025E \\ \text{cost of } \omega_2(average - A) &= 0.015E \end{aligned}$$



$\lambda_{ij} = \lambda(\alpha_i|\omega_j)$ loss associated with action α_i when the correct class is ω_j

Bayes Rule for Minimum Risk

Let us assume: $\omega_1 = super(S)$ $\omega_2 = Average(A)$

- we define action: $\alpha_i = \{SB, NB\}$

$$\begin{aligned}[R(\alpha_1|\mathbf{x})] &= [\lambda(SB|S) \;\; \lambda(SB|A)]\,[P(S|\mathbf{x})] \\ [R(\alpha_2|\mathbf{x})] &= [\lambda(NB|S) \;\; \lambda(NB|A)]\,[P(A|\mathbf{x})]\end{aligned}$$

$$\begin{aligned}R(\alpha_1|\mathbf{x}) &= R(SB|\mathbf{x}) = \lambda(SB|S)P(S|\mathbf{x}) + \lambda(SB|A)(P/A|\mathbf{x}) \\ R(\alpha_1|\mathbf{x}) &= 0.015P(A|\mathbf{x})\end{aligned}$$

$$\begin{aligned}R(\alpha_2|\mathbf{x}) &= R(NB|\mathbf{x}) = \lambda(NB|S)P(S|\mathbf{x}) + \lambda(NB|A)(P/A|\mathbf{x}) \\ R(\alpha_2|\mathbf{x}) &= 0.01P(S|\mathbf{x})\end{aligned}$$

- In the risk evaluation the only influence is thus from the wrong decisions

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{k} \lambda(\alpha_i|\omega_j)P(\omega_j|\mathbf{x}) \qquad (17)$$

Bayes Rule for Minimum Risk

- Thus
  - $R(\alpha_1|\mathbf{x})$ - Risk of taking decision α_1 is influenced only by a wrong classification in ω_1 (i.e., when the correct class is ω_2)
  - $R(\alpha_2|\mathbf{x})$ - Risk of taking decision α_2 is influenced only by a wrong classification in ω_2 (i.e., when the correct class is ω_1)
- Now:
  - We are interested in minimizing the risk for an arbitarrily large number of cork stoppers The Bayes Rule for Minimum Risk achieves this through minimization of conditional risks

Bayes Rule for Minimum Risk

- Assume that wrong decisions imply the same loss:

$$\lambda_i = \lambda(\alpha_i|\omega_j) = \begin{cases} 0 \text{ if } i = j \\ 1 \text{ if } i \neq j \end{cases}$$

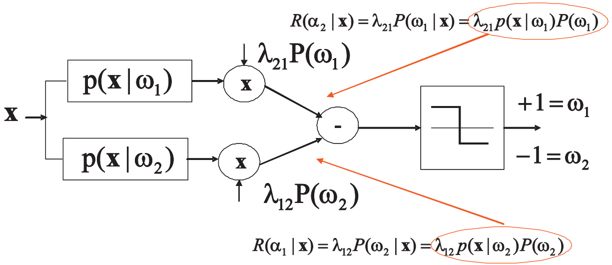
- Since posterior probabilities add up to 1, we have to minimize:

$$R(\alpha_i|\mathbf{x}) = \sum_{j \neq i} P(\omega_i|\mathbf{x}) = 1 - P(\omega_i|\mathbf{x})$$

- This corresponds to **maximize** the posterior $P(\omega_i|\mathbf{x})$
- Decide ω_i if $P(\omega_i|\mathbf{x}) > P(\omega_i|\mathbf{x}),\ \forall j \neq i$
- The Bayes decision rule for minimum risk when correct decisions have zero loss and wrong decisions have equal losses, corresponds to select class with maximum posteriori probability $\longrightarrow$ MAP.

Discriminant Decision Function

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



- Implementation of Bayesian decision rule for two classes with different loss factors for wrong decisions

Average Bayesian Risk

- c=2 (two class problem)

$$R = \int_{R_1} \lambda_{12} P(\omega_2|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{R_2} \lambda_{21} P(\omega_1|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- Ω (set of classes)

$$R = \sum_{\omega_i \in \Omega_X} \int \lambda(\alpha(\mathbf{x})|\omega_i) P(\omega_i|\mathbf{x}) d\mathbf{x}$$

Normal Bayesian Classification

- A normal likelihood for class ω_i is expressed by the following **pdf** (probability distribution function)

$$P(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{1/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}-\mu_i\right)'\Sigma_i^{-1}\left(\mathbf{x}-\mu\right)\right) \tag{18}$$

Normal Bayesian Classification

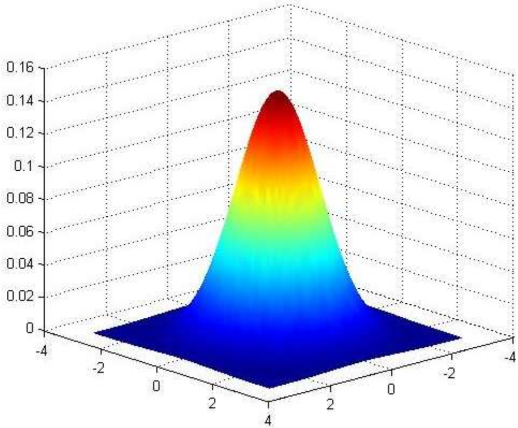
- A normal likelihood for class ω_i is expressed by the following **pdf** (probability distribution function)

$$P(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{1/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)' \Sigma_i^{-1} (\mathbf{x}-\mu)\right) \tag{19}$$

- Distribution Parameters

True Mean	Covariance
$\mu_i = E_i[\mathbf{x}]$	$\Sigma_i = E_i(\mathbf{x}-\mu_i)'(\mathbf{x}-\mu_i)$

Normal Bayesian Classification



- The Bell shaped surface of a two-dimensional normal distribution

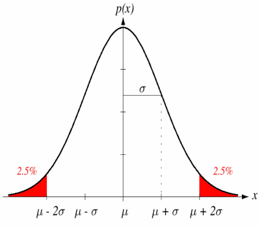
Likelihood of Θ

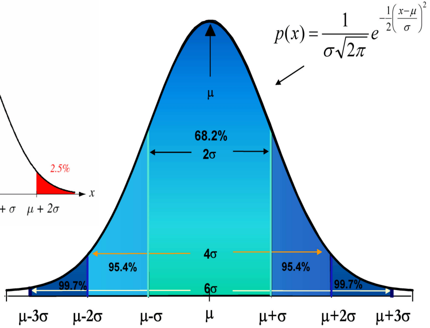
- Given a training set of Patterns $T = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_k\}$ characterized by a distribution with pdf $p(T/\Theta)$ where Θ is a parameter vector of the distribution (**mean** and **covariance**) we can obtain estimates of Θ maximizing $P(T/\Theta)$ given by:

$$p(T|\Theta) = \prod_{i=1}^{n} P(\mathbf{x}|\Theta)$$

Normal Probability Distribution

- By far the most important and most commonly observed (cont.) probability distribution

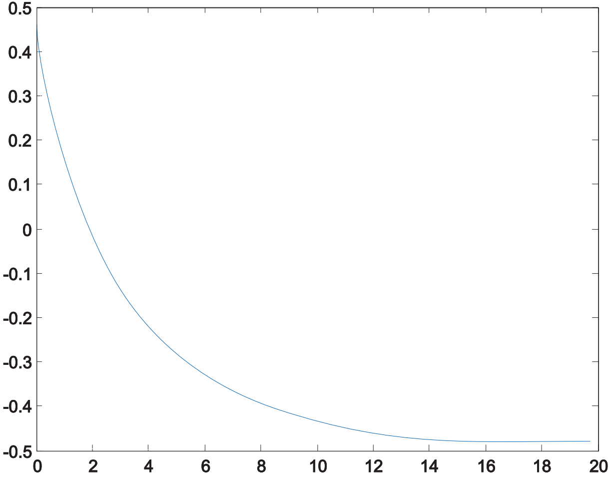




-

Bayesian Error

- Bayesian Error Pe



- with normal distributions and equal prevalences and covariance

Bayesian Theoretical Error

$$P_e = 1 - erf(\delta/2)$$

with $erf(x) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{x} \exp(-t^2/2)dt$ known as error funtion

- Bhattacharyya Distance, a Mahalanobis distance of the difference of the means, reflecting the class separability

$$\delta^2 = (\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2)$$

Pattern Recognition Techniques

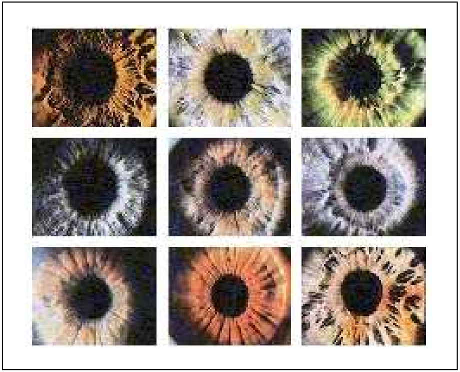
Chapter 6: TRP Non-Parametric Methods

Pattern Recognition Techniques

Chapter 6: TRP Non-Parametric Methods

Chapter 6: Non-Parametric Methods

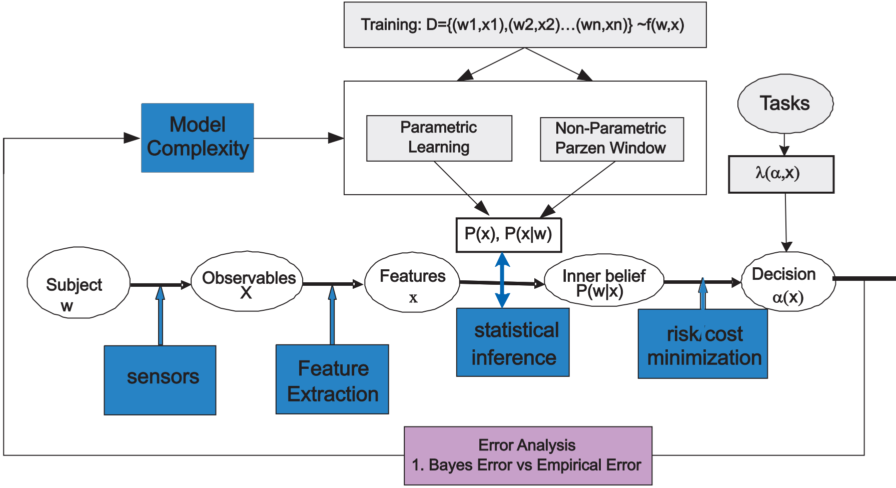
TRP: 2009-2010



Model Free Techniques (Non-Parametric Learning)

- Methods do not make any assumptions about the underlying pattern distributions
  - Parzen Window
  - K-Nearest Neighbours Method (K-NN)
  - ROC Curves
- Parzen Window and K-NN based on the idea of estimating pdf of the pattern distributions

Parametric and NonParametric Learning



K-Nearest Neighbours Method

- Fix the number of points $k(n)$ that exist in a certain region centred on a feature vector $\mathbf{x}$. The region has a Volume $V(n)$ and the **pdf** estimate is:

$$p(\mathbf{x}, n) \approx \frac{k(n)/n}{V(n)}$$

- If there are a few points around $\mathbf{x}$ we obtain a low density value; if we there are many points around $\mathbf{x}$ yields to high density value

K-NN Classifier

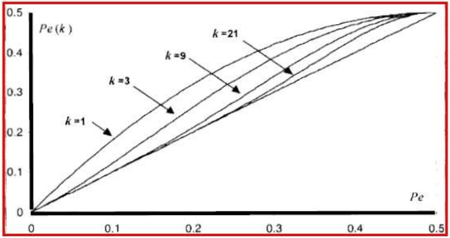
- Nearest Neighbour Rule:
  1. Consider $k(n)$ points that are the nearest neighbours of $\mathbf{x}$, using a certain distance metric
  2. The classification of $\mathbf{x}$ is the class label that is found in majority among the $k(n)$ neighbours

K-NN Classifier

- When applying the k-NN method, we are interested in the Performance Error Pe(k) for an arbitarily large population, i.e, $n \rightarrow 8$ For e.g. with $k = 1$

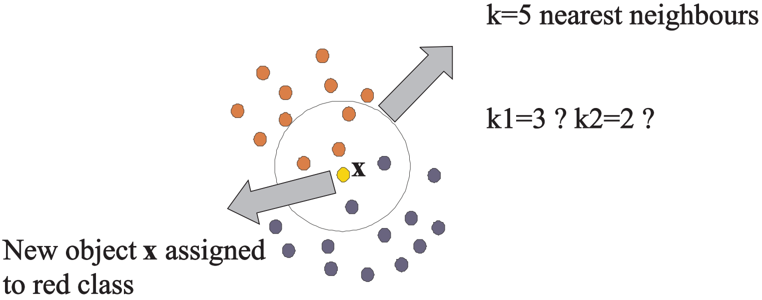
$$P_e(k) \leq 2P_e(1 - P_e)$$

where P_e is the Performance Bayes Error



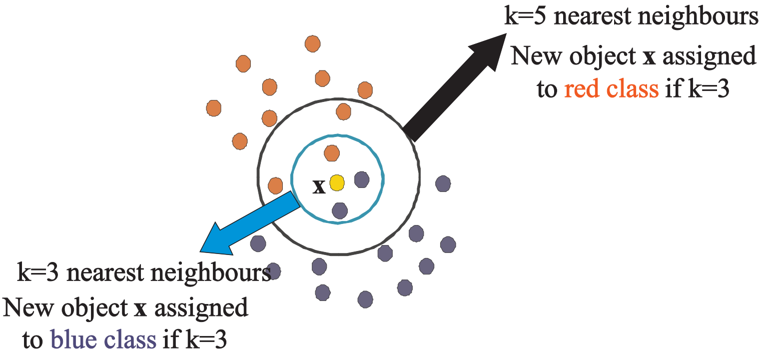
K-Nearest Neighbours Method

- Example 1:

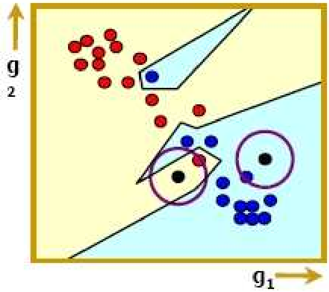


K-Nearest Neighbours Method

- Example 2:



K-NN Classifier

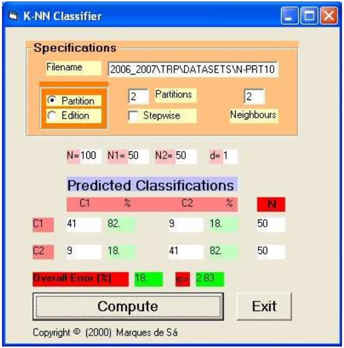


- **1-NN**
  - Set class of the new sample to the class of the nearest neighbout in the training set
- **2-NN**
  - Find *k* nearest training samples
  - Set class of the new sample to the class that os most frequent present within these *k* nearest neighbours

K-NN Classifier

- **Algorithm**
  1. The training examples are vectors in a multidimensional feature space. A point in the space is assigned to the class c if it is the most frequent class label among the k nearest training samples. Usually Euclidean distance is used.
  2. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.
  3. In the actual classification phase, the test sample (whose class is not known) is represented as a vector in the feature space.
  4. Distances from the new vector to all stored vectors are computed and k closest samples are selected.
  5. To classify the new vector to a particular class, the most common class amongst the K nearest neighbors is assigned to it.

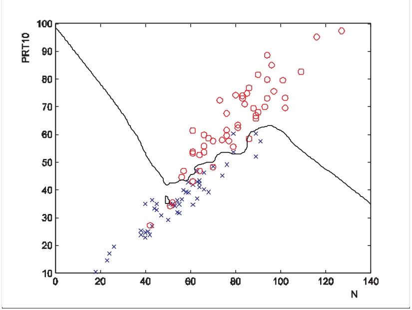
K-NN Cork Classifier



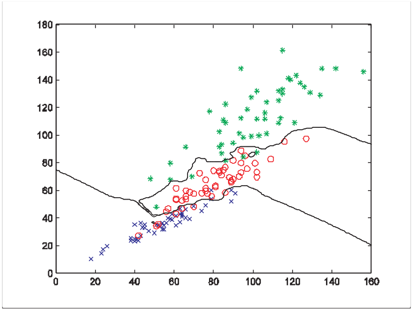
- *N*-PRT10.txt
  - Format:
    - n - number of patterns
    - N_1 - number of patterns of first class
    - D - dimension
    - ...
    - N lines with d values, first n_1 lines for the first class, followed by $n-_1$ lines for the second class

K-NN Cork Classifier

$K = 3 \;\; \omega_1, \omega_2$
$Pe(d) = 0.09(9\%)$



$K = 3 \;\; \omega_1, \omega_2, \omega_3$
$Pe(d) = 0.086(8.6\%)$



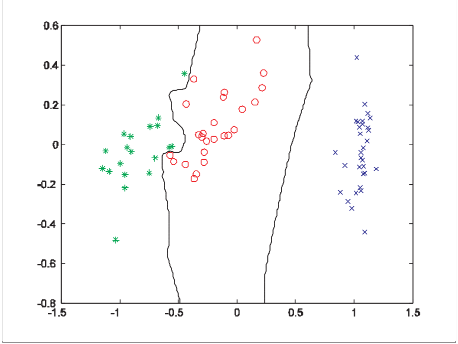
K-NN Cork Classifier

- Iris Data Set
- Three Classes $\omega_1, \omega_2, \omega_3$

Iris_setosa	Iris_versicolor	Iris_virginica
		

K-NN Iris Classifier

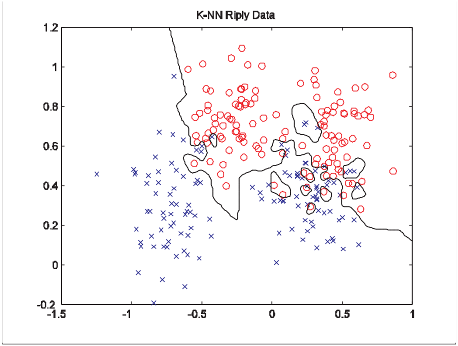
- Iris data set



- $K = 2, \omega_1, \omega_2, \omega_3$
- $Pe(d) = 0.00(0\%)$
- $Pe(t) = 0.04(4\%)$

K-NN Riply Classifier

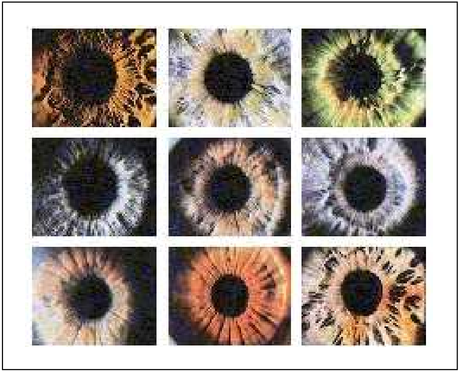
- Riply data set



- $K = 8, \omega_1, \omega_2$
- $Pe(d) = 0.088(8.8\%)$
- $Pe(t) = 0.116(11.6\%)$

Chapter 6: ROC Curves

TRP: 2009-2010



ROC Curves

- Receiver Operator Characteristics (ROC) curves are interesting tools in two-class problems
- For example in situations where we want to detect rarely ocurring events such as signal, a disease, etc.
- Let's call the absence of the event (**Normal**) and the occurrence of the rare event (**Abnormal**)
- Classification Matrix (Confusion Matrix)

Corrected (rows) / Predicted (columns)

	A	N
A	a	b
N	c	d

ROC Curves

True Positive Ratio	TPR=a/(a+b)	sensitivity	How sensitive is our decision method in detection of rare event	Rarely misses Event A
True Negative Ratio	TNP=d/(c+d)	specificity	How specific is our decision method in detection of rare event	Low rate of False Alarms
False Positive Ratio	FPR=c/(c+d)	1-specificity		
False Negative Ratio	FNR=b/(a+b)	1-sensitivity		

		A	N	
Corrected	A	a	b	Predicted
	N	c	d	

b - False Negative c - False Positive

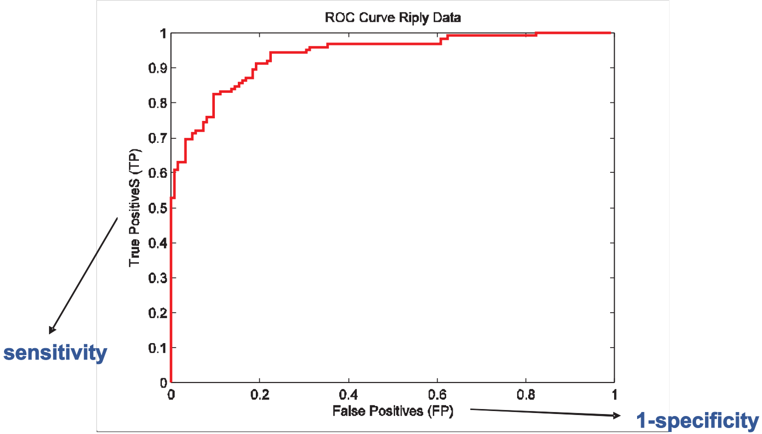
ROC Curves

- The ROC curve plots the proportion of correct responses (hits) against the false positives as the threshold Δ changes.
- Requires altering the loss function of observers by rewards and penalties
- The ROC curve gives information which is independent of the observer's loss function.

ROC Curves

- ROC curve represents a trade-off between sensitivity and specificity. (If sensitivity increases the specificity decreases and vice versa.)
- ROC curves start $(0, 0)$ and end at $(1, 1)$
- A perfect classifier corresponds to point $(0, 1)$; An arbitrary (random) classifier corresponds to the diagonal (45 degree line)

ROC Curves on Riply Data

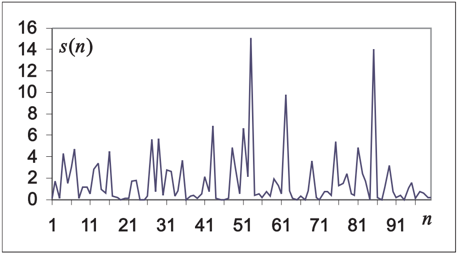


ROC Curves

- SignalNoise.xls data set (random noise plus signal impulses)

if $s(n) > \Delta$ then we decide "impulse"

we decise "noise"



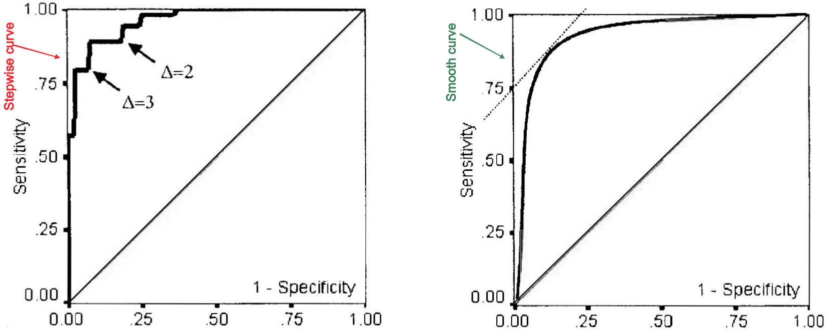
ROC Curves on SignalNoise Data

- SignalNoise.xls data set (random noise plus signal impulses)

Threshold	Sensitivity	Specificity
1	0.90	0.66
2	0.80	0.80
3	0.70	0.87
4	0.70	0.93

- There is clear a compromise to be made between **sensitivity** and **specificity**

ROC Curves on Signal Noise data



(a)
Eight threshold values
($\Delta = 2$ and $\Delta = 3$)

(b)
A large number of threshold values (expected curve)

ROC Curves

- How to choose the best threshold?
- Let us assume that: sensitivity - $s(\Delta)$; specificity - $f(\Delta)$
- Let us represent this as a cost decision issue

$$
\begin{aligned}
R &= \lambda_{aa}P(A)s(\Delta) + \lambda_{an}P(A)(1 - s(\Delta)) + \\
&\quad \lambda_{na}P(N)f(\Delta) + \lambda_{nn}P(N)(1 - f(\Delta))
\end{aligned}
$$

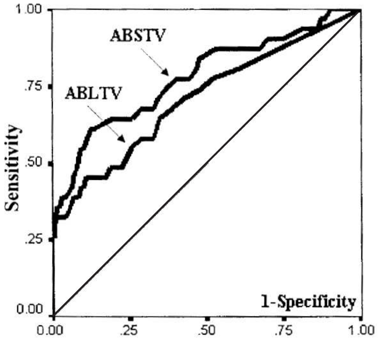
$$
\begin{aligned}
R &= s(\Delta)(\lambda_{aa}P(A) - \lambda_{an}P(A)) + f(\Delta)(\lambda_{na}P(N) + \lambda_{nn}P(N)) \\
&\quad +constant
\end{aligned}
$$

- In order to obtain the best threshold we minimize the risk by differentiating and equalizing to zero, obtaining
- Point in the ROC curve with slope given by:

$$
\frac{ds(\Delta}{df(\Delta)} = \frac{(\lambda_{nn} - \lambda na)}{(\lambda_{aa} - \lambda an)}
$$

ROC Curves

- Another application of ROC Curves is in the comparision of classification methods
  - FHR Apgar Data Set
    - several parameters from foetal heart rate (FHR) creates Apgar index
  - Measurements of:
    - ABSTV - % Abnormal short term variability
    - ABLTV - % Abnormal long term variability

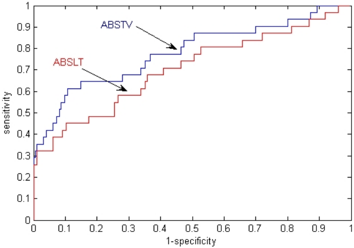


- Question: Which of these parameters is better in clinical practice to discriminate:
  - Apgar $> 6 \longmapsto$ Normal Situation
  - Apgar $= 6 \longmapsto$ Abnormal or Suspect Situation

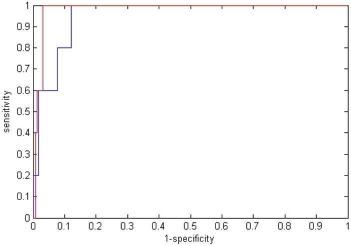
ROC Curves

- FHR Apgar Data Set

Apgar1 index

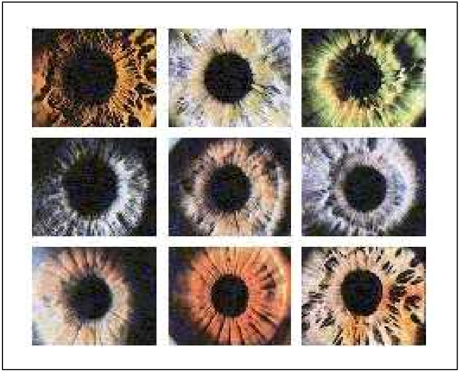


Apgar5 index



Chapter 6: Classifier Evaluation

TRP: 2009-2010



Error Estimation

- Pe- Probability of an Optimum Bayesian Classifier
- $Pe_d(n)$ - Training (design) set estimate of Pe for n patterns
- $Pe_t(n)$ - Test set estimate of Pe for n patterns
- $Pe_d(\infty) = Pe$ and
- $Pe_t(\infty) = Pe$ with increasing n patterns

In normal practice these probabilities are not known so we compute estimates $\widehat{Pe}_d(n)$ and $\widehat{Pe}_t(n)$ of misclassified patterns

Dimensionality Ratio

- The dimensionality ratio is an essential issue to design a classifier
- An adequately high dimensionality ratio will garantee that the designed classifierhas reproducible results,i.e., performs equally well when presented with **new patterns**

Probability of Misclassified Patterns

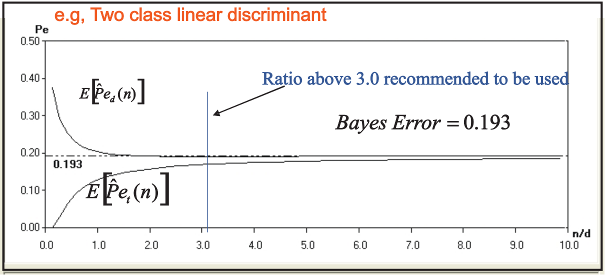
- Probability of k misclassified patterns out of n for a classifier with Pe given by the binomial law:
$P(k) = C(n, k)Pe^k(1 - Pe)^{n-k}$

Maximum Likelihood $\widehat{Pe} = \frac{k}{n}$

Standard Deviation $s = \sqrt{\frac{Pe(1-Pe)}{n}}$

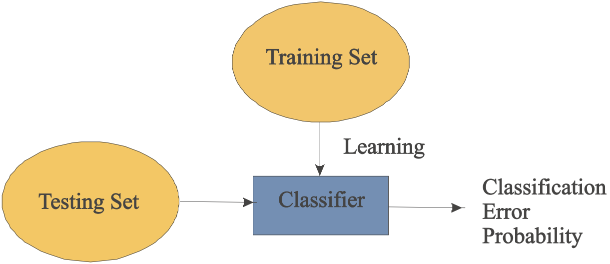
Dimensionality Ratio (n/d)

- PR size illustrates how the expected values of the error estimate evolve with n patterns per class (n/d)
- Both curves have **assymptotic behavior**



Classifier Evaluation

- Determination of reliable estimates of the **classifier error rate** is an essential task to assess its usefulness and compare it with alternative solutions

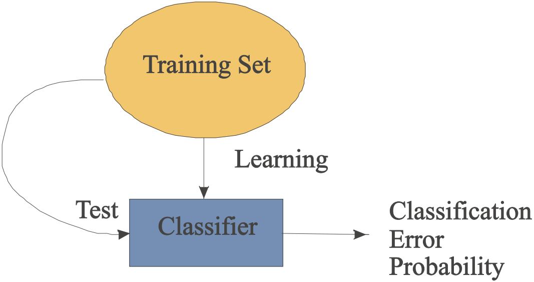


Classifier Evaluation

- Resubstitution Method
- Hold out Method
- Partition Methods
- Bootstrap Method

Resubstitution Method

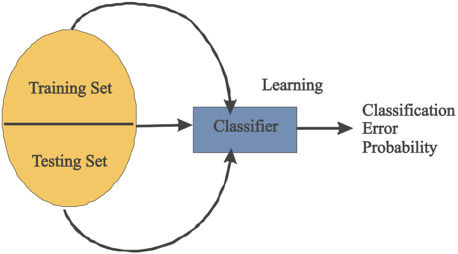
- The whole set X is used for training and testing



- Corresponds to setting the error estimate on the test set (lower curve)

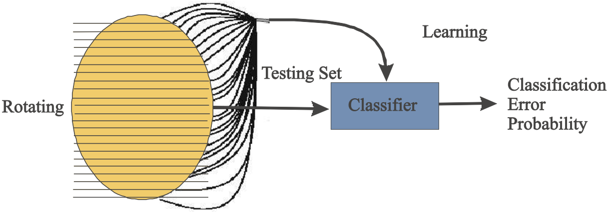
Hold out Method

- The available n samples of X are randomly divided into two disjoint sets X_d and X_t
  - 50% for training
  - 50% for testing
- Or more common
  - 70% for training
  - 30% for testing



- The error estimate is obtained from the test set

Partition Methods



$$Pe_t = \sum_{i=1} k Pe_{ti} k$$

- $k = 2 \longrightarrow$ leave-one-out

Partition Methods

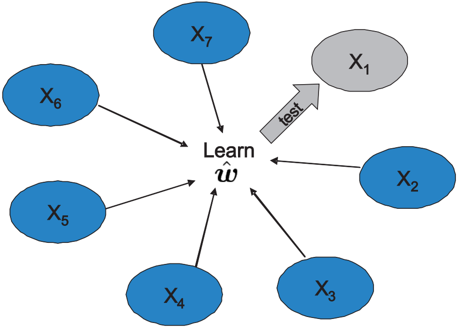
- Divide X into $k > l$ subsets of **randomly chosen** patterns, with each subset having n/k patterns
- Design the classifier using the patterns of $k - 1$ subsets and test it on the remaining one.
- A test set estimate Pe_{ti} is obtained
- Repeat the previous step rotating the position of the test set, obtaining thereby k estimates Pe_{ti}
- Compute the average test set estimate

$$Pe_t = \sum_{i=1} k Pe_{ti} k$$

- and the variance of Pe_t

K-fold Cross Validation

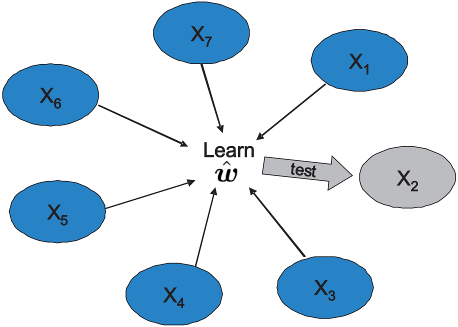
- A technique for estimating test error
- Uses all of the data to validate
- Divide data into K groups $\{X_1, X_2, \cdots, X_K\}$.
- Use each group as a validation set, then average all validation errors



Pe_{t1}

K-fold Cross Validation

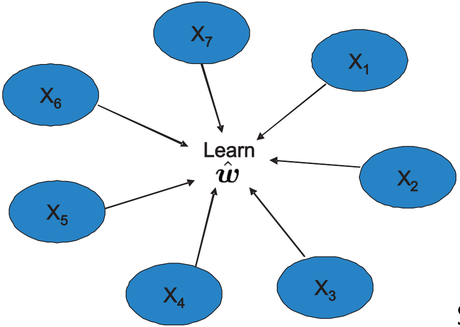
- A technique for estimating test error
- Uses all of the data to validate
- Divide data into K groups $\{X_1, X_2, \cdots, X_K\}$.
- Use each group as a validation set, then average all validation errors



Pe_{t2}

K-fold Cross Validation

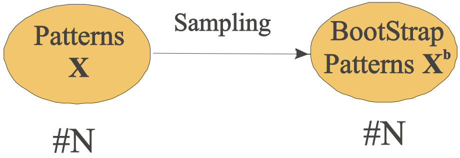
- A technique for estimating test error
- Uses all of the data to validate
- Divide data into K groups $\{X_1, X_2, \cdots, X_K\}$.
- Use each group as a validation set, then average all validation errors



S $\quad Pe_t = \sum_{i=1} k Pe_{ti} k$

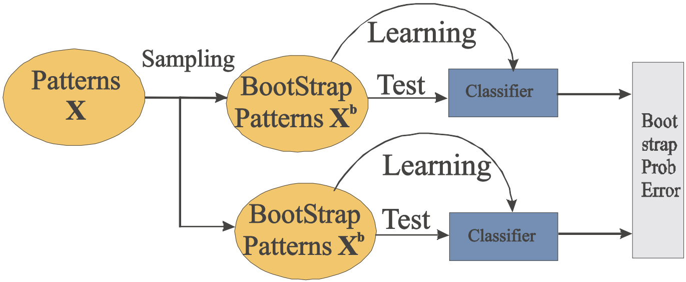
Bootstrap Method

- Based on the generation of artificial data samples by randomly drawing existing samples within uniform distribution within each class



Bootstrap Method

- The error estimate is computed on the original set with the classifier designed using large sets of the bootstrap samples



Summary

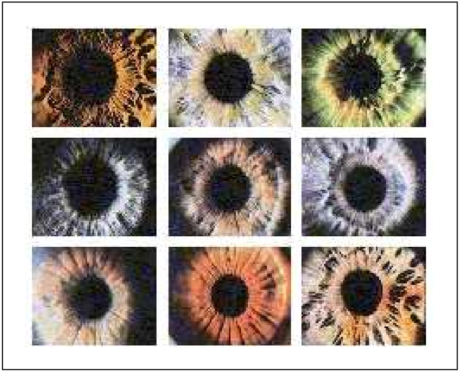
- In practice training set and test set are finite
  - Sets should be independent (at least different) but represent same distribution
  - If training set large and test set small reliable classifier, but error estimate unreliable
  - If training set small and test set large unreliable classifier

Pattern Recognition Techniques

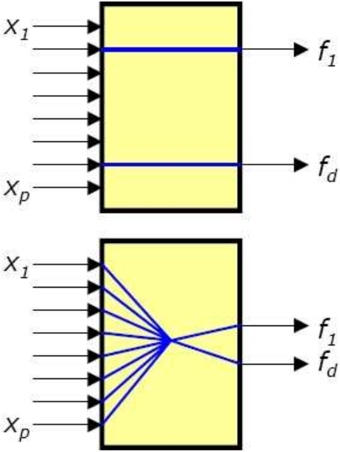
Chapter 7: TRP Feature Selection

Chapter 7: Feature Selection

TRP: 2009-2010



Feature Reduction



- **Feature Selection**
  - Select d out of p measurements
  - Positive Issues: Easy interpretation
  - Negative Issues: Expensive, Approximate
- **Feature Extraction**
  - Map p measurements to d
  - Positive Issues: Cheap, Non-linear
  - Negative Issues: Need all measurements, criterion sub-optimal

Feature Selection

- Reduce number of features - problem of **high dimensionality ratio**
- Modeling an unknown function of a number of variables (features) based on data
- Relative significance of variables is unknown; variables may be
  - Important variables
  - Secondary variables
  - Dependent variables
  - Useless variables

Feature Selection

- Which features are **truly important?**
- Difficult to decide due to:
  - Limited amount of data
  - Lack of algorithm
  - Exhaustive analysis requires 2n experiments
  - Need an empirical method.

Feature Selection

- Reducing the feature space by throwing out some of the features (covariates)
- Also called variable selection
- Motivating idea: try to find a simple, **"parsimonious"** model
- Occam's Razor principle : simplest explanation that accounts for the data is best

Feature Selection

- Why to do it?
  1. Case 1:
     - We are interested in features: we want to know which are **relevant**. If we fit a model, it should be **interpretable**.
  1. Case 2:
     - We are interested in prediction; features are not interesting in themselves, we just want to build a **good classifier** (or other kind of predictor).

Feature Selection

- Why to do it? Case 1: We want to know relevant features.

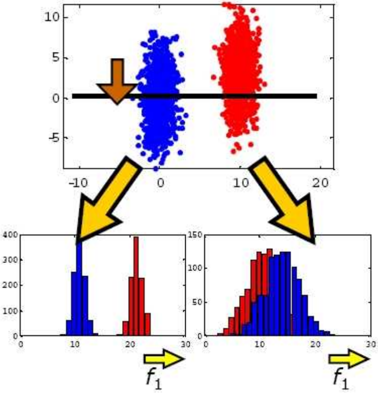
1. What causes lung cancer?
   - Features are aspects of a patient's medical history
   - Binary response variable: did the patient develop lung cancer?
   - Which features best predict whether lung cancer will develop?
2. What causes a program to crash?
   - Features are aspects of a single program execution
   - Which branches were taken?
   - What values did functions return?
   - Binary response variable: did the program crash?
   - Features that predict crashes well are probably bugs.
3. What stabilizes protein structure?
   - Features are structural aspects of a protein
   - Real - valued response variable - protein energy
   - Features that give rise to low energy are stabilizing.

Feature Selection

- Why to do it? Case 2: We want to build a good predictor.

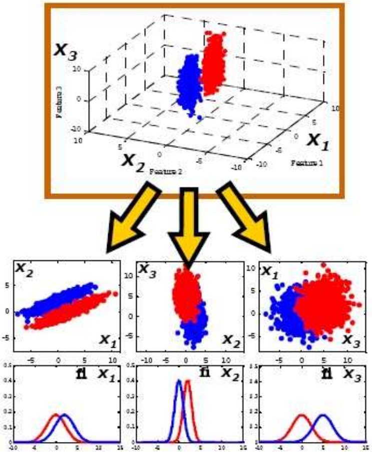
1. Text classification
   - Features for all 105 English words, and maybe all word pairs
   - Common practice: throw in every feature you can think of, let feature selection get rid of useless ones
   - Training too expensive with all features
   - The presence of irrelevant features hurts generalization.
2. Classification of leukemia tumors from microarray gene expression data 72 patients (data points)
   - 7130 features (expression levels of different genes)
   - Disease diagnosis
   - Features are outcomes of expensive medical tests
   - Which tests should we perform on patient?
   - Embedded systems with limited resources
   - Classifier must be compact
3. Voice recognition on a cell phone
   - Branch prediction in a CPU (4K code limit)

Feature Selection



- Two aspects:
  - Selection criterion
  - Search algorithm
- Selection Criteria
  - Individual
  - Probabilities
  - Scatter matrices
  - Warpper
- Search Algorithms
  - Branch-and-bound
  - Sub-optimal

Search Algorithms

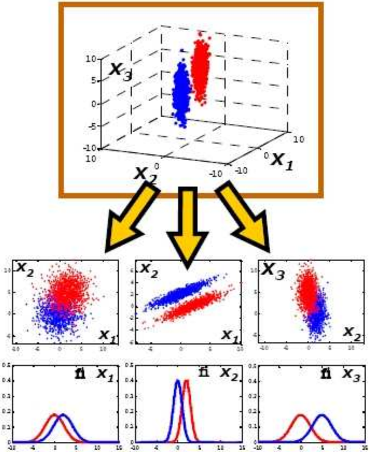


- Select d out of p
  - Select d out of p measurements which optimize criterion J
  - Needs to evaluate all subsets!
- Exhaustive
  - Evaluates all subsets

$$\left(\begin{array}{c} p \\ d \end{array} \right) = \frac{d!}{(p-d)!p!}$$

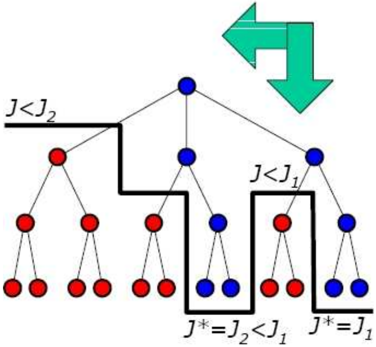
$p = 50, d = 10 :\approx 10^{10}$ subsets
  - Need Search Algorithms

Search Algorithms



- Optimal
  - Branch-and-bound
- Sub-Optimal
  - Forward
  - Backward
  - Plus-l-takeaway-r
- Stopping Criterion

Branch and Bound Search

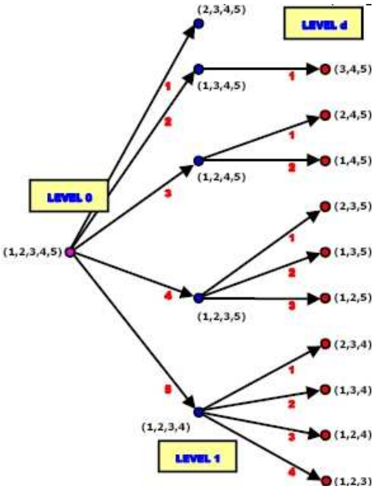


- Optimal search but not exhaustive
  - Exploits monotonicity property of selection criterion j

$$X \subset Y \Rightarrow J(X) < J(Y)$$

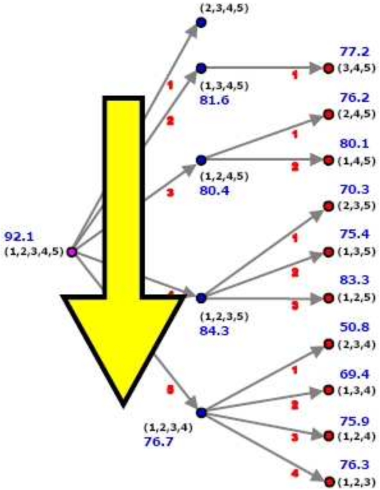
- Principle
  - Constructing tree representing tree subjects
  - Depth-first search
  - Backtrack using J^* as bounds

Branch and Bound Search



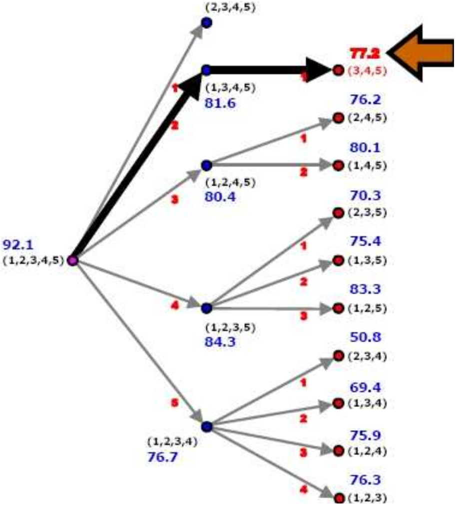
- Find best 3 out of 5
- Construct tree with depth d
  - Level 0: All features
  - Level 1: Subsets of total set one feature removed
  - Level k: Subsets of level $k-1$, one feature removed
- Not Symmetrical
  - Removing $\{4,5\}$ and $\{5,4\}$ from $\{1,2,3,4,5\}$ gives same subset $\{1,2,3\}$
  - Avoid unnecessary calculations: Remove features in decreasing order

Branch and Bound Search



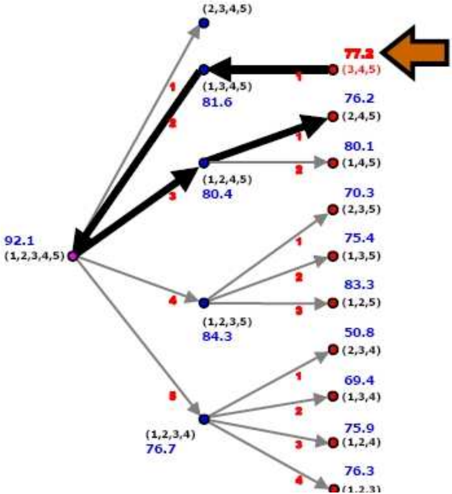
- Criterion
  - For each node criterion, J can be calculated (blue)
- Search
  - Fom least dense part to part with most branches (top-to-bottom)
- Only calculate criterion of those subset tested!

Branch and Bound Search



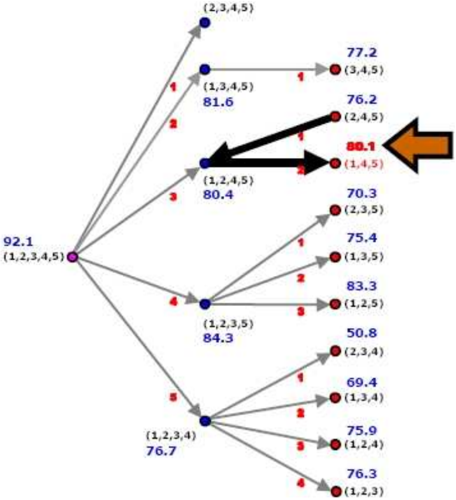
- Top most set
  - $J^* = 77.2$

Branch and Bound Search



- Top most set
  - $J^* = 77.2$
- Backtrack
  - Evaluate $\{2, 4, 5\}$
    - Lower value: discard

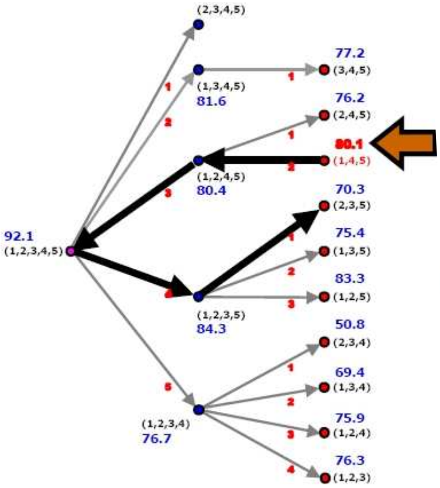
Branch and Bound Search



- Top most set
  - $J^* = 77.2$
- Backtrack
  - Evaluate $\{2, 4, 5\}$
    - Lower value: discard

    Evaluate $\{1, 4, 5\}$
    - Higher value
    - $J^* = 80.1$

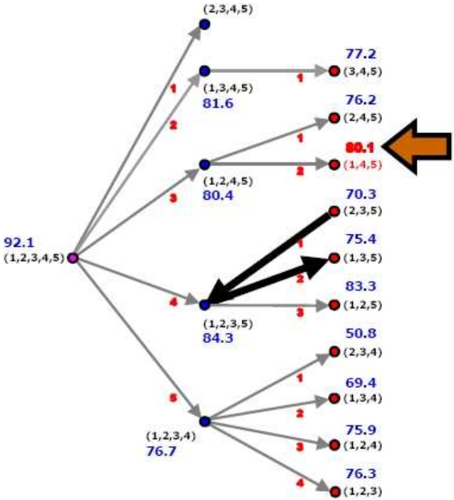
Branch and Bound Search



- Top most set
  - $J^* = 77.2$
- Backtrack
  - Evaluate $\{2, 4, 5\}$
    - Lower value: discard

    Evaluate $\{1, 4, 5\}$
    - Higher value
    - $J^* = 80.1$
- Backtrack
  - Evaluate $\{2, 3, 5\}$
    - Lower value: discard

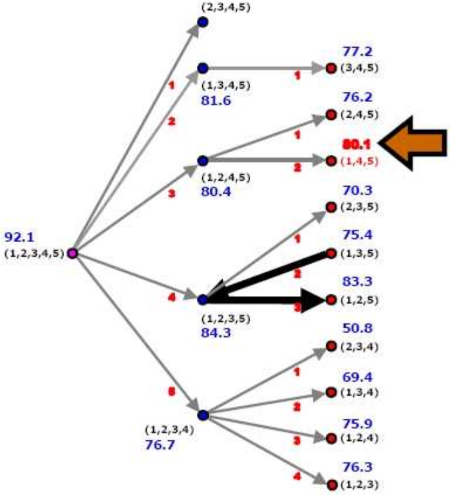
Branch and Bound Search



- Top most set
  - $J^* = 77.2$
- Backtrack
  - Evaluate $\{2, 4, 5\}$
    - Lower value: discard

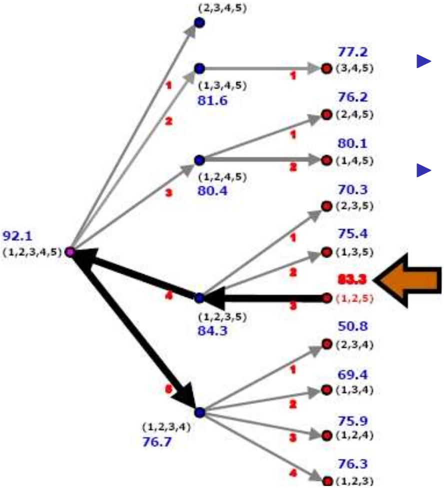
    Evaluate $\{1, 4, 5\}$
    - Higher value
    - $J^* = 80.1$
- Backtrack
  - Evaluate $\{2, 3, 5\}$
    - Lower value: discard
  - Evaluate $\{1, 3, 5\}$
    - Lower value: discard

Branch and Bound Search



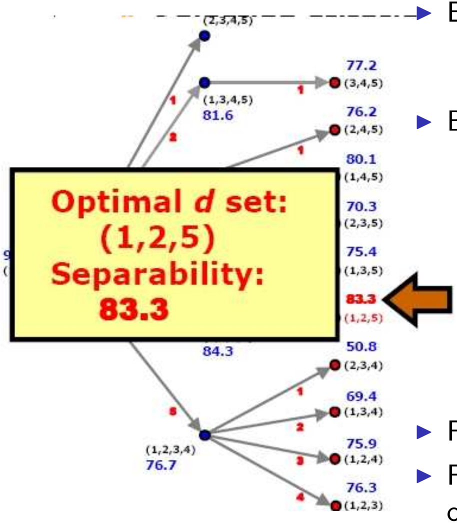
- Top most set
  - $J^* = 77.2$
- Backtrack
  - Evaluate $\{2,4,5\}$
    - Lower value: discard
  - Evaluate $\{1,4,5\}$
    - Higher value
    - $J^* = 80.1$
- Backtrack
  - Evaluate $\{2,3,5\}$
    - Lower value: discard
  - Evaluate $\{1,3,5\}$
    - Lower value: discard
  - Evaluate $\{1,2,5\}$
    - Higher value
    - $J^* = 83.3$

Branch and Bound Search



- Evaluate $\{1, 2, 5\}$
  - Higher value
  - $J^* = 83.3$
- Backtrack
  - Evaluate $\{1, 2, 3, 4\}$
    - Lower value: discard
  - Now all sets below can also be discarded since $X \subset Y \Rightarrow J(X) < J(Y)$ Thus, no subset of $\{1, 2, 3, 4\}$ can give a higher score than J^*

Branch and Bound Search



- Evaluate $\{1,2,5\}$
  - Higher value
  - $J^* = 83.3$
- Backtrack
  - Evaluate $\{1,2,3,4\}$
    - Lower value: discard
  - Now all sets below can also be discarded since $X \subset Y \Rightarrow J(X) < J(Y)$ Thus, no subset of $\{1,2,3,4\}$ can give a higher score than J^*
- Finished
- Found optimalset of d features out of p measurements

Sub-Optimal Methods

- **Genetic Algorithm** Search
- **Sequential** Search (**direct**)
  - **Backward** Search
  - **Forward** Search
- **Sequential** Search (**dynamic**)

Genetic Algorithm Search

- Genetic Algorithm Search
  - Stochastic search in the feature space guided by the idea of inheriting, at each search step, good properties of parent subsets found in previous steps

Sequential Search (direct)

- Backward Search
  - Process starts with the whole feature data set and at, each step, the feature that contributes the least for class discrimination is removed
  - Process goes until the merit criterion for any candidate feature is above a specified threshold
- Forward Search
  - Process starts with feature of most merit and, at each step, all the features not yet included are revised; the one which contributes the most to class discrimination is included
  - Process goes until the merit criterion is below a specified threshold

Sequential Search (dynamic)

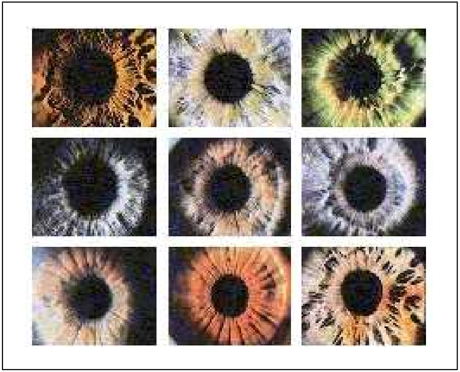
- Plus-l take away r
  - Combination of backward and forward searches at each level
  - Trade-off in terms of computational effort between Branch and Bound method and Direct Search
  - Some implementations of the technique automatically compute l and r

Pattern Recognition Techniques

Chapter 8: TRP Support Vector Machines (SVM)

Chapter 8: Support Vector Machines (SVM)

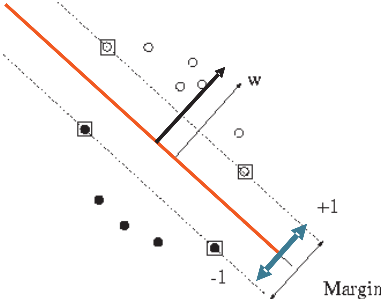
TRP: 2009-2010



Support Vector Machines (SVM)

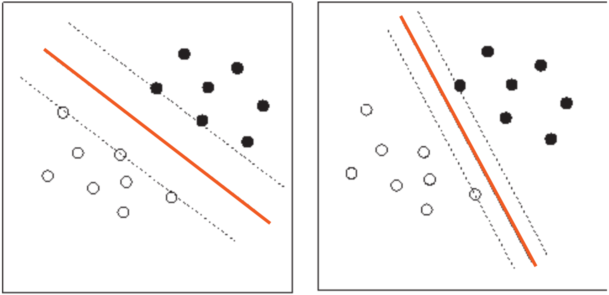
- Fundamentals
  - Based on **Statistical Learning Theory** (Vapnik 1995)
  - Choose the kernel before the learning process
- Recent applications of SVM
  - **Pattern recognition**
    - Isolated handwritten digit recognition
    - Object recognition
    - Speaker identification
    - ...
  - **Regression estimation**
  - **Density estimation**

Idea of SVM in Pattern Classification



- The support vector algorithm simply looks for largest margin.
- $d + (d-)$ is the shortest distance from the separating plane to the closest positive (negative) point.
- $(d+ == d-)$
- Margin equal to $d+$ plus $d-$

Idea of SVM in Pattern Classification



(a) Larger margin (b) Smaller margin

- For these machines, the **support vectors** are the critical elements of the training set.
- If other training points are removed, and the training was repeated, the same separating **hyperplane** would be found.

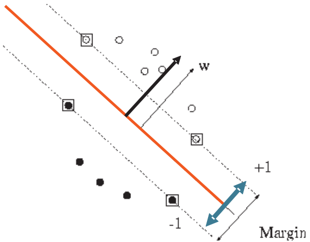
A general two-class pattern classification problem

- sample point $(x_1, y_1), (x_2, y_2) \cdots (x_i, y_i)$
- **x** is the vector of the point
- y is the class label
- For example, in two-class pattern classification

$$y = \{+1, -1\}$$

- Find a classifier with the decision function $f(\mathbf{x})$ such that $y = f(\mathbf{x})$

Linear Support Vector Machine



- **w** is the normal to the hyperplane
- $|w0|/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin

- decision function is $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + w_0 = 0$
- Notice that there is ambiguity in the magnitude of **w** and w_0. They can be arbitrary scaled such that :$H1 = \mathbf{w}'\mathbf{x} + w_0 = 1$, $H2 = \mathbf{w}'\mathbf{x} + w_0 = -1$
- $d+ = d- = 1//||\mathbf{w}||$, so , margin $= d + +d- = 2/||\mathbf{w}||$
- This optimization problem is solved using the Lagrangian formulation

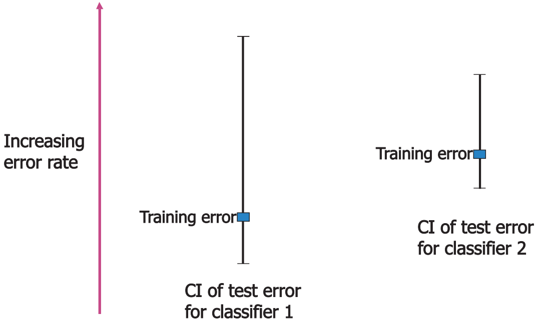
A general two-class pattern classification problem

- Perform the principle of **structural risk minimization**

  Empirical risk + learning function complexity

- As a consequence SVM can provide a good generalisation independent of the distributions of patterns
- The basic idea is the adjustment of a discriminating function that optimally uses the separability information of the boundary patterns

Structural Risk Minimization (SRM)



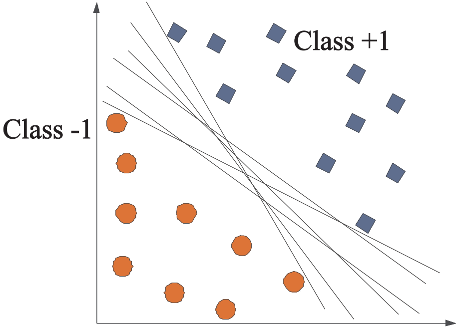
- SRM prefers classifier 2 although it has a higher training error, because the upper limit of CI is smaller

SVM

- Assume a linear discriminating function and two linearly separate classes with target values +1 and -1. A discriminating hyperplane will satisfy:

$$
\begin{aligned}
\mathbf{w}'\mathbf{x} + w_0 &\geq 0 \qquad \text{if } \mathbf{t}_i = +1 \\
\mathbf{w}'\mathbf{x} + w_0 &< 0 \qquad \text{if } \mathbf{t}_i = -1
\end{aligned}
$$

Separating Hyperplane

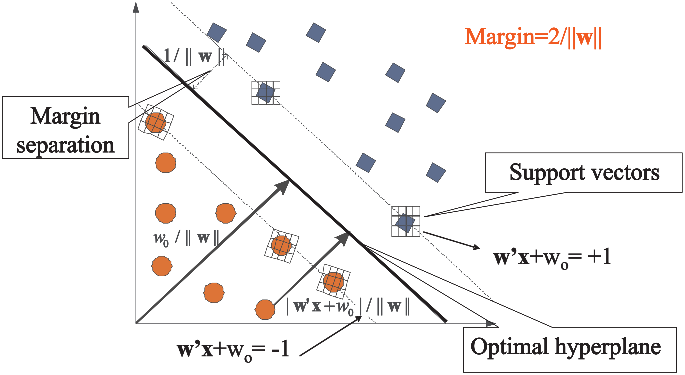


Values of **w** obviously are not unique

↓

Infinity of solutions . . .

Optimal Separating Hyperplane



Canonical Hyperplane

- Minimum distance from a point to the hyperplane is $1/||\mathbf{w}||$
- Canonical hyperplane satisfying the condition

$$\min_k |\mathbf{w}'\mathbf{x} + w_0| = 1$$

$$\mathbf{t}_i(\mathbf{w}'\mathbf{x}_i + w_0|) = 1$$

if and only if $\mathbf{x}_i$ is a support vector

Maximisation of the Margin

- Optimisation Problem:
- Primal Problem

$$\begin{cases} \text{minimize } \Phi(\mathbf{x}) = \frac{1}{2}||\mathbf{w}||^2 \\ \text{subject to } t_i(\mathbf{w}'\mathbf{x}_i + w_0) \geq 1 \\ i = 1 \cdots n \end{cases}$$

Quadratic Programming Problem

- Lagrange Multipliers Method

$$J(\mathbf{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \boldsymbol{\alpha}_i \left(\mathbf{t}_i \left(\mathbf{w}'\mathbf{x}_i + w_0\right) - 1\right)$$

Lagrangian Function

- Differentiating Lagrangian Function with respect to **w** and w_0, the following optimality conditions are derived

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{t}_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i \mathbf{t}_i = 0$$

- In order to compute the weights we need the Lagrange multipliers

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j \mathbf{t}_i \mathbf{t}_j \mathbf{x}_i' \mathbf{x}_j$$

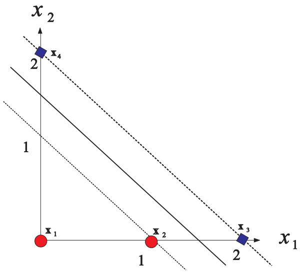
Optimal Linear Discriminating

Optimal weight vector $\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathbf{t}_i \mathbf{x}_i$

Optimal bias $w_0^* = -\frac{1}{2}\mathbf{w}^* \mathbf{x}$

$\mathbf{x}_i$	-	Support Vectors (SV)
$\mathbf{x}_p$	-	SV (positive class)
$\mathbf{x}_n$	-	SV (negative class)

Example: Lagrange Multipliers



- class $+1$
  $\mathbf{x}_1 = [0\ 0]'; \mathbf{x}_2 = [0\ 0]'$;
- class -1
  $\mathbf{x}_3 = [2\ 0]'; \mathbf{x}_4 = [0\ 2]'$;
- Let us solve the dual problem $Q(\boldsymbol{\alpha})$:

$$Q(\boldsymbol{\alpha}) = (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2}\left(\alpha_2^2 - 4\alpha_2\alpha_3 + 4\alpha_3^2 + 4\alpha_4^2\right)$$

and differentiate with respect to $\boldsymbol{\alpha}$:

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0 \\ \alpha_2 - 2\alpha_3 = 1 \\ -2\alpha_2 + 4\alpha_3 = 1 \\ 4\alpha_4 = 1 \end{cases}$$

Example: Lagrange Multipliers

- It has the following solution of nonnegative $\alpha' s$:

$$\alpha_1 = 0; \alpha_2 = 1; \alpha_3 = 3/4; \alpha_4 = 1/4.$$

- Optimal weight $\mathbf{w}^*$

$$\mathbf{w}^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{3}{4} \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -1/2 \end{bmatrix}$$

- Hence the linear discriminant is a straight line at $-45°$ and the **support vectors** are the points x_2, x_3 and x_4 (the vectors with non zero **Lagrange multipliers**)
- Bias

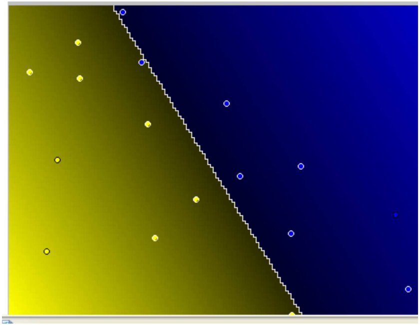
$$w_0^* = -\frac{1}{2} [-1/2 \quad -1/2] \begin{bmatrix} 3 \\ 0 \end{bmatrix} = 3/4$$

- Discriminating Hyperplane

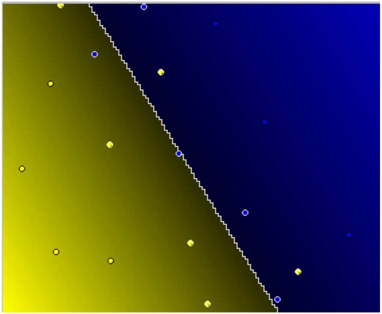
$$d(\mathbf{x}) = 3 - 2x_1 - 2x_2 = 0$$

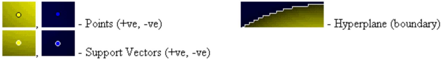
Linear SVM

Linear separable classes



Non-linear separable classes

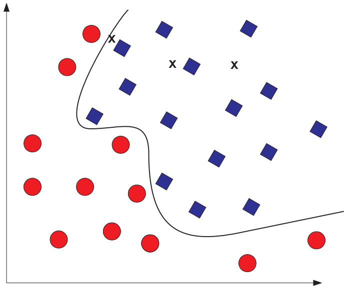




[1]

[1] Demo: http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml

Non-Separable Classes

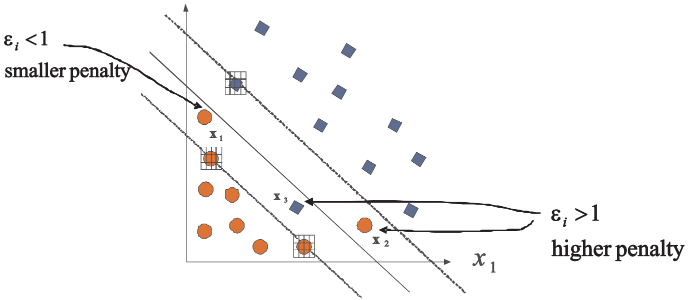


- Conditions for determination of the separating hyperplane are now reformulated to accomodate non-separable classes

$$\begin{cases} \text{minimize } \Phi(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \epsilon_i \\ \text{subjected to } \mathbf{t}_i\left(\mathbf{w}'\mathbf{x}_i + w_0\right) \geq 1 - \epsilon_i, i = 1 \cdots n. \end{cases}$$

- Slack variables ϵ_i penalising the deviation of a data point from the ideal separable situation

Optimal Non-Linear Separating Hyperplane (1)



- **Support vectors** must satisfy the condition:

$$\mathbf{t}_i \left(\mathbf{w}'\mathbf{x}_i + w_0 \right) = 1 - \epsilon_i \qquad (1)$$

- and

$$C \sum_{i=1}^{n} \epsilon_i = C\xi$$

Optimal Non-Linear Separating Hyperplane (2)

- **Trial and error** choice of parameter C which has to be chosen experimentally
- Solution of **quadratic programming problem** is obtained the same way as before
- In the formulation of the dual form the Lagrange multipliers must satisfy to:

$$0 \leq \boldsymbol{\alpha}_i \leq C$$

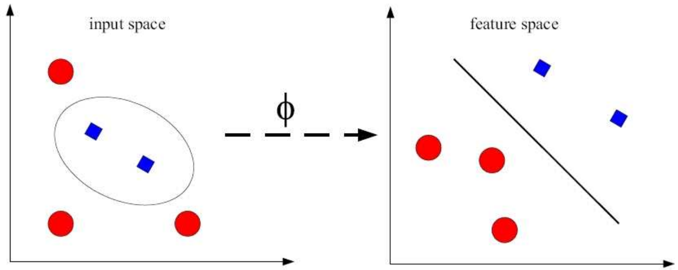
- Also we have for the **weight vector** a summation over the SVs

$$\mathbf{w}^* = \sum_{SVs} \boldsymbol{\alpha}_i^* \mathbf{t}_i \mathbf{w} \qquad (2)$$

Non-Linear Decision Functions

- Perform a a non-linear mapping into a higher dimension feature space

$$\begin{aligned} d(\mathbf{x}) &= w_1 f_1(\mathbf{x}) + w_1 f_1(\mathbf{x}) + \cdots + w_k f_k(\mathbf{w}) = \mathbf{w}'\mathbf{y} \\ \text{with } \mathbf{y} &= [f_1(\mathbf{x}) \;\; f_2(\mathbf{x}) \; \cdots \; f_k(\mathbf{x}) \;] \end{aligned}$$



Non-Linear Decision Functions

- Perform a non-linear mapping into a higher dimension feature space

$$\begin{aligned} d(\mathbf{x}) &= w_1 f_1(\mathbf{x}) + w_1 f_1(\mathbf{x}) + \cdots + w_k f_k(\mathbf{w}) = \mathbf{w}'\mathbf{y} \\ \text{with } \mathbf{y} &= [f_1(\mathbf{x}) \;\; f_2(\mathbf{x}) \; \cdots \; f_k(\mathbf{x}) \;] \end{aligned}$$

- **Optimal weight vector**:

$$\begin{aligned} \mathbf{w}^* &= \sum_{SVs} \alpha_i \mathbf{t}_i f(\mathbf{x}_i) \\ d(\mathbf{x}) = \mathbf{w}'\mathbf{y} &= \sum_{SVs} \alpha_i^* \mathbf{t}_i f(\mathbf{x}_i)' f(\mathbf{x}) \\ d(\mathbf{x}) = \mathbf{w}'\mathbf{y} &= \sum_{SVs} \alpha_i^* \mathbf{t}_i K(\mathbf{x}_i, \mathbf{x}) \end{aligned}$$

Kernel Trick

$$K(\mathbf{x}_i, \mathbf{x}) = f(\mathbf{x}_i) f(\mathbf{x})$$

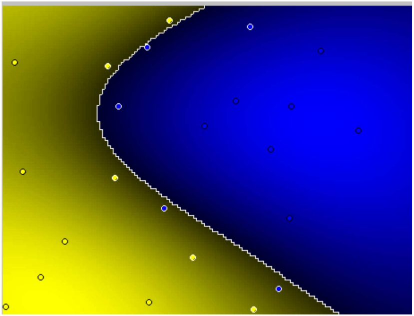
Kernel Function

$K(\mathbf{x}, \mathbf{x}_i) -$ **inner product kernel function**

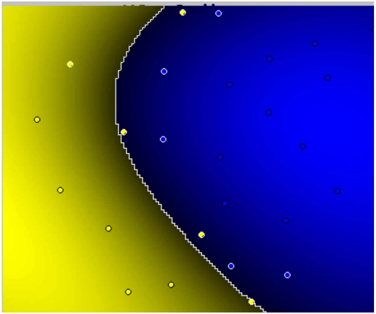
Linear	$K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}'\mathbf{x}_i$
Polynomial	$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}'\mathbf{x}_i + 1)^p$
Gaussian Radial Basis Function	$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-(\mathbf{x} - \mathbf{x}_i)^2 / 2\sigma^2\right)$
Exponential Radial Basis Function	$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\|\mathbf{x} - \mathbf{x}_i\|^2 / 2\sigma^2\right)$
Tangent Hyperbolic Sigmoid	$K(\mathbf{x}, \mathbf{x}_i) == \tanh\left(a\mathbf{x}'\mathbf{x}_i + b\right)$

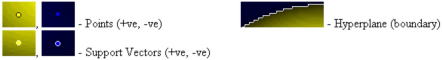
Non-Linear SVM

Non-linear separable classes



Non-linear separable classes

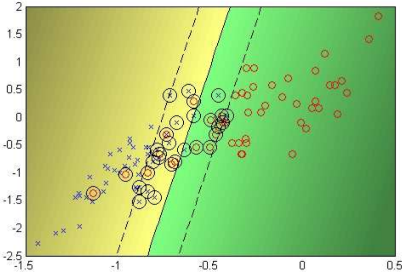




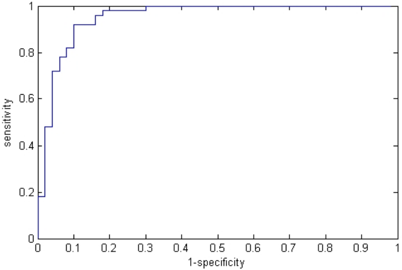
[2]

[2] Demo: `http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml`

Example 1:SVM Cork Stoppers

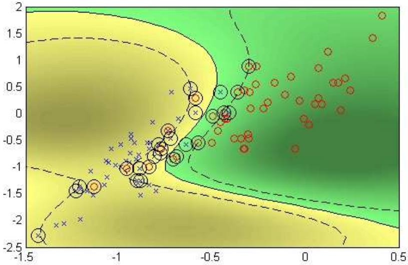


$C = 10, N_{SV} = 28$
Kernel RBF, arg $= 0.8$

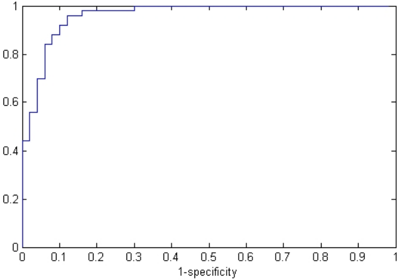


$Pe = 0.08(8\%)$

Example 2:SVM Cork Stoppers

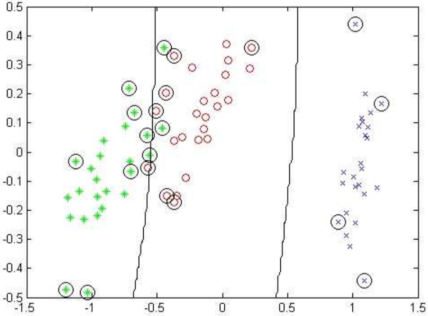


$C = 10, N_{SV} = 31$
Linear Kernel



$Pe = 0.08(8\%)$

Example 3:Multi-Class SVM IRIS Data Set



$$C = 10, N_{SV} = 31 \ \ Pe_d = 0.04(4\%); Pe_t = 0.0267(2.67\%)$$

Kernel: rbf, arg= 0.5

Pattern Recognition Techniques

Bibliography

Bibliography (1)

1. Marques de Sá, J.P. (2001). *Pattern Recognition Concepts, Methods and Applications* XIX, 318 p., 197 illus., ISBN: 3-540-42297-8
2. Duda, R. O., Hart, P.E., and Stork, D.G. (2001). *Pattern Classification*, 2nd ed. Wiley Interscience, ISBN: 0-471-05669-3.
3. David G. Stork and Elad Yom (2001) *Computer Manual in MATLAB to accompany Pattern Classification* , Wiley Interscience, 136 pages, ISBN: 0-471-42977-5
4. Kevin L. Priddy and Paul E. Keller (2005), *Artificial Neural Networks: An Introduction*, SPIE - The International Society for Optical Engineering

Bibliography (2)

5 X. Wu and V. Kumar, (2009),*The Top Ten Algorithms in Data Mining*, Chapman & Hall/CRC, Data Mining and Knowledge Discovery Series, ISBN: 978-1-4200-8964-6,2009 http://www.crcpress.com/product/isbn/9781420089646

6 Ian H. Witten, Eibe Frank (2005). *Data Mining For WEKA*, 2nd ed., Amsterdam : Morgan Kaufmann, cop. 2005, ISBN: 0-12-088407-0

7 Bishop, C.M. (2006) *Pattern Recognition and Machine Learning*, Springer Verlag

8 Jorge Salvador Marques (2005),*Reconhecimento de Padrões: Métodos Estatísticos e Neuronais*, ISBN: 972-8469-08-X Ano: 2005 2ª Edição http://istpress.ist.utl.pt/lrecpad.html