Αναγνώριση Προτύπων: Ενοποιημένη παρουσίαση και σύγχρονες τάσεις

Σέργιος Θεοδωρίδης
Καθηγητής Τμήματος Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστημίου Αθηνών

- Στην ομιλία αυτή γίνεται μία ενοποιημένη παρουσίαση διαφόρων τεχνικών και μεθόδων Αναγνώρισης Προτύπων τόσο για την περίπτωση εκπαίδευσης με επίβλεψη (supervised) όσο και για την περίπτωση εκπαίδεσυης χωρίς επίβλεψη (unsupervised). Η ομιλία εστιάζει σε όλες τις φάσεις σχεδιασμού ενός συστήματος αναγνώρισης προτύπων, όπως γένεση χαρακτηριστικών, επιλογή χαρακτηριστικών, σχεδιασμό ταξινομητών και τέλος στην αξιολόγηση του συστήματος.
- Η ομιλία εστιάζει στις πλέον σύγχρονες τεχνικές σχεδιασμού ταξινομητών, όπως Support Vector Machines και τεχνικές γένεσης χαρακτηριστικών βασισμένων σε Independent Component Analysis. Επίσης, παρουσιάζονται και πιο κλασσικές μεθοδολογίες που χρησιμοποιούνται σήμερα για ανάκτηση δεδομένων με βάση το περιεχόμενο (content based retrieval), όπως Δυναμική Στρέβλωση και Κρυφά Μαρκοβιανά μοντέλα (Hidden Markov Models) .

PATTERN RECOGNITION
REVIEW AND RECENT TRENDS

SERGIOS THEODORIDIS

Dept. of Informatics and

Telecommunications

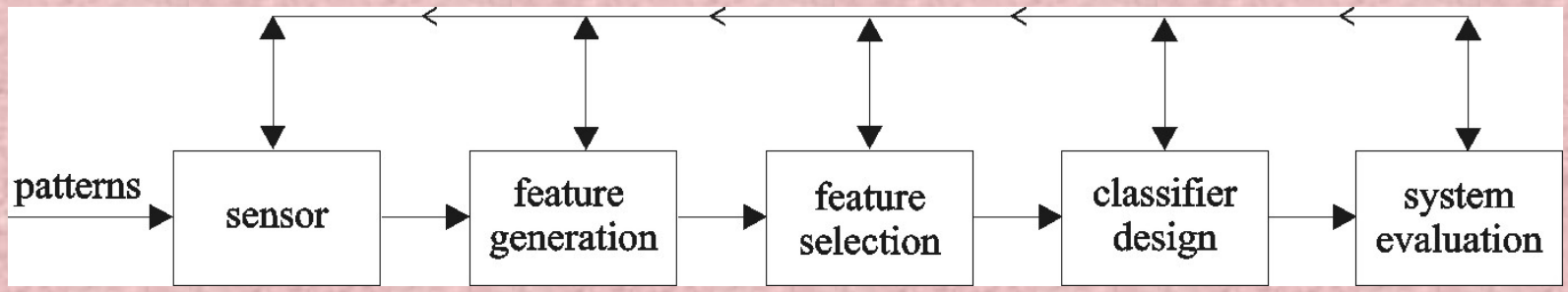
Univ. of Athens

- What is Pattern Recognition?
  - Pattern Recognition is the Scientific Discipline whose goal is the classification of objects into a number of classes or categories.
  - Depending on the application these objects can be
    - Images
    - Speech waveforms
    - Any measurement set need to be classified
  - The objects to be classified are known as Patterns.

- Typical Applications
  - Machine Vision
  - OCR
  - Computer Aided Diagnosis
  - Speech Recognition
  - Data Base Retrieval (PRBMS)
  - Remote Sensing
  - Bioinformatics
  - Data Mining

- Supervised versus Unsupervised Pattern Recognition
  - **Supervised PR** : The number of classes are known to the PR-system designer
    - A set of training data is available and the class label of each one of the data points is known
  - **Unsupervised PR or Clustering** : There is no training data set with known class labels
    - Data have to be clustered (grouped) together by unraveling similarities among them

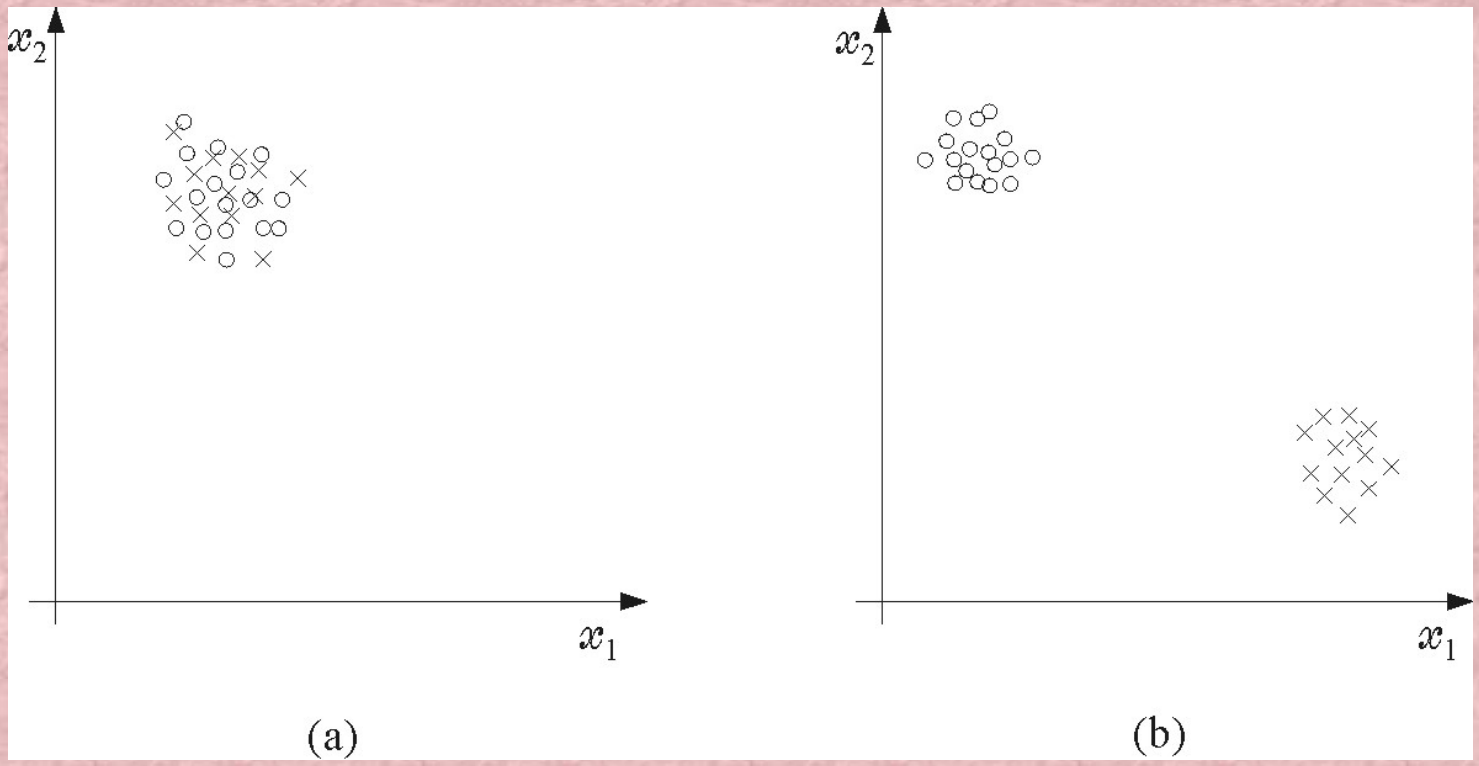
- A Pattern Recognition system



- Feature Generation Stage
  - **Features** : Measurable quantities, each carrying information that can be used for the classification task.
  - A large number of features, xi, i=1, 2, … L, can be generated during this stage.
    - Some of them may convey no discriminative information and have to discarded
    - Some may convey discriminative information, but to a varying degree

Feature Selection Stage

- During this stage, from the large number of features generated in the previous stage, select the "best", i.e., the most informative ones, from the class - discrimination point of view.

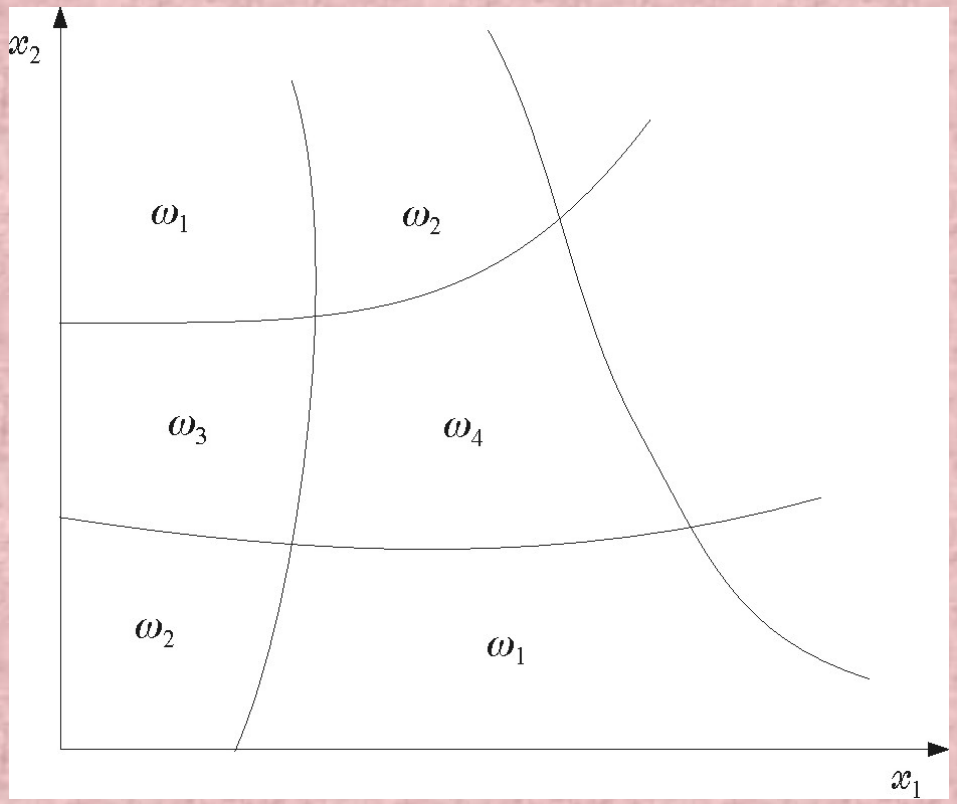


- Select the "best" number l.
  This is a very crucial stage. A bad choice of the l and / or the best features, results in bad performance, irrespective of the classifier design stage

- Classifier Design Stage
  - Form Feature vectors from the l selected features

$$\underline{x} = [x_1, x_2, \ldots, x_l]^T \in R^l$$

> Partition the feature space into regions. Each region is associated with a single class.



> The (hyper)Surfaces that partition the space are known as Decision Surfaces

- System Validation
  - Estimate the error probability of the designed PR system

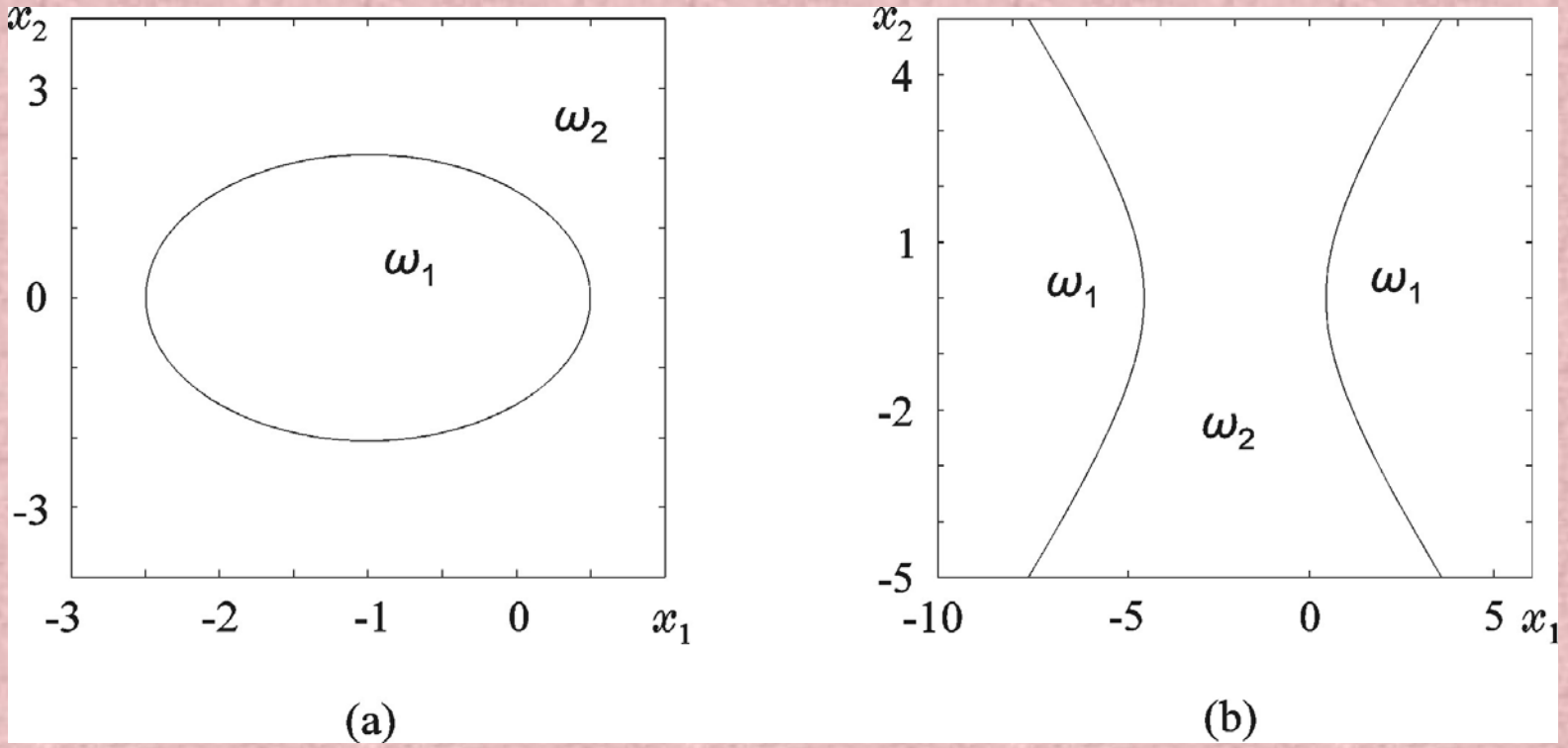
- Classifier Design
  - Bayesian Classifier
    - For an M-class task assign

$$\underline{x} \rightarrow \omega_i :$$

$$\arg\max_i P(\omega_i \mid \underline{x}) = \arg\max_i P(\omega_i) P(x \mid \omega_i)$$

  - The Bayesian classifier is optimum, i.e. Minimizes the Error Probability

> If All $p(\underline{x}|\omega_i)$ are normal, then the Decision Hypersurfaces are quadrics



- k-Nearest Neighbor Classifier
  - **The steps** :
    - 1) Choose k
    - 2) From the training set xi, i=1, 2, … N find the k nearest neighbors to x. Assign x to the class with the highest number ki among k
  - As $N \to \infty$

$$P_B \le P_{KNN} \le P_B + \frac{1}{\sqrt{kl}}$$

    - For the simplest k=1

$$P_B \le P_{NN} \le 2P_B$$

- Linear Classifiers
  - **The goal** : Partition the feature space via hyperplanes. That is, for each ω_i define a linear discriminant function fi($\underline{x}$)
  - Assign $\underline{x}$ to ω_i

$$\omega_i : \arg\max_i f_i(\underline{x})$$

  - Decision surfaces are of the form

$$\underline{w}^T \underline{x} + w_0 = 0$$

- The Perceptron
  - Assume classes linearly separable
  - Cost function

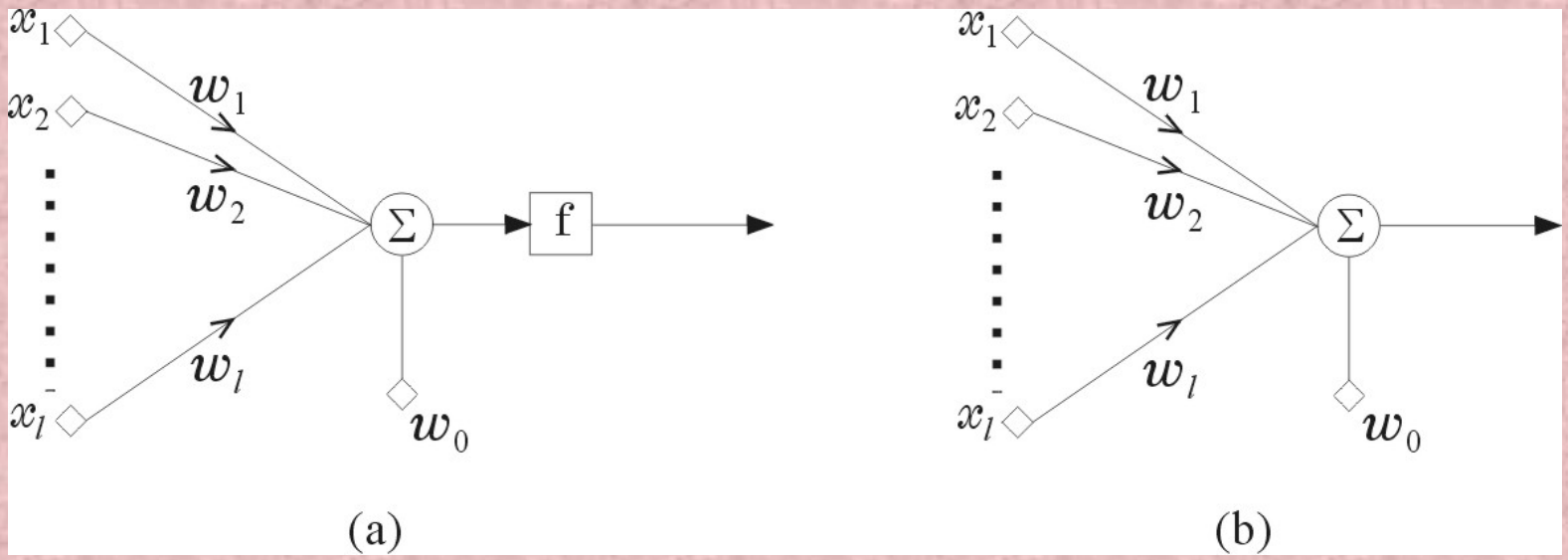
$$J(\underline{w}) = \sum_{\underline{x} \in Y} \delta_x \underline{w}^T \underline{x}$$

  - The algorithm

$$\underline{w}(t+1) = \underline{w}(t) - \rho_t \sum_{\underline{x} \in Y} \delta_x \underline{x}$$

(Y set of misclassified training pattern by $\underline{w}$)

  - Converges in finite number of steps


x_1
x_2
x_l
w_1
w_2
w_l
w_0
Σ
f
(a)
x_1
x_2
x_l
w_1
w_2
w_l
w_0
Σ
(b)

- Support Vector Machines
  - **The goal** : Design the linear classifier that leaves the maximum margin from both classes (separable case)

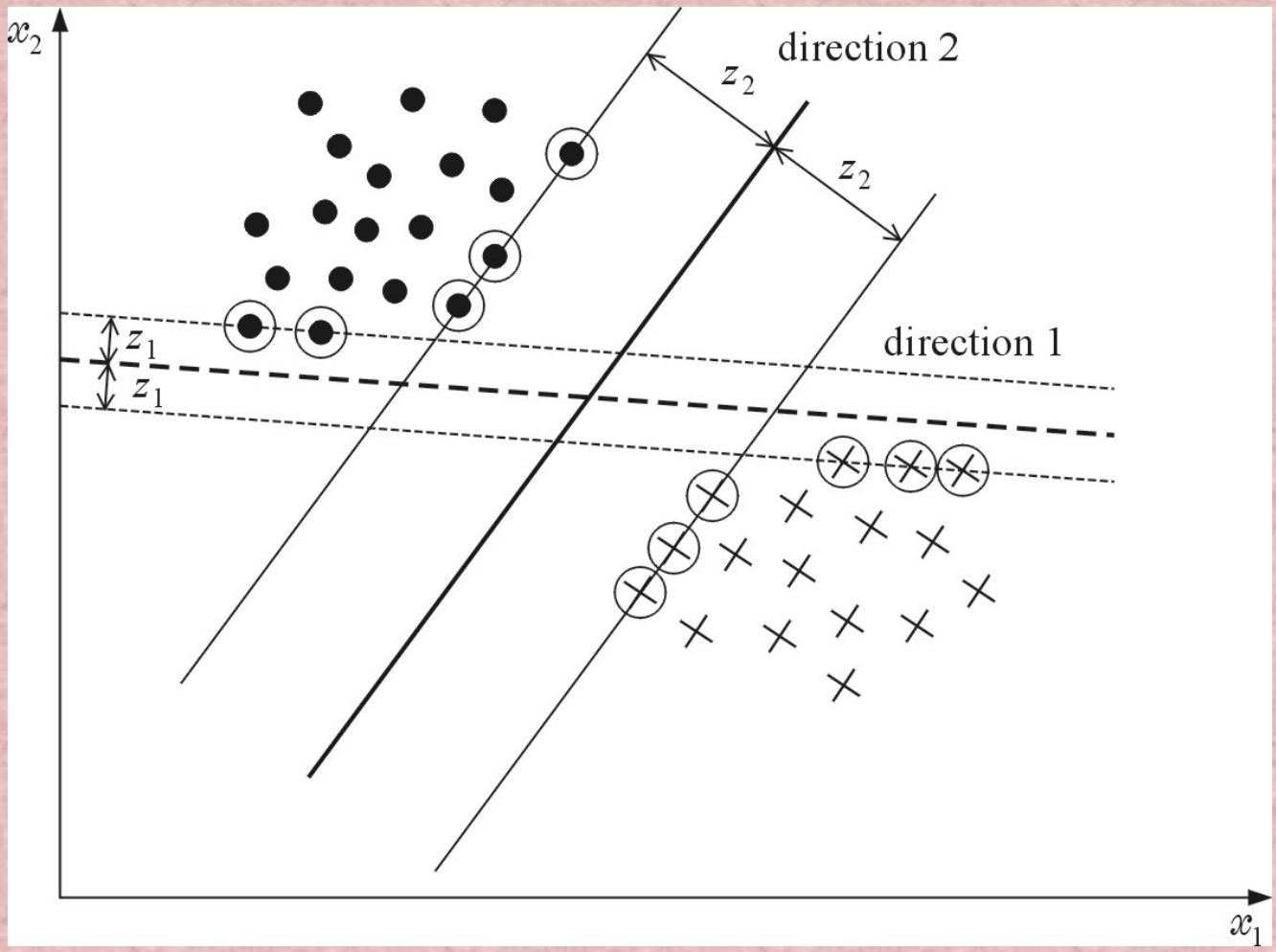
minimize

$$J(w) = \frac{1}{2}\|w\|^2$$

subject to :

$$y_i(w^T \underline{x}_i + w_0) \geq 1, i = 1,2,\ldots,N$$

$$y_i = \begin{cases} 1 & \underline{x} \in \omega_1 \\ -1 & \underline{x} \in \omega_2 \end{cases}$$


x_2
direction 2
z_2
z_2
direction 1
z_1
z_1
x_1

> **The solution** :

$$g(\underline{x}) \equiv \underline{w}^T \cdot \underline{x} + w_0$$

$$= \sum_{i=1}^{N_s} \lambda_i y_i \underline{x}_i^T \underline{x} + w_0$$

λ_i : Lagrange multipliers

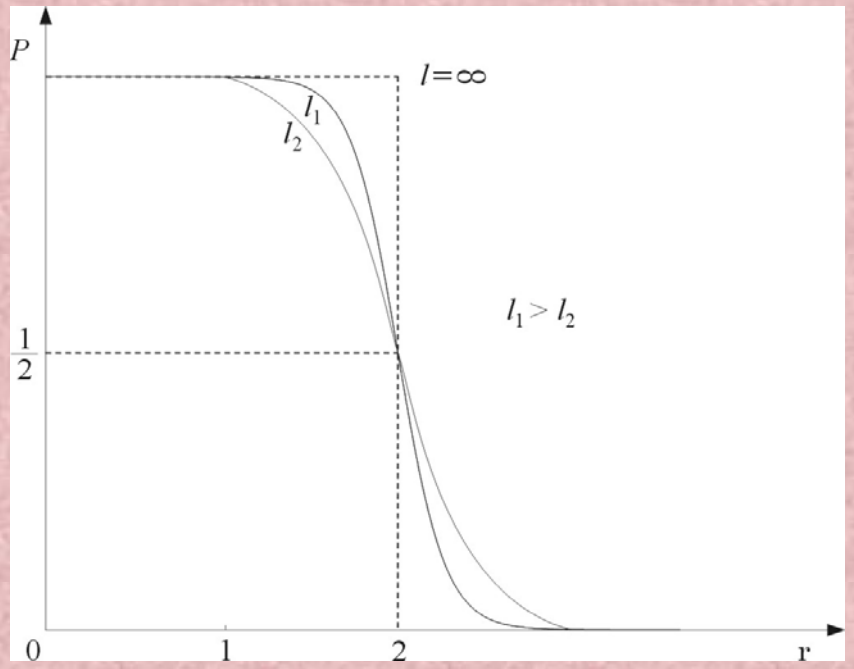
N_s : Number of Support vectors from the training set

> The task is a quadratic convex optimization problem with linear inequality constraints.

> It guarantees good generalization properties

- Nonlinear Classifiers
  - Capacity of the l-dimensional space in linear dichotomies

$$P_N^l = \begin{cases} \frac{1}{2^{N-1}} \sum_{i=0}^{l} \binom{N-1}{1} & N > l+1 \\ 1 & N \le l+1 \end{cases}$$

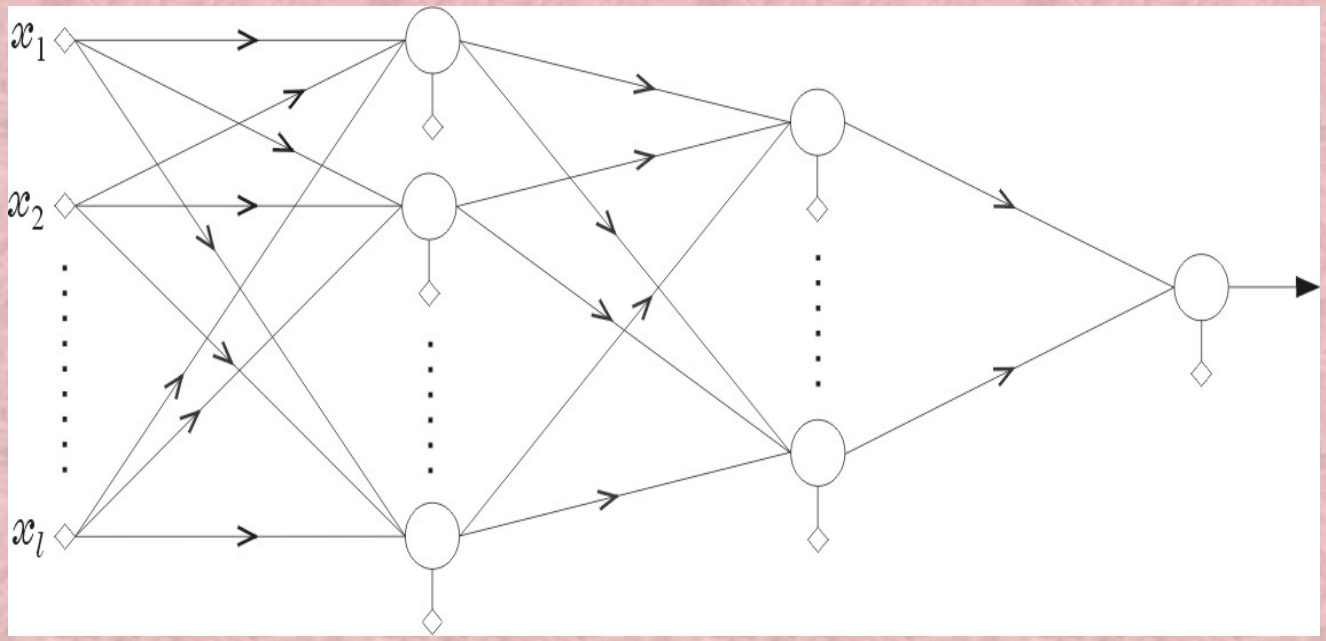


> Given N points, then mapping into a higher dimensional space

$$R^l \rightarrow R^k, k > l$$

increases the probability of locating the N points so that to be linearly separable.

The Multilayer Perceptron



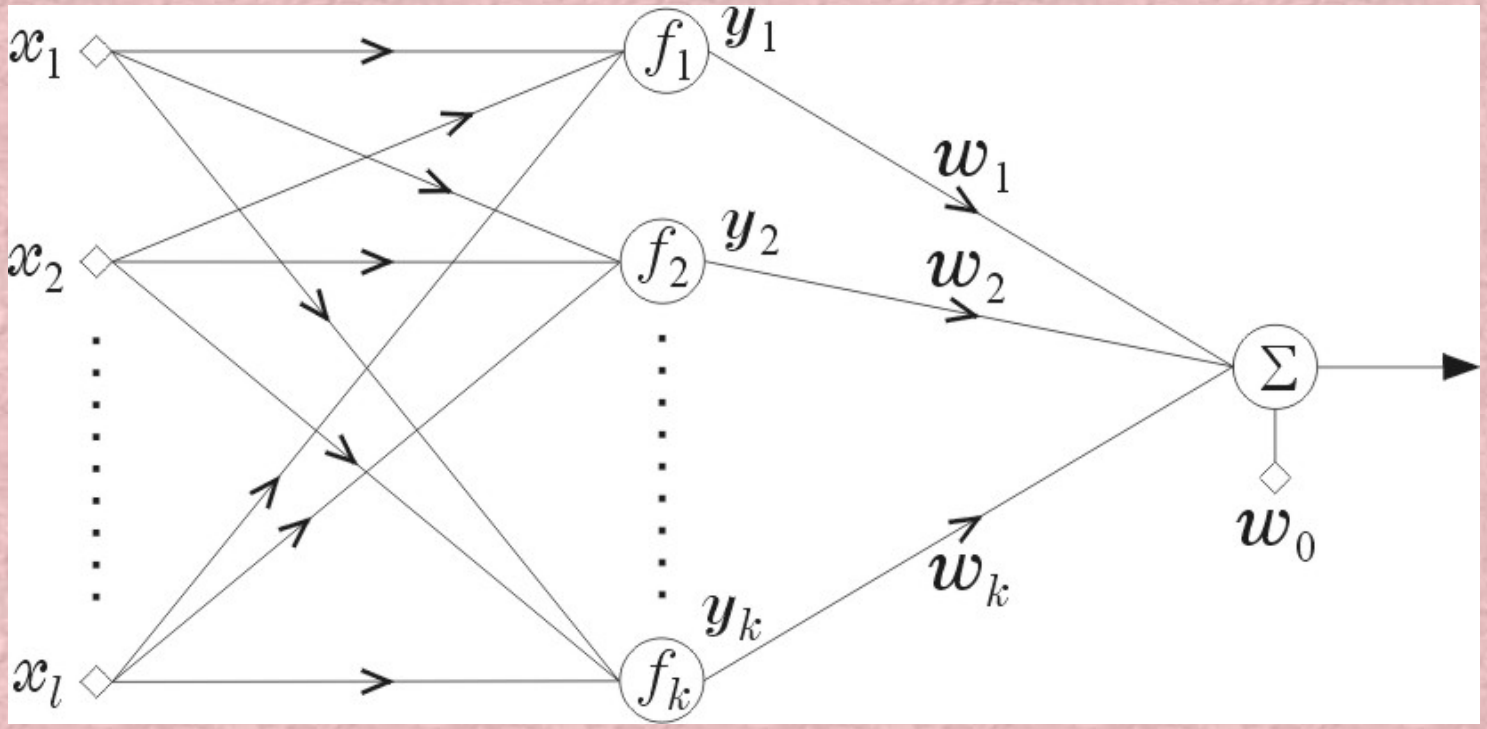
- Generalized linear Classifiers
  - Define the mapping

$$\underline{x} \in R^l \rightarrow \underline{y} \in R^k$$

$$\underline{y} = \left[f_1(\underline{x}), f_2(\underline{x}), .., f_k(\underline{x}) \right]^T$$

> Design a linear classifier in the new space

$$g(\underline{x}) = w_0 + \sum_{i=1}^{k} w_i f_i(\underline{x})$$



- Radial Basis Classifiers

$$f_i(x) = \exp\left(-\frac{\|\underline{x} - \underline{c}_i\|^2}{2\sigma_i^2}\right)$$

- **Universal Aproximators:**

  Multilayer perceptrons with sigmoid functions and RBF networks can arbitrarily approximate any non-linear function.

- Nonlinear Support Vector Machines
  - Remember from the linear case

$$g(\underline{x}) = \sum_{i=1}^{N_s} \lambda_i y_i \underline{x}_i^T \underline{x}$$

$\underline{x}_i^T \cdot \underline{x} \rightarrow$ inner products

- Choose the mapping

$$\underline{x} \in R^l \xrightarrow{\Phi} \underline{y} \in R^k$$

  - So that

$$\sum_r \underline{\Phi}_r(\underline{x})\underline{\Phi}_r(\underline{z}) = K(\underline{x}, \underline{z})$$

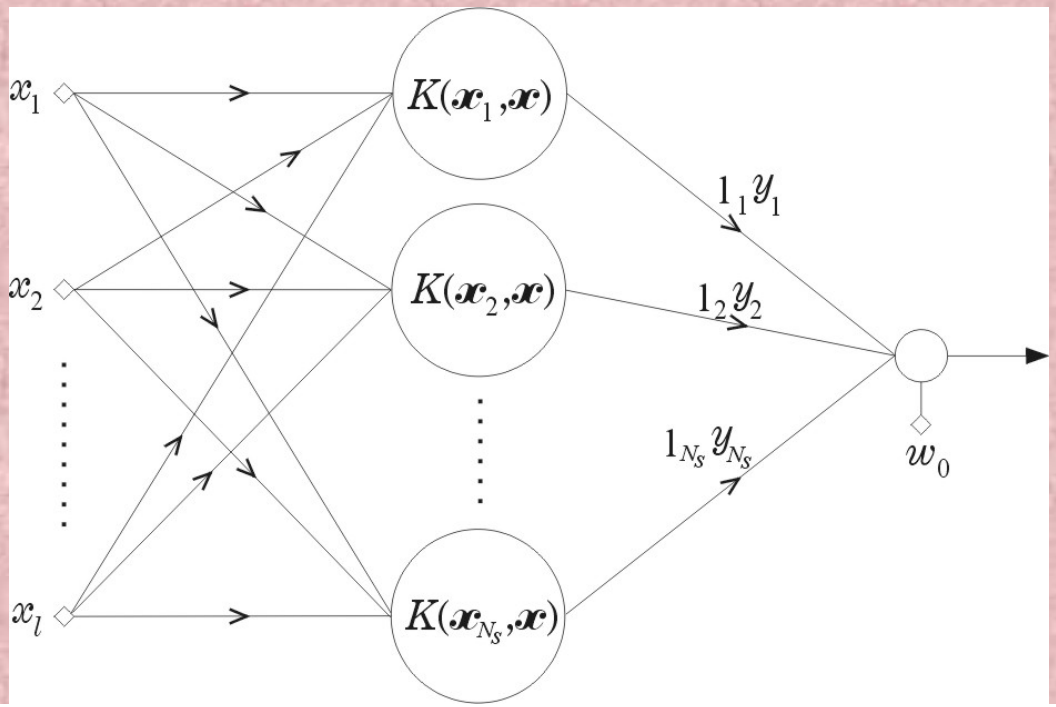
  - That is, inner products in the high dimensional space are expressed in terms of a Kernel function of the input low dimensional space.

- The above is guaranteed by Mercer's theorem, for a class of kernel functions

> Example:

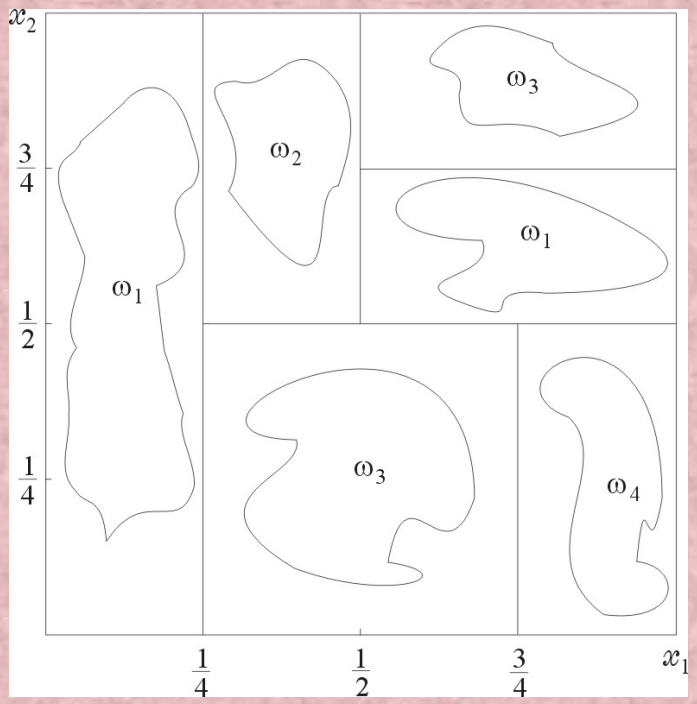
$$\underline{x} \equiv [x_1, x_2]^T \in R^2 \rightarrow \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} \equiv \underline{y} \in R^3$$

$$\underline{y}_i^T \underline{y}_j = (\underline{x}_i^T \underline{x}_j)^2$$



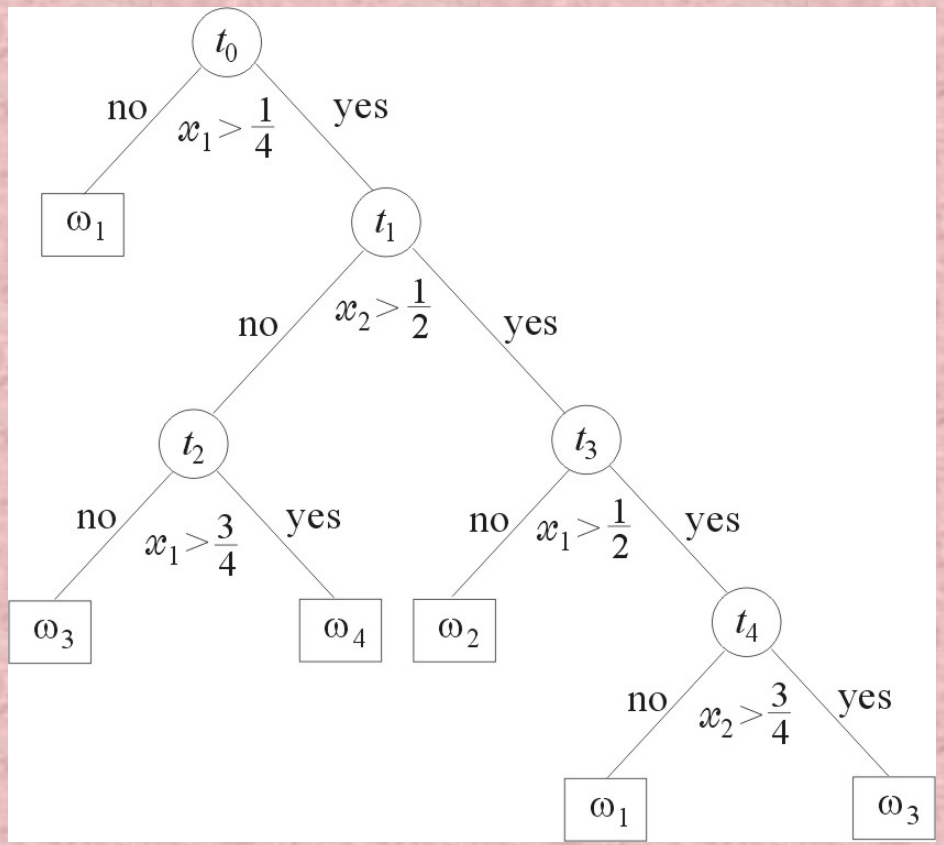
Decision Trees

- These are Multistage decision systems in which classes are sequentially rejected until we reach into a finally accepted class.
- Feature Space is split into regions in a sequential manner



> Decisions / rejections are performed by applying a sequence of questions applied to individual features xi, i=1, 2, .., l

Is feature xi <= α ??



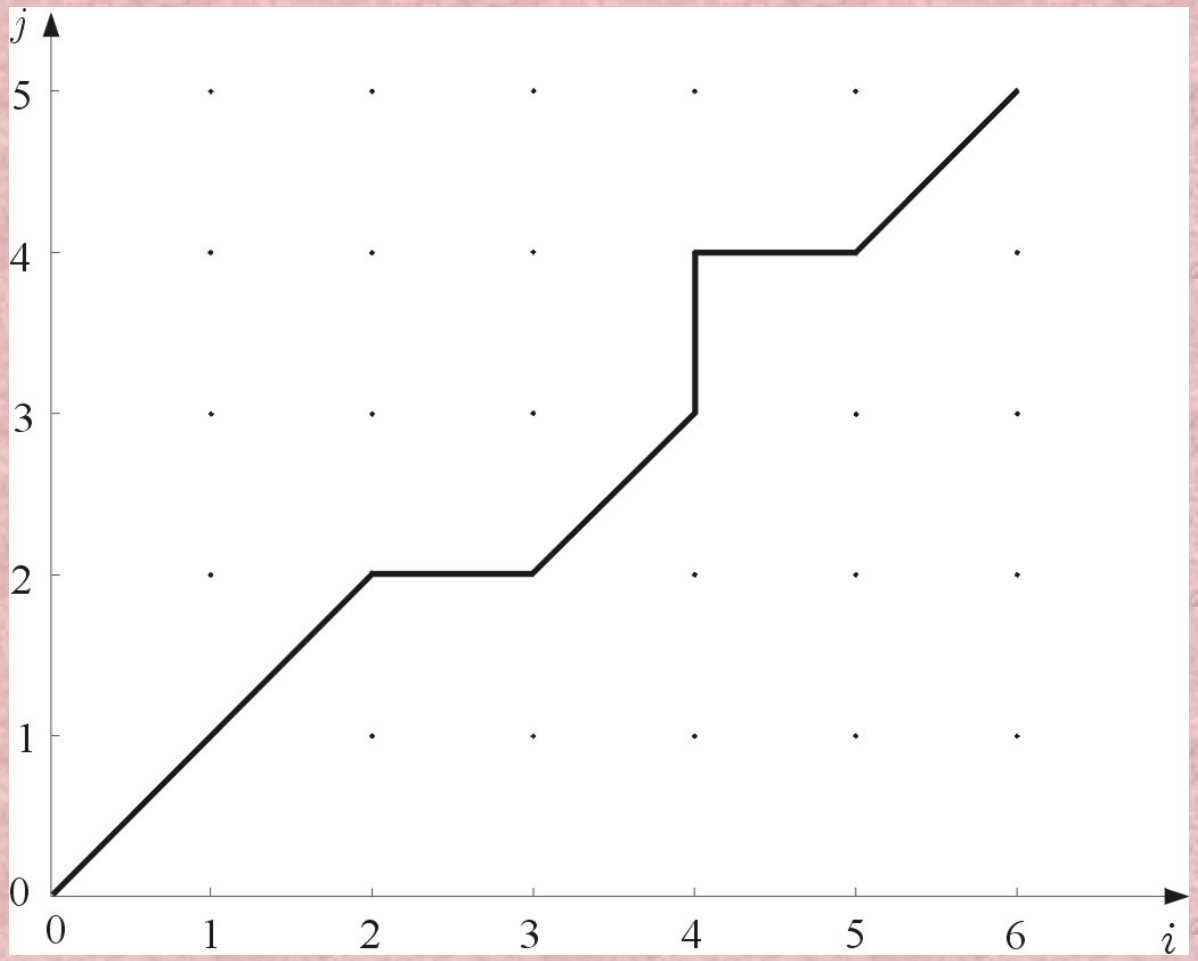
- **Template Matching**:

  Each class is represented by a single reference pattern. An unknown pattern is classified to a class, according to its "similarity" to the corresponding reference pattern.

- A Similarity Measure, or Matching Cost must first be defined.

  - Optimal path searching techniques.
    Each pattern is represented by a sequence of feature vectors

- Reference pattern $\Rightarrow \underline{r}(i),\ i = 1,2,..I$
- Unknown pattern $\Rightarrow \underline{t}(j),\ j = 1,2,..J$



- Construct ALL possible paths through the grid
- For each point (i,j) of the grid define a cost, e.g.

$$d(i,j) = \left\| \underline{r}(i) - \underline{t}(j) \right\|$$

- Similarity between two patterns is defined as the total cost of the optimal path

$$D = \sum_{k=0}^{K-1} d(i_k, j_k)$$

- That is D minimum

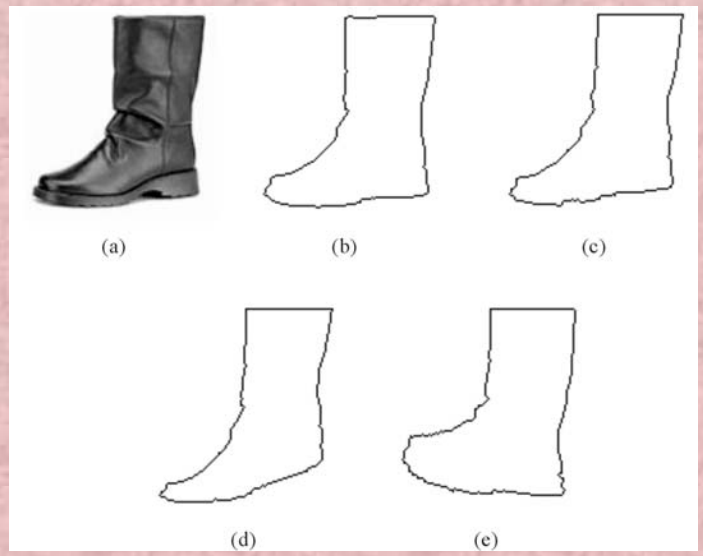
> Typical Examples :

. EDIT distance for matching written words

. Dynamic Time Warping

> Deformable Template Matching

- The philosophy : The reference and test patterns very often do not look exactly the same. Thus, allow this to be accommodated in the similarity measure.
- Create deformed variants of the original reference pattern (prototype)



- The Steps :
  - Adopt a parametric transformation to deform the reference prototype

$$T_{\xi}[\underline{r}]$$

  - Adopt a cost that measures disimilarity between the test pattern and the deformed

$$E_m(\underline{\xi}) : \underline{t} \leftrightarrow T_{\underline{\xi}}[\underline{r}]$$

  - Adopt a cost the measures deformation energy, i.e. dissimilarity between $\underline{r}$, $\underline{T\xi}[\underline{r}]$

$$E_d(\underline{\xi}) : \underline{r} \leftrightarrow T_{\underline{\xi}}[\underline{r}]$$

. Choose $\underline{\xi}$:

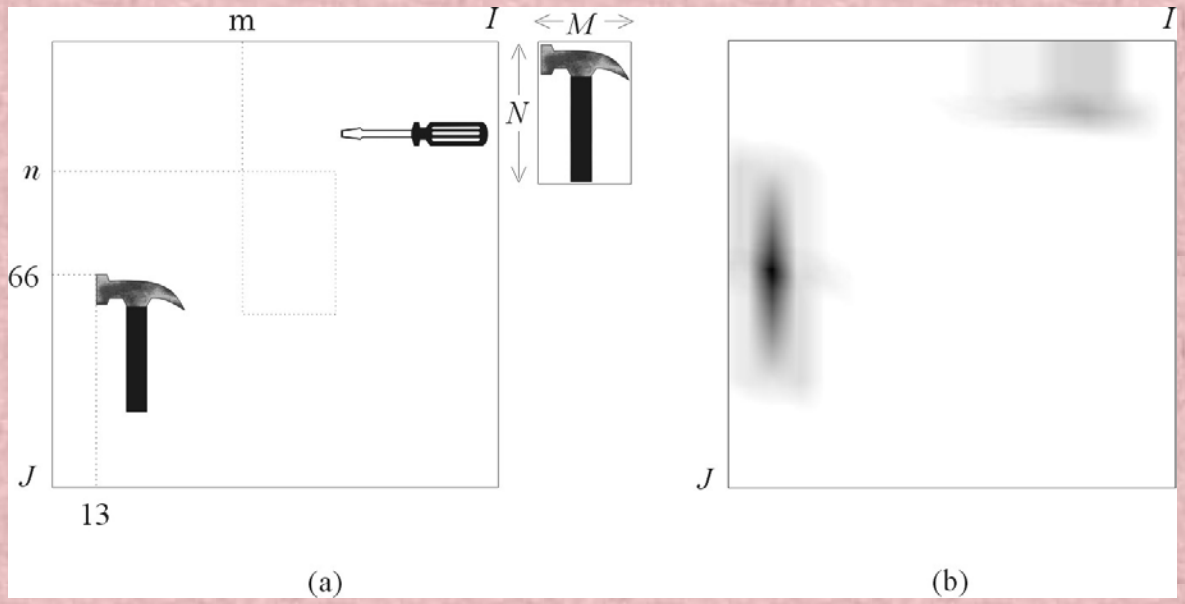
$$\underline{\xi} : \min_{\underline{\xi}} \left[E_m(\underline{\xi}) + E_d(\underline{\xi}) \right]$$

. Best matching cost between $\underline{t}$ and $\underline{r}$:

$$E_m(\underline{\xi}_{min}) + E_d(\underline{\xi}_{min})$$

Correlation Measures

- **The Goal** : Given a block of recorded data find if a specific known reference pattern is contained and where it is located.
- Compute the correlation between the test pattern and the reference pattern and find the maximum



- Context Dependent Classification
  - **The task** : A number N of patterns $\underline{x}_1,..,\underline{x}_N$, is given for classification to a set of classes $\omega_1, \omega_2, .., \omega_M$. However classes are NOT independent. Assigning a pattern $\underline{x}_i$ to a class depends
    - On its own value $\underline{x}_i$
    - On the values of the other patterns $\underline{x}_j$, $i \neq j$
    - On the existing relation among the classes

– For the sequence of patterns

$$X : \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$$

and M possible classes, there is a total of M^N possible sequences of corresponding classes;

$$\Omega_i : \omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_N}$$

$$i = 1,2,\ldots, M^N$$

where

$\underline{x}_i$ is assigned to ω_{i_k}

$$\omega_{i_k} \in \{1,2,..,M\}$$

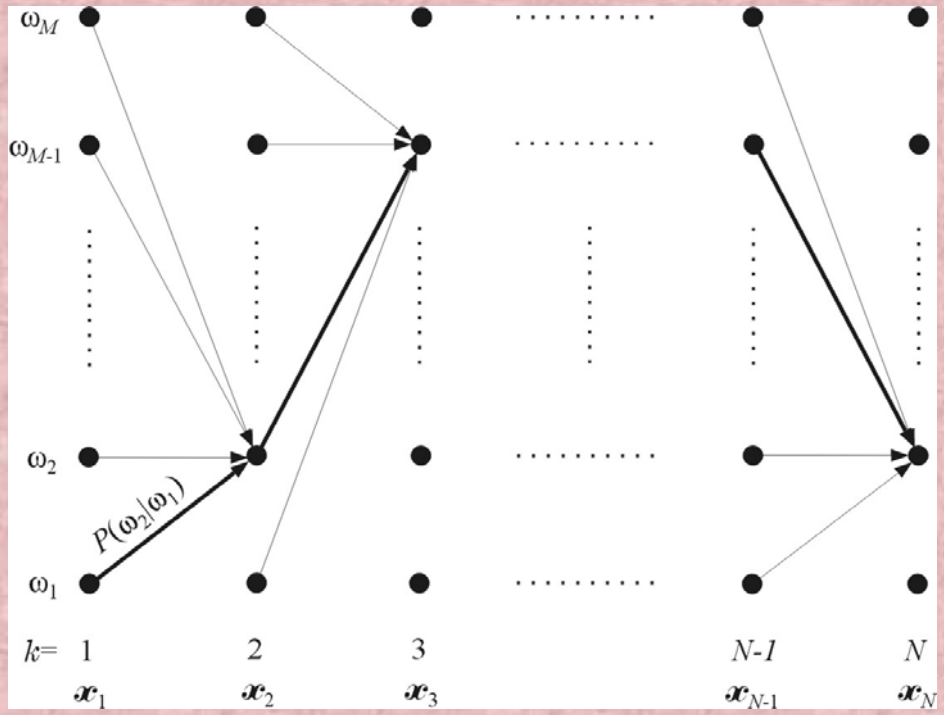
Bayesian Classification

- Assign x : to Ω_i : $\arg\max_i P(\Omega_i \mid X)$
- Complexity very high for computing the maximum

The Viterbi Algorithm

Assume a Markovian class dependence model

$$P(\omega_{i_k} \mid \omega_{i_{k-1}}, \omega_{i_{k-2}}, .., \omega_{i_1}) =$$
$$= P(\omega_{i_k} \mid \omega_{i_{k-1}})$$



- Applications
  - Channel Equalization, where classes are related to states
  - Hidden Markov Modeling

- Feature Generation

  A problem dependent stage. Typical directions for generating features:

  - Statistical
    - Moments
    - Parametric models (AR, ARMA)
    - Fractal dimension

- Geometric
  - Perimeter
  - Curvature
- Transform based features
  - Fourier Transforms
  - DCT, DST
  - Wavelet transforms
  - Principal Component Analysis (PCA)
  - Singular Value Decomposition
  - Independent Component Analysis (ICA)

Principal Component Analysis (PCA)

- Generate, from data vector $\underline{x}$, mutually uncorrelated features

$$\underline{y} = A^T \underline{x}$$

$$E[y_i y_j] = \delta_{ij}, \quad i, j = 1,2,\ldots,N$$

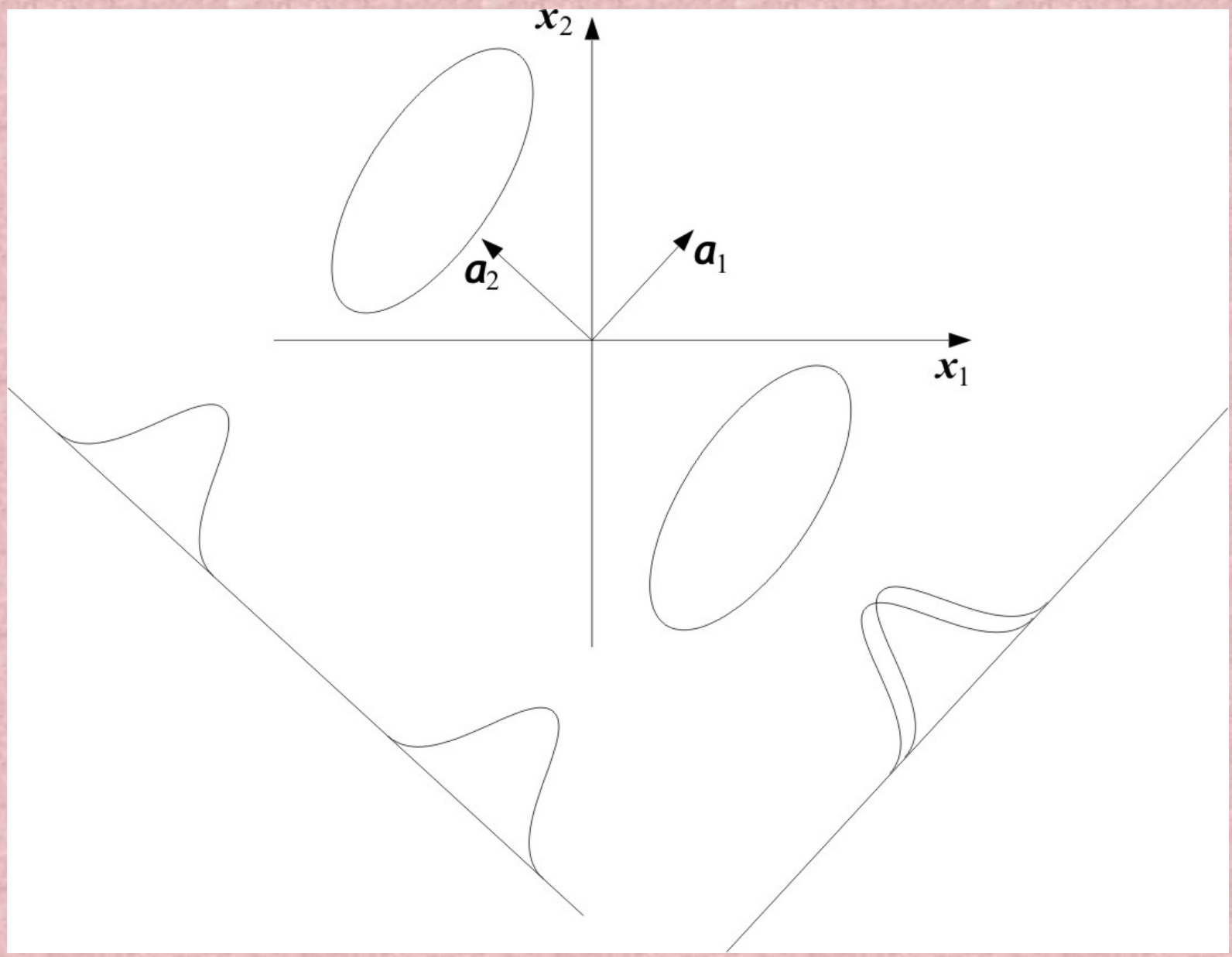
- The above is fulfilled if A is chosen to have the eigenvectors, $\underline{a}_i, i = 1,2,\ldots,N$, of $R_x = E[\underline{x}\underline{x}^T]$ as its columns.

. Choose $y_i, i = 1,2,\ldots,m < N$, to correspond to the m largest eigenvalues

. Then

$$\hat{\underline{x}} = \sum_{i=1}^{m} y_i \underline{a}_i$$

is the MMSE approximation to $\underline{x}$.


x_2
a_2
a_1
x_1

Independent Component Analysis (ICA)

- Barlow's hypothesis: The outcome of the early processing performed in our visual cortical feature detectors might be the result of a redundancy reduction process.

- The neural outputs are as independent as possible.

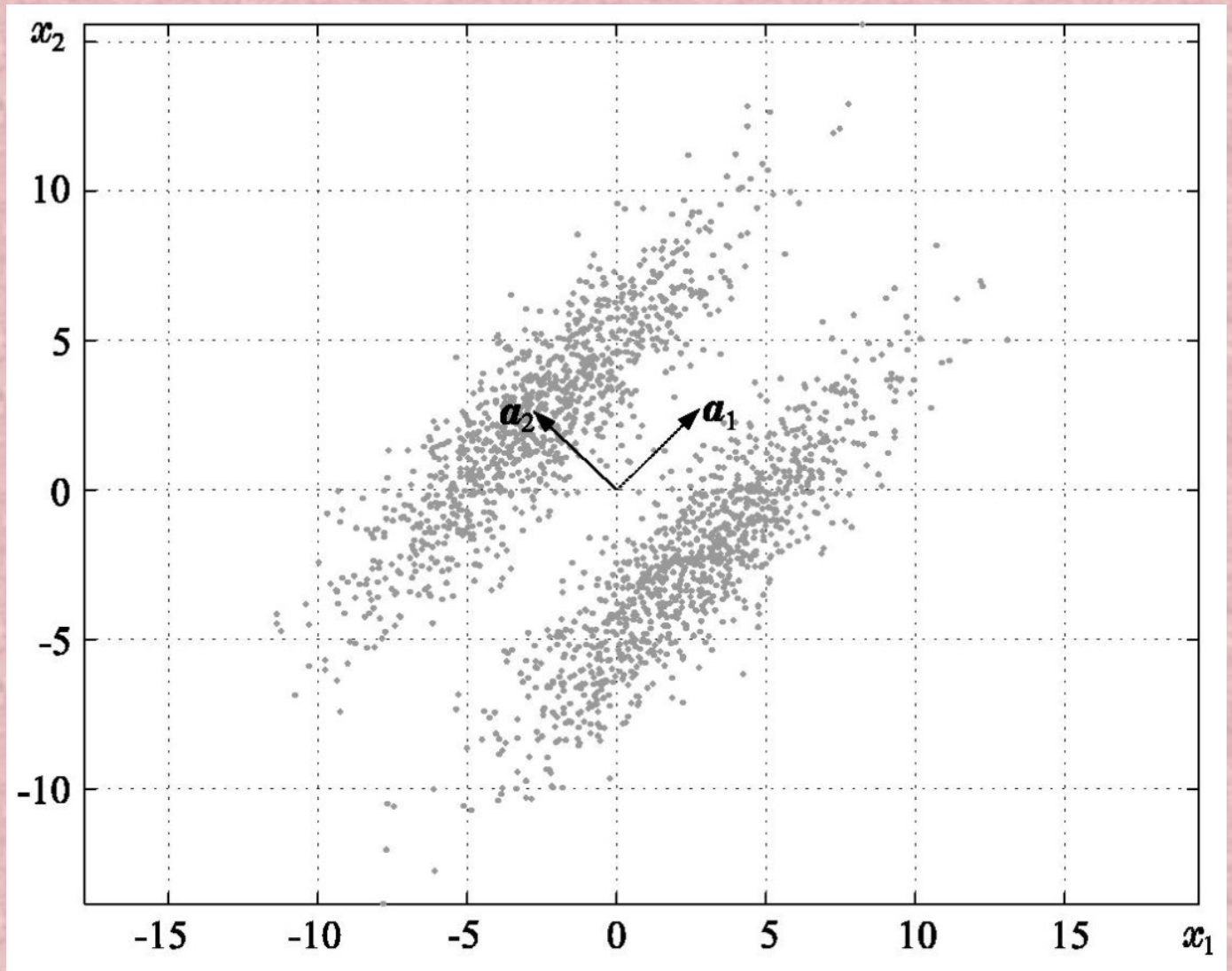
- Given $\underline{x}$ produce $\underline{y}$

$$\underline{y} = W\underline{x}$$

so that y_i, y_j are as statistically independent as possible.

- Higher-order statistics are required to ensure independence.
- Non-Gaussian processes must be assumed.
- The goal: Given N independent components $y_i, i = 1,2,\ldots,N,$ choose the 1 best ones.
- What is best?

- The least resemblance to Gaussian, the better it is.
- A Gaussian random process has all its higher-order cumulants identically zero.
- The larger the absolute value of kurtosis (4^{th}-order cumulant), the less the resemblance to Gaussian.



$k_4(y_2) = -1.7$

$k_4(y_1) = 0.1$

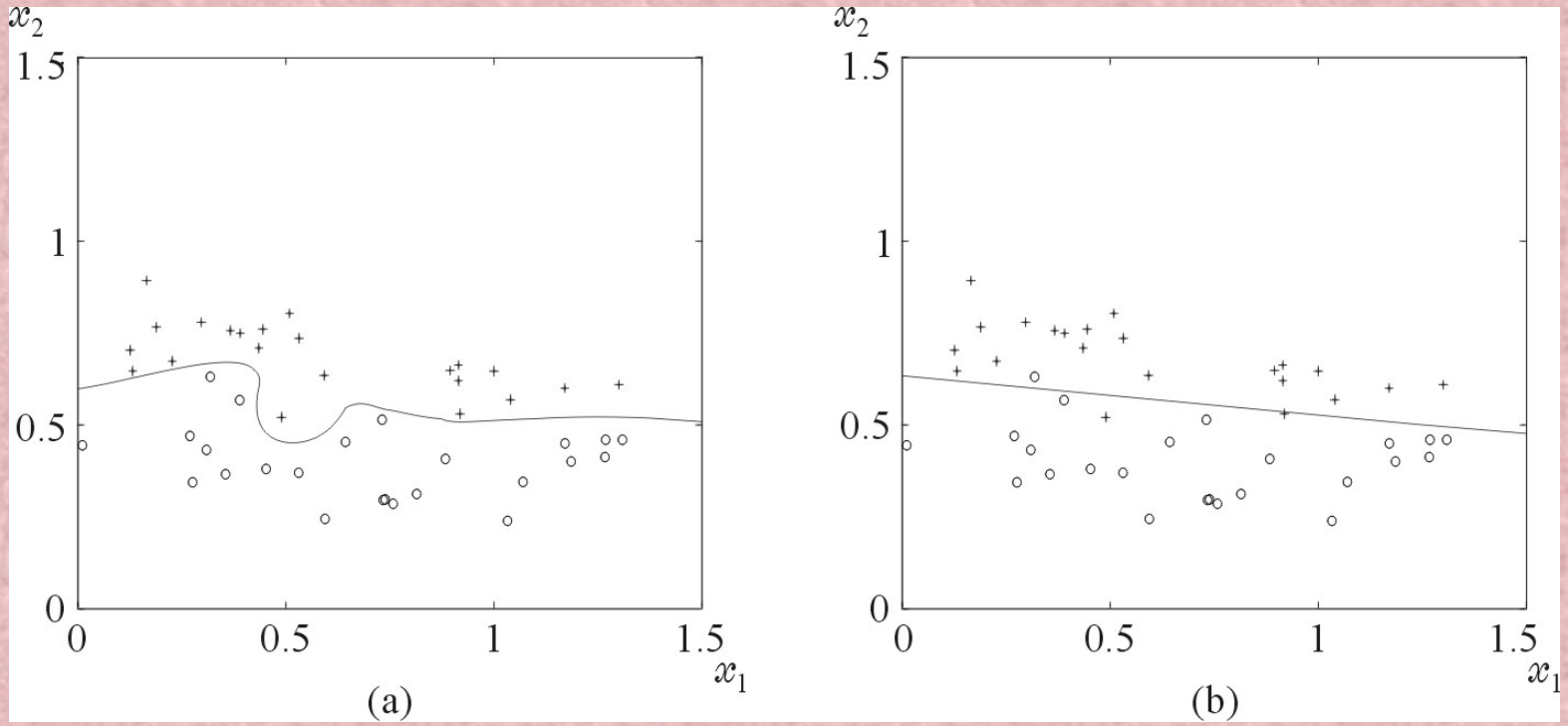
- Feature Selection
  - **The goal**: Select the l most informative features from the (large) set of generated ones. Two questions are involved.
    - What is the "optimal" number l
    - Which are the best l features
  - **Generalization Performance of a classifier**: This refers to the capability of a classifier to perform well, when faced with data unknown to it, i.e., with data outside the training set.

- For finite number N of training data
  - l affects the number of free parameters of a classifier.The number of free parameters must be
    - large enough, w.r. to N,

For the classifier to learn what makes "*similar*" the patterns *within* each class and what makes one class *different* from another

- small enough

Not to learn underlying differences among data of *same* class


x_2
1.5
1
0.5
0
0
0.5
1
1.5
x_1
(a)
x_2
1.5
1
0.5
0
0
0.5
1
1.5
x_1
(b)

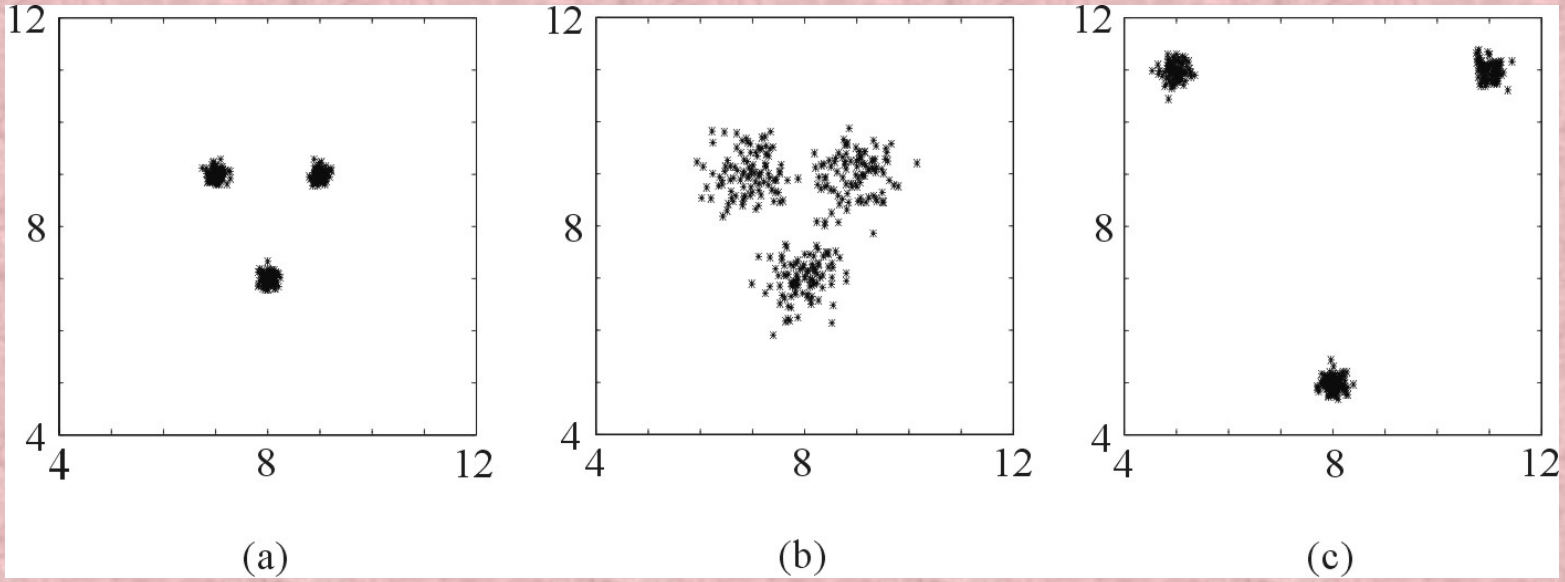
- Vapnic - Chernovenkis bound and Vc dimension
  - Let P_e^N be the empirical error based on the training data samples
  - Let P_e the true error
  - P_e^N is close enough to P_e with high probability provided N is large enough w.r to the V_c dimension
  - V_c is a measure of the intrinsic capacity of a classifier and it is related to the maximum number of dichotomies the classifier can perform on N training points

- **An example** : For a multilayer perceptron

$$2\left\lfloor \frac{k_n^h}{2} \right\rfloor l \le V_c \le 2k_w \log(ek_n)$$

  - l input space dimension
  - k^h number of hidden layer nodes
  - k_n total number of nodes
  - k_w total number of weights

- How to select the best l features
  - Feature vectors must exhibit
    - Large between - class distance
    - Small within class variance



- System evaluation
  - The error counting technique is adopted using a appropriately selected test set .
  - A theoretically derived rule of thumb for the size L of the test set, for an error probability of P, is

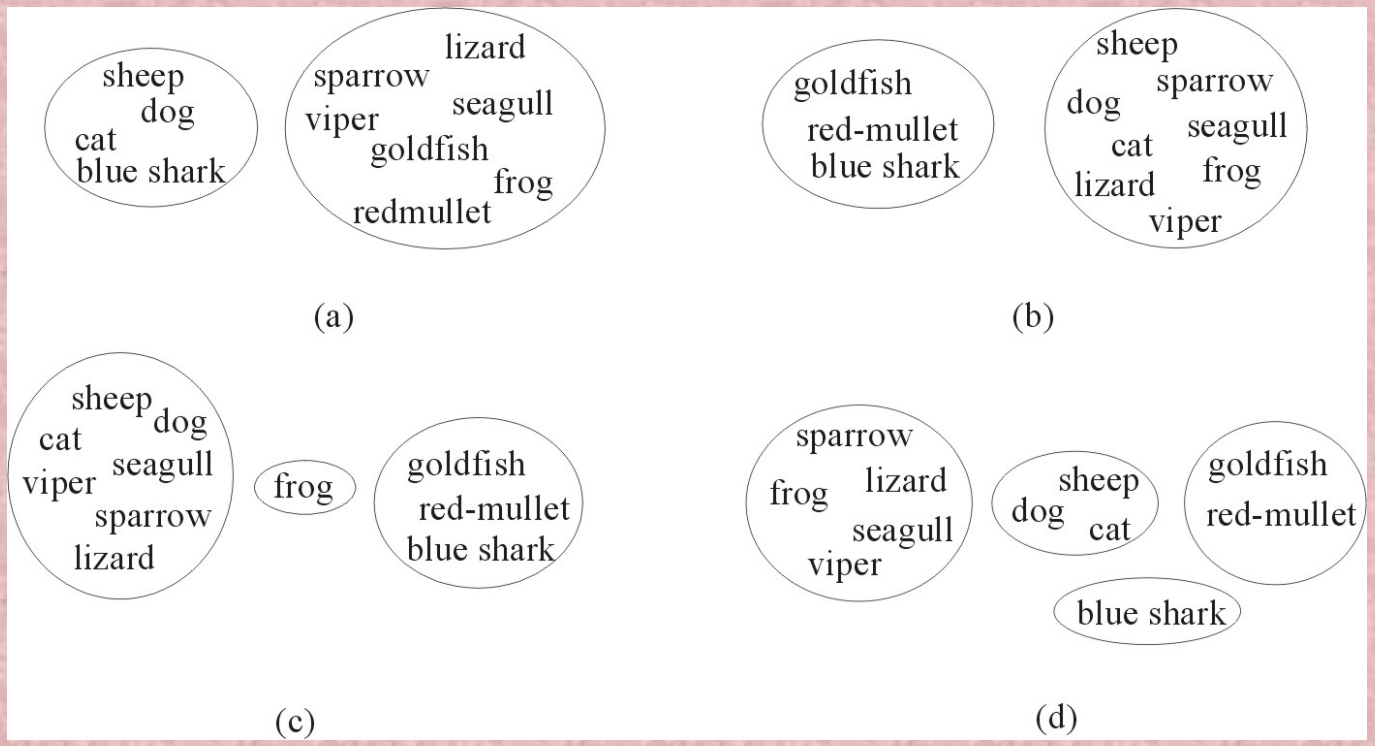
$$L \approx \frac{100}{P}$$

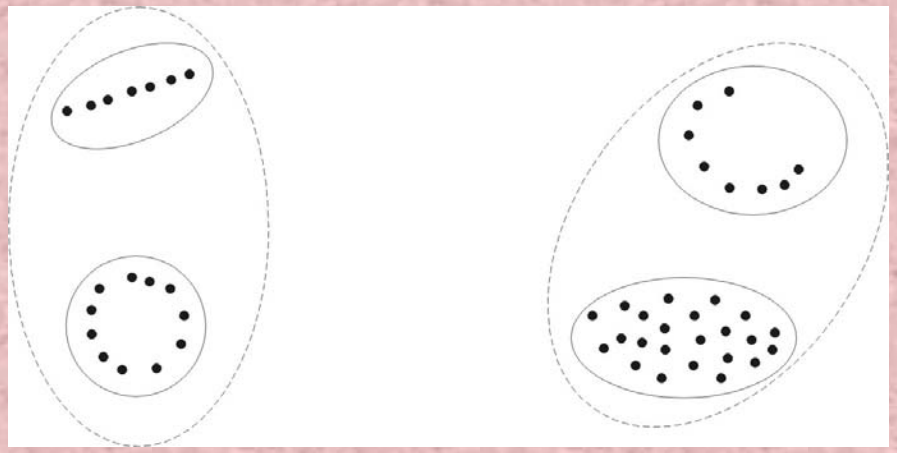
For P=0.01, L=10 000, for P=0.03, L=3000

- **The Holdout Method**: Divide the training set into two subsets. *One for training* the classifier and *one for testing* it.

- **Leave-one-out method**: From the N training points use N-1 for training and one for testing. Repeat this N times, each time with a different point for testing.This guarantees independence between training and testing. Count the errors and average them to obtain the error percentage.

- Clustering
  - The major task of clustering is to reveal the organization of patterns into sensible clusters
  - Subjectivity is a reality one has to live with when dealing with clustering


sheep
dog
cat
blue shark
lizard
sparrow
seagull
viper
goldfish
frog
redmullet
(a)
goldfish
red-mullet
blue shark
sheep
sparrow
dog
seagull
cat
lizard
frog
viper
(b)
sheep
dog
cat
seagull
viper
sparrow
lizard
frog
goldfish
red-mullet
blue shark
(c)
sparrow
frog
lizard
seagull
viper
sheep
dog
cat
goldfish
red-mullet
blue shark
(d)



- Definitions of clustering
  - Given $X=\{\underline{x}_1, \underline{x}_2, .., \underline{x}_N\}$ an m-clustering of X, is the partition of X into m sets, $C_1, C_2, .., C_m$,

$$C_i \neq \varnothing \quad i = 1,.., m$$
$$\cup_{i=1}^{m} C_i = X$$
$$C_i \cap C_j = \varnothing, \quad i \neq j, \quad i, j = 1,2,.., m$$

  - Fuzzy clustering of X into m clusters is characterized by m membership function, u_j

$$u_j : X \to [0,1], \quad j = 1,2,....,m$$

$$\sum_{j=1}^{m} u_j(x_i) = 1, \quad i = 1,2,..,N$$

$$0 < \sum_{i=1}^{N} u_j(x_i) < N \ , j = 1,2,..,m$$

- To perform any clustering task the following steps are required
  - Feature Selection : Features must be selected so that to encode as much information as possible
  - Proximity measure : A measure that quantifies the terms “similar” or “dissimilar”
  - Clustering Criterion : It depends on the interpretation the expert gives to the term “sensible”. The criterion can be expressed via a cost.
  - Clustering Algorithm : Having adopted a proximity measure and a criterion, the algorithm will reveal the srtucture

- Validation of results
- Interpretation of results

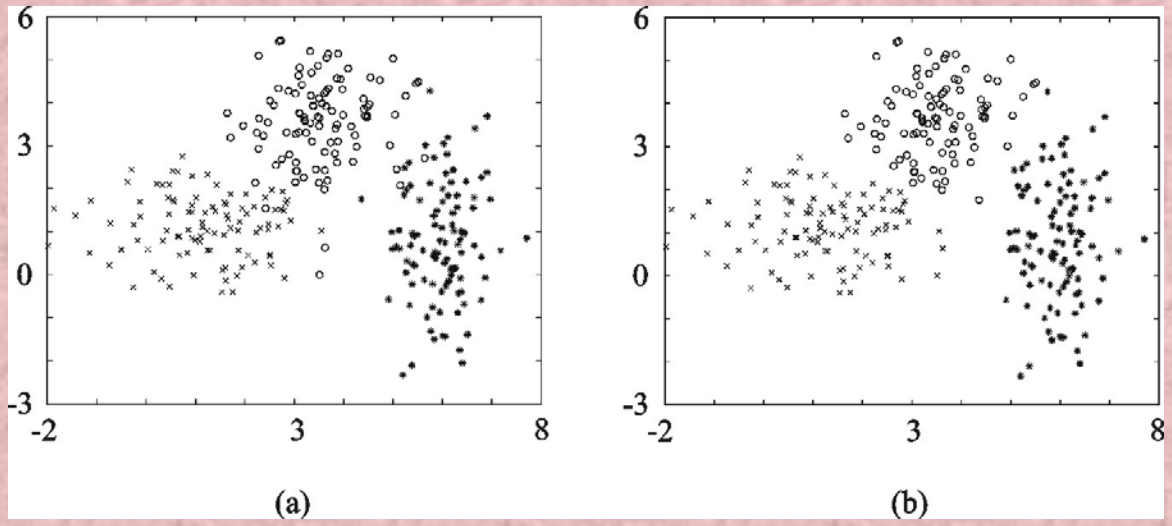
- Clustering Algorithms
  - The optimal way to partition X into m clusters is to consider ALL possible partitions and select the "best". This is an NP-hard problem
  - A clustering algorithm searches a small fraction of all possible partitions

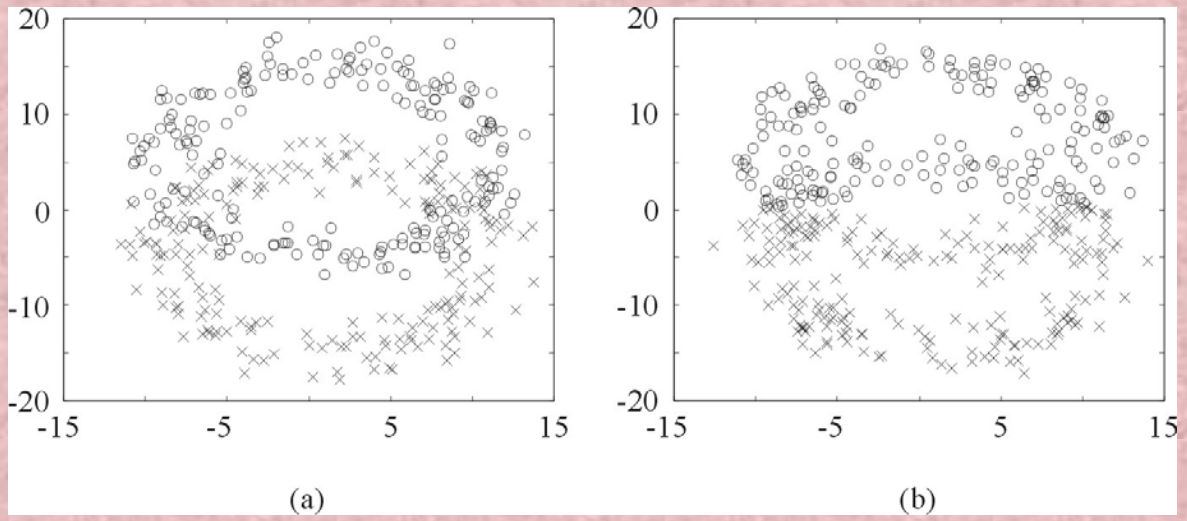
- Categories of clustering algorithms
  - **Sequential** : A *Single* clustering is produced. The data are presented sequentially, at least once
  - **Hierarchical algorithms** : A *sequence* of clusterings is recovered
    - Agglomerative algorithms : The clusterings are obtained in a decreasing number of clusters at each step
    - Divisive : These act in the opposite direction

- Algorithms based on cost function optimization
  - Hard clustering algorithms
  - Fuzzy algorithms
  - Possibilistic algorithms
- Other
  - Genetic
  - Stochastic relaxation
  - Competitive learning

- Choice of the appropriate algorithm, similarity measure and clustering criterion depends on the specific problem

Example 1: Use of the Generalized Mixture Decomposition Scheme





Example 2 : Use of the Fuzzy Shell Clustering Scheme