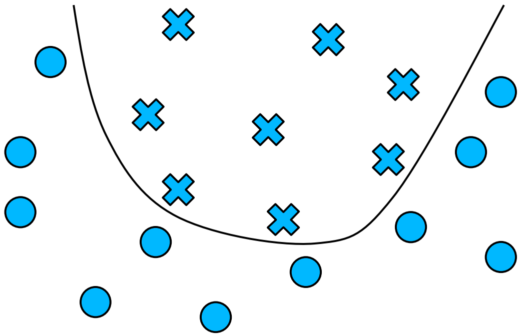
Pattern Recognition (Örüntü Tanıma)

Basic Approaches and Techniques



Neşe Yalabık

Hasat Projesi Eğitim Programı
ODTÜ, 31 Ağustos 2010

Outline

Pattern Recognition: Definitions and Objectives

Bayes Classifiers

Estimation of Parameters

Features and Feature Selection

K-Nearest Neighbor Classifiers

Linear Discriminant Function Classifiers

Tree Classifiers

Clustering

Training and Performance Testing in PR

References

Pattern Recognition: Definitions

Pattern Recognition(PR): The process of machine perception for an automatic labeling of an object or an event into one of the predefined categories.

Pattern Classification: Final step in a PR system

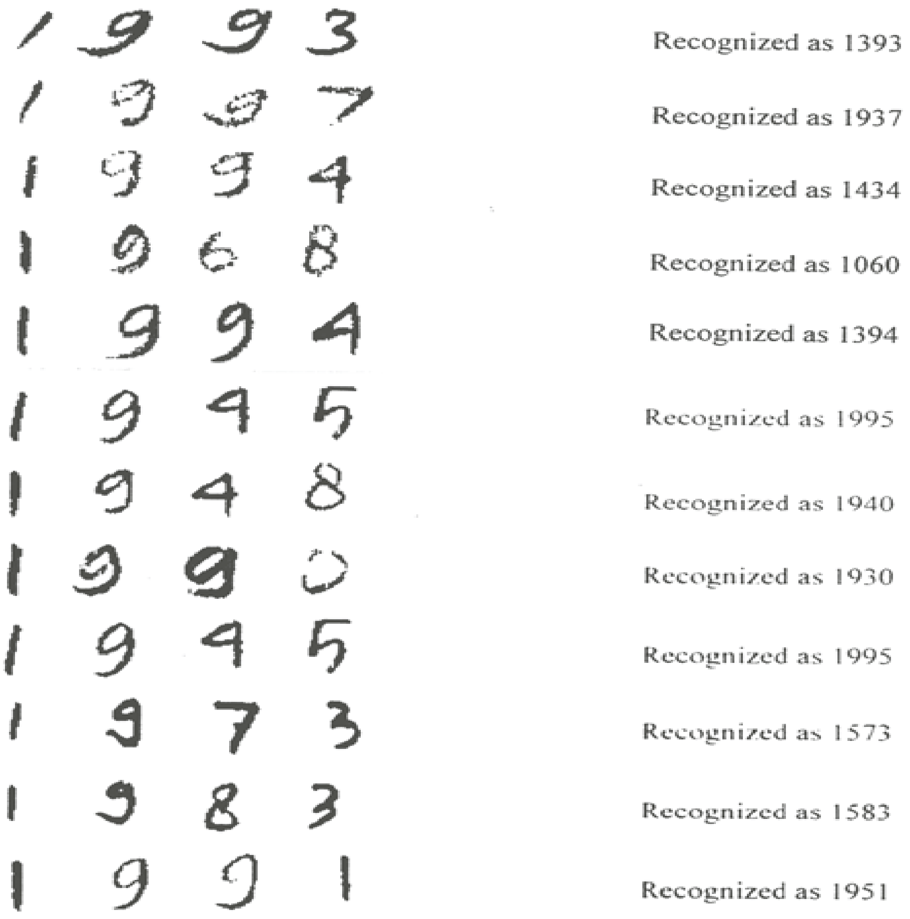
We human beings do pattern recognition everyday.
We "**recognize**" and **classify** many things,
even if it is corrupted by **noise**, **distorted** and **variable**.

Classification is the result of **recognition:learning, categorization, generalization**

A problem is a PR problem only if it involves '**statistical variation**'

Example

We see here that all 9's are different from each other and 9's and 4's can easily be mixed



Recognized as 1393

Recognized as 1937

Recognized as 1434

Recognized as 1060

Recognized as 1394

Recognized as 1995

Recognized as 1940

Recognized as 1930

Recognized as 1995

Recognized as 1573

Recognized as 1583

Recognized as 1951

Example Applications of Pattern Recognition

- Reading hand-written text to classify it into letters and words
- Analyzing fingerprints to find the owner
- Recognizing the faces of people to name them
- Finding buildings in a satellite image
- Naming a gun from its bullet mark (Ballistics)
- Identifying different objects on a conveyor belt
- Analyzing test results in decision support for any illness

Pattern Recognition: Definitions

A Pattern Recognition System consists of the following parts:

- Pre-processing and Feature Extraction
- Learning
- Classification
- Post- processing

Pattern Recognition: Definitions

“Pre-processing and Feature Extraction” Converts 'data' to 'features'

“Data” raw data taken as samples

“Feature” a discriminating, easily measurable characteristics of our data.

“Feature Vector”: A set of variables that represent different features

“Feature Space” : is defined by a feature vector

Pattern Recognition: Definitions

Pre-processing and Pattern Extraction

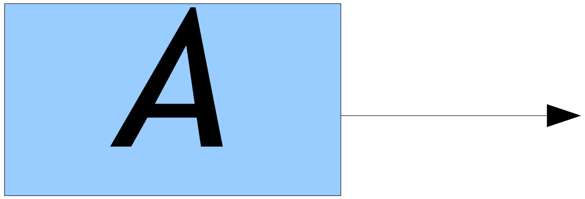
All operations over raw data (such as a remotely sensed image) to enhance and process it , leading to extraction of features .

Includes enhancement, edge extraction, segmentation etc.

Pattern extraction results with a feature vector X

Example

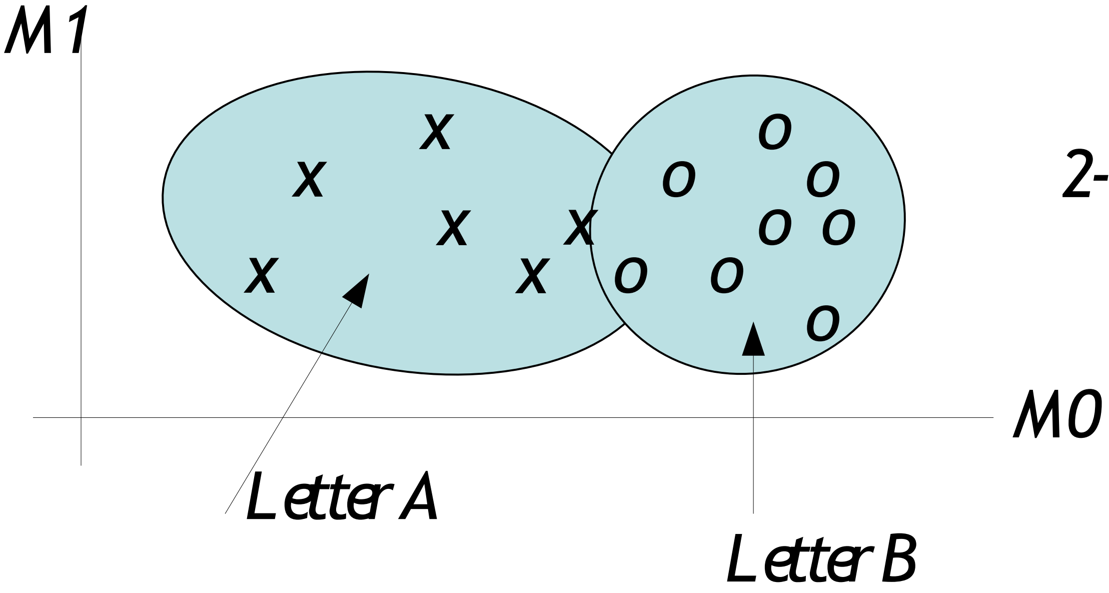
Consider recognition of handwritten characters:



$$[M_0, M_1, \ldots, M_k]$$

Raw data: Bitmap

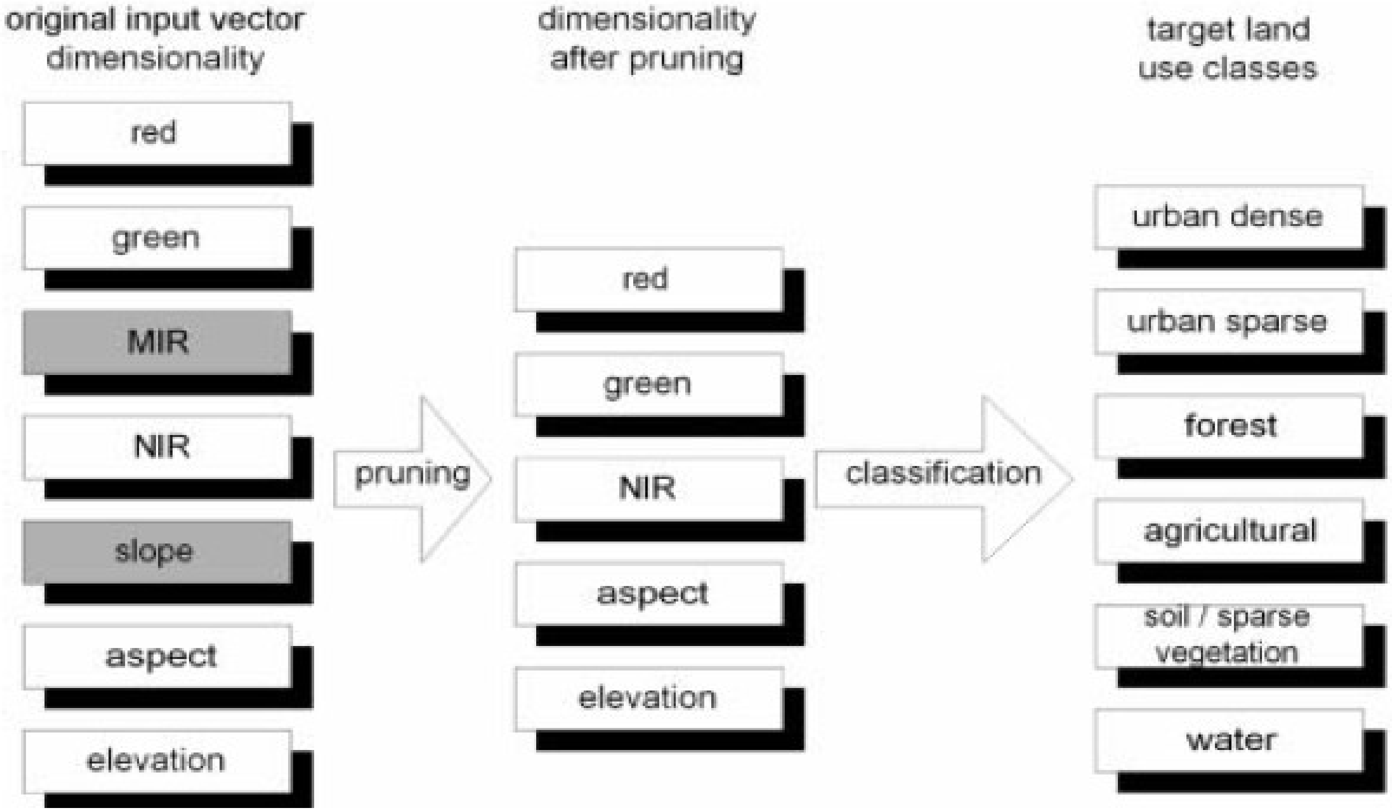
Features: Moment invariants

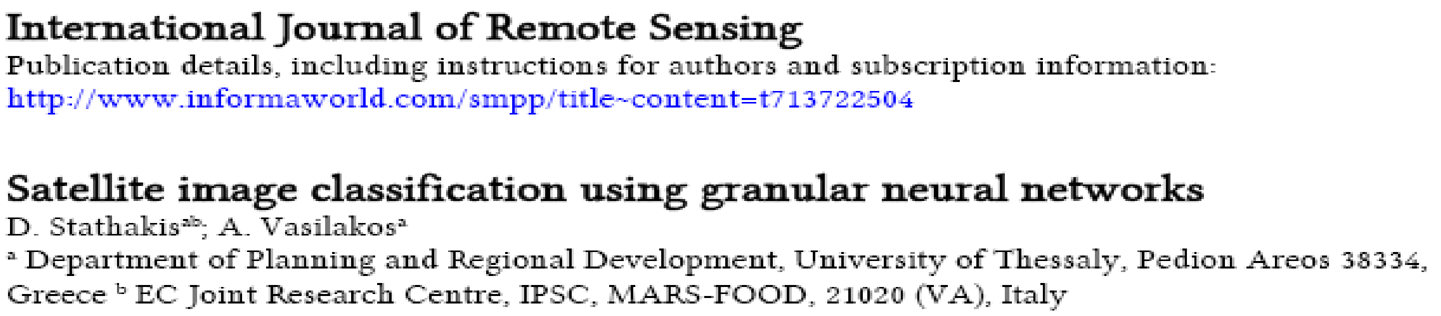


2-d feature space

Example

Clasification of land use from multispectral data




International Journal of Remote Sensing

Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713722504

Satellite image classification using granular neural networks

D. Stathakis[a,b]; A. Vasilakos[a]

[a] Department of Planning and Regional Development, University of Thessaly, Pedion Areos 38334, Greece [b] EC Joint Research Centre, IPSC, MARS-FOOD, 21020 (VA), Italy

Pattern Recognition: Definitions

Learning: Devising a classifier from collected samples with or without known labels (categories)

"Learning samples" Large data sets to be used in training, or estimating parameters, etc. They may be labeled or not.

Given the learning data set with known labels:**supervised learning;**Unknown labels: **unsupervised learning and clustering**

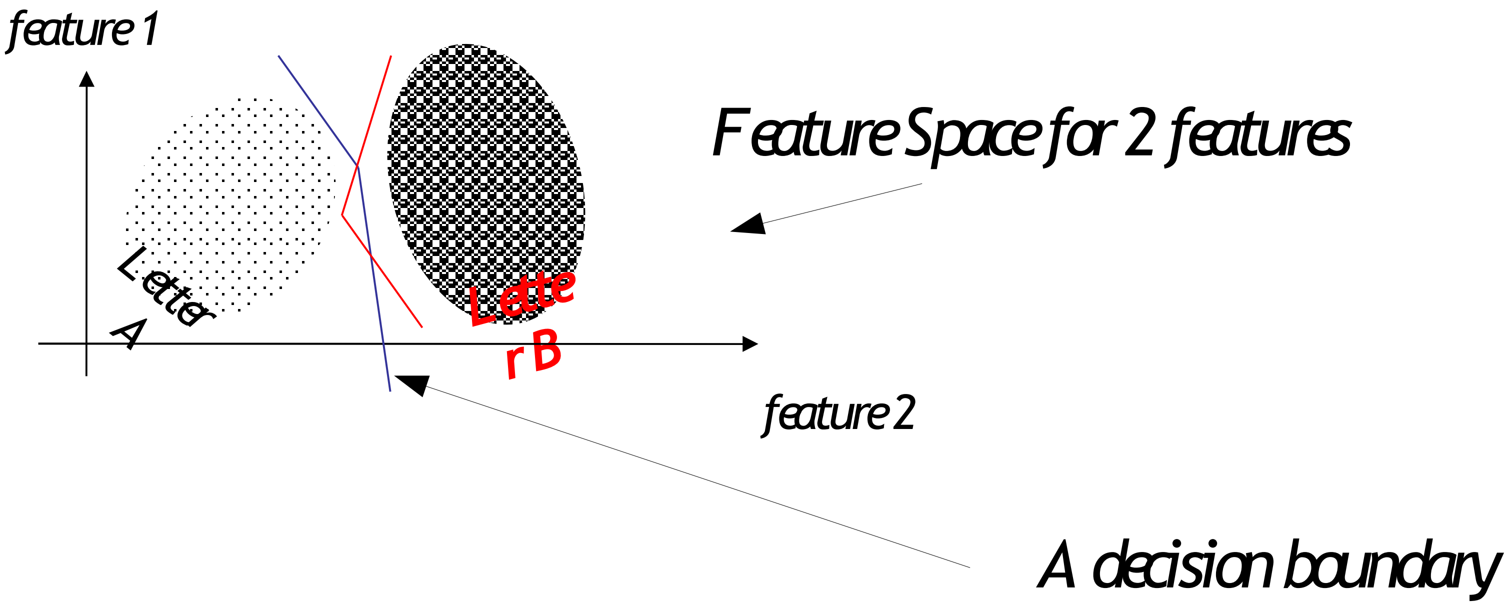
"Test Samples" used in testing the classifier performance.

"Result" a decision on the category sample belongs.

"Performance" How well a classifier correctly recognizes test samples
"Correct Classification ratio" ratio of correctly classified samples to all test samples

Pattern Recognition: Definitions

Classification



Feature Space for 2 features

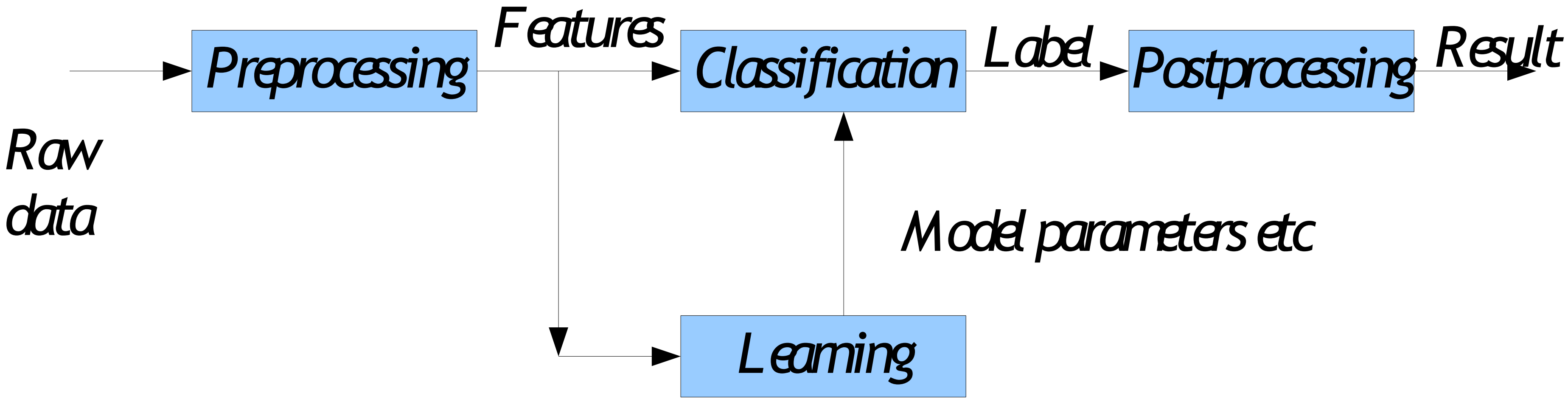
A decision boundary

How do we separate A 's from B 's?
Form a decision boundary
Classify the sample to the side it falls

Pattern Recognition: Definitions

Post- processing: Domain knowledge may be incorparated to correct mistakes, such as using language to correct letter classifiers

A Pattern Recognition System



Objective in PR

Performance criteria: Minimize the average error (at least as good as a human being)

Minimize the risk: wrong decision could be more risky in some cases such as medical diagnosis

Why automise? **Obvious reason: save from time and effort**

(Ex: consensus forms: enter 100 million records into electronic medium).

How do machines solve it: Many different approaches in history

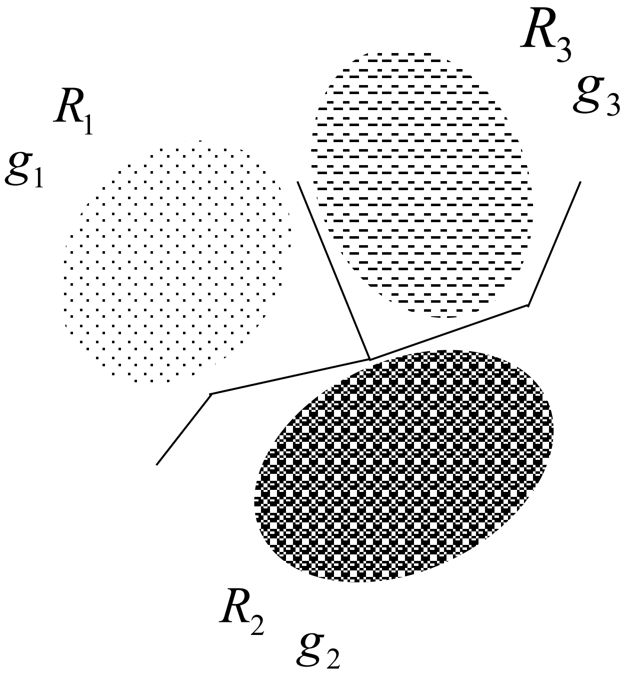
Statistical Pattern Recognition: relies on statistics of collected data

Structural Pattern Recognition: tries to discover the structure inherent in data

(ex: may assume letters are composed of strokes etc)

Statistical Approach to P.R

$X = [X_1, X_2, ..., X_d]$



Dimension of the feature space: d

Set of different states of nature: $\{\omega_1, \omega_2, ..., \omega_c\}$

Categories: c

find R_i $R_i \cap R_j = \varphi$ $uR_i = R^d$

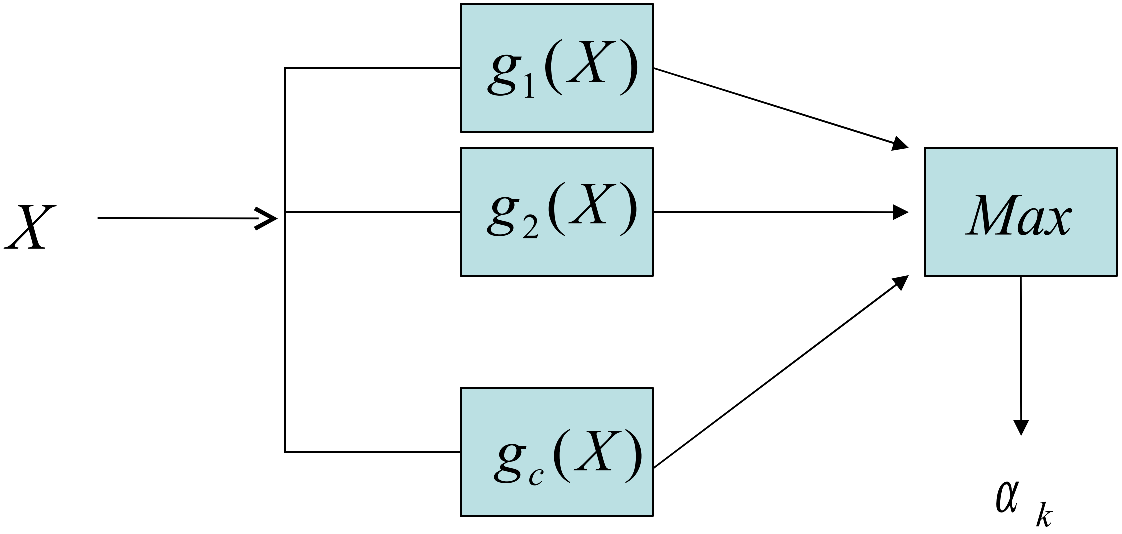
set of possible actions (decisions): $\{\alpha_1, \alpha_2, ..., \alpha_a\}$

Here, a decision might include a 'reject option'

A Discriminant Function $g_i(X)$ $g_i(X) \geq g_j(X)$

in region R_i ; decision rule : if α_k $g_k(X) > g_j(X)$ $1 \leq i \leq c$

A Pattern Classifier



So our aim now will be to define these functions $g_1, g_2, \ldots, g_c$ to *minimize* or *optimize* a criterion.

Bayes Classifiers

A Parametric approach which assumes that the feature vectors are random variables with known probability distributions.

'Bayes Decision Theory' is used or minimum-error-minimum risk pattern classifier design.

It is assumed that if a sample X is drawn from category wi, it is a random variable represented with a multivariate probability density function called

‘Class-conditional density function'

$$P(X \mid Wi)$$

We also know a-priori probability $P(\omega_i)$

Then, we can talk about a decision rule that minimizes the probability of error.

Suppose we have the observation X

This observation is going to change a-priori assumption to a-posteriori probability:

$$P(\omega_i | X)$$

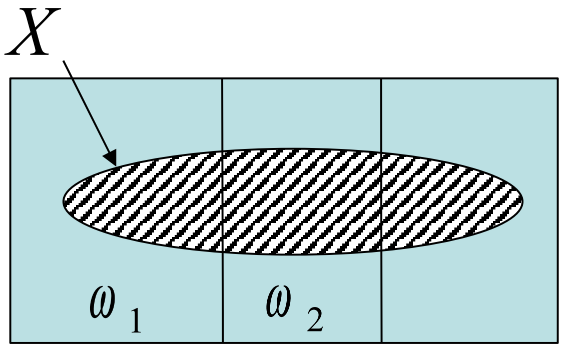
which can be found by the Bayes Rule

$$P(\omega_i | X) = P(\omega_i, X) / P(X)$$

$$\boxed{= \frac{P(X|\omega_i).P(\omega_i)}{P(X)}}$$

$P(X)$ can be found by Total Probability Rule:

When

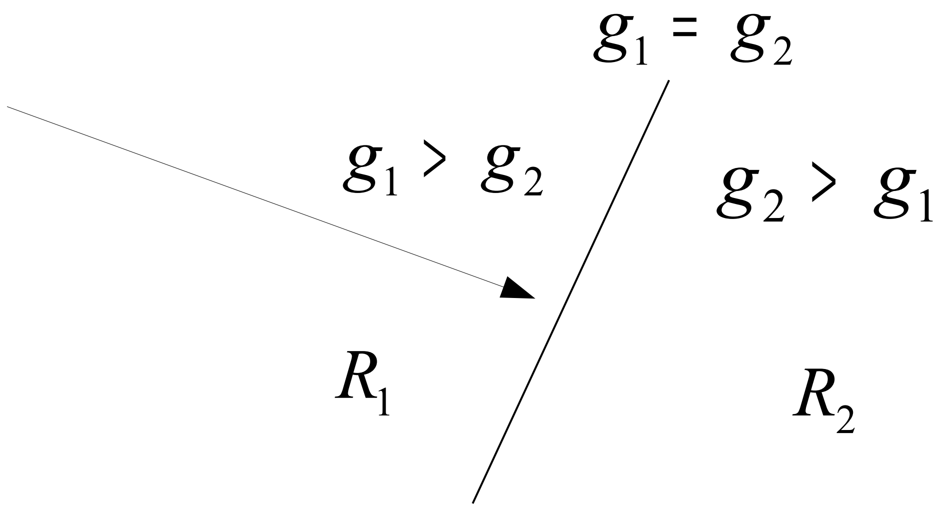


$$P(X) = \sum_{i=1}^{c} P(\omega_i, X)$$

Decision Rule: Choose the category with highest a-posteriori probability, calculated as above, using Bayes Rule.

then, $g_i(X) = P(\omega_i | X)$

Decision boundary:



$g_1 = g_2$

$g_1 > g_2$

$g_2 > g_1$

R_1

R_2

or in general, decision boundaries are where:

$$g_i(X) = g_j(X)$$

between regions R_i and R_j

Single feature - decision boundary - point

2 features - curve (quadratic for gaussian distribution)

3 features - surface

More than 3 - hypersurface

$$g_i(X) = P(X|\omega_i).P(\omega_i)$$

$$gi(X) = \frac{P(X|\omega_i).P(\omega_i)}{P(X)}$$

Sometimes, it is easier to work with logarithms

$$g_i(X) = \log[P(X|\omega_i).P(\omega_i)]$$

$$g_i(X) = \log P(X|\omega_i) + \log P(\omega_i)$$

Since logarithmic function is a monotonically increasing function, log fn.will give the same result.

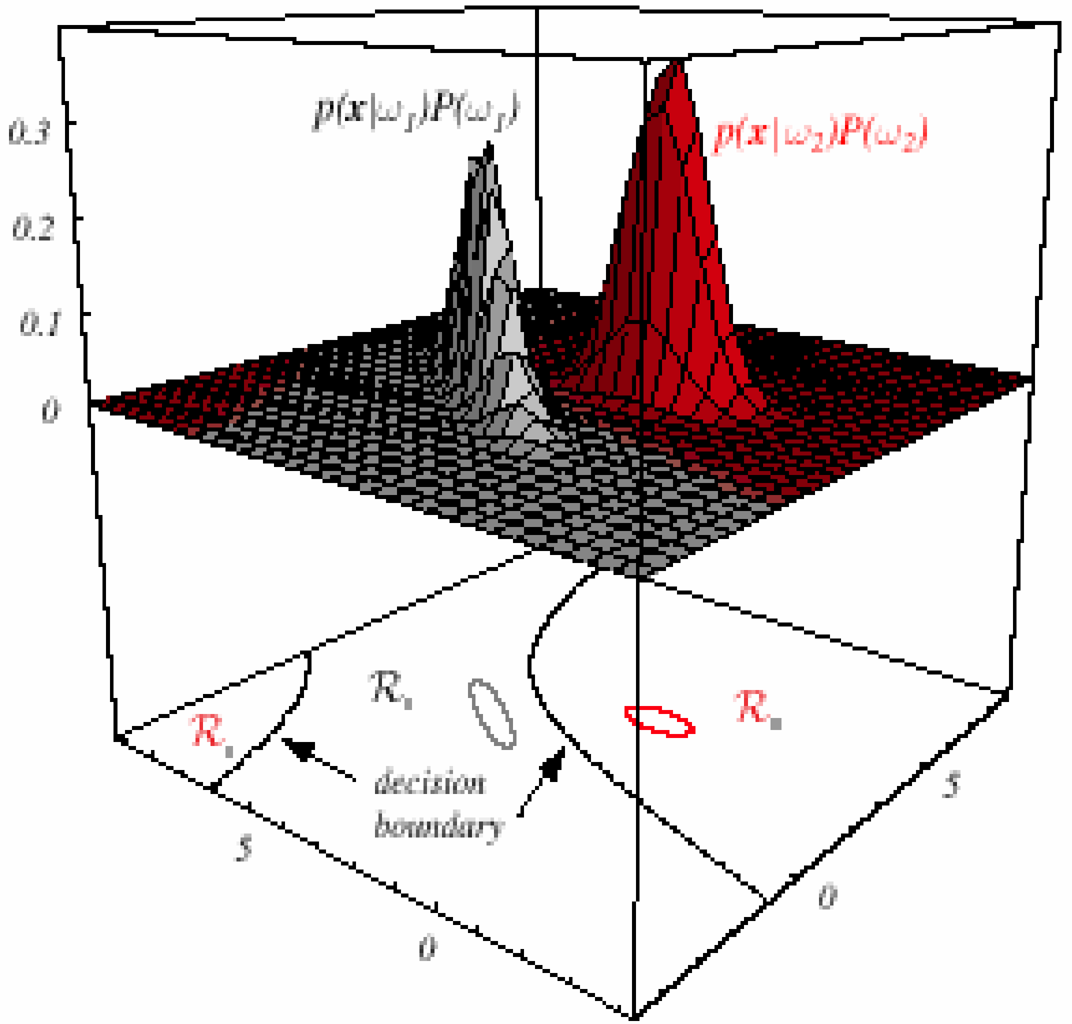


FIGURE 2.6. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region $\mathcal{R}_2$ is not simply connected. The ellipses mark where the density is $1/e$ times that at the peak of the distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayesian Discriminant Functions

For Minimum Error:

$$+\ P(\omega_i | X)$$

$$+\ P(X | \omega_i).P(\omega_i)$$

$$+\ \log P(X | \omega_i) + \log P(\omega_i)$$

For Minimum Risk:

$$-\ R^i(X)$$

Where

$$R^i(X) = \sum_{j=1}^{c} \lambda(\alpha_i | \omega_j).P(\omega_j | X)$$

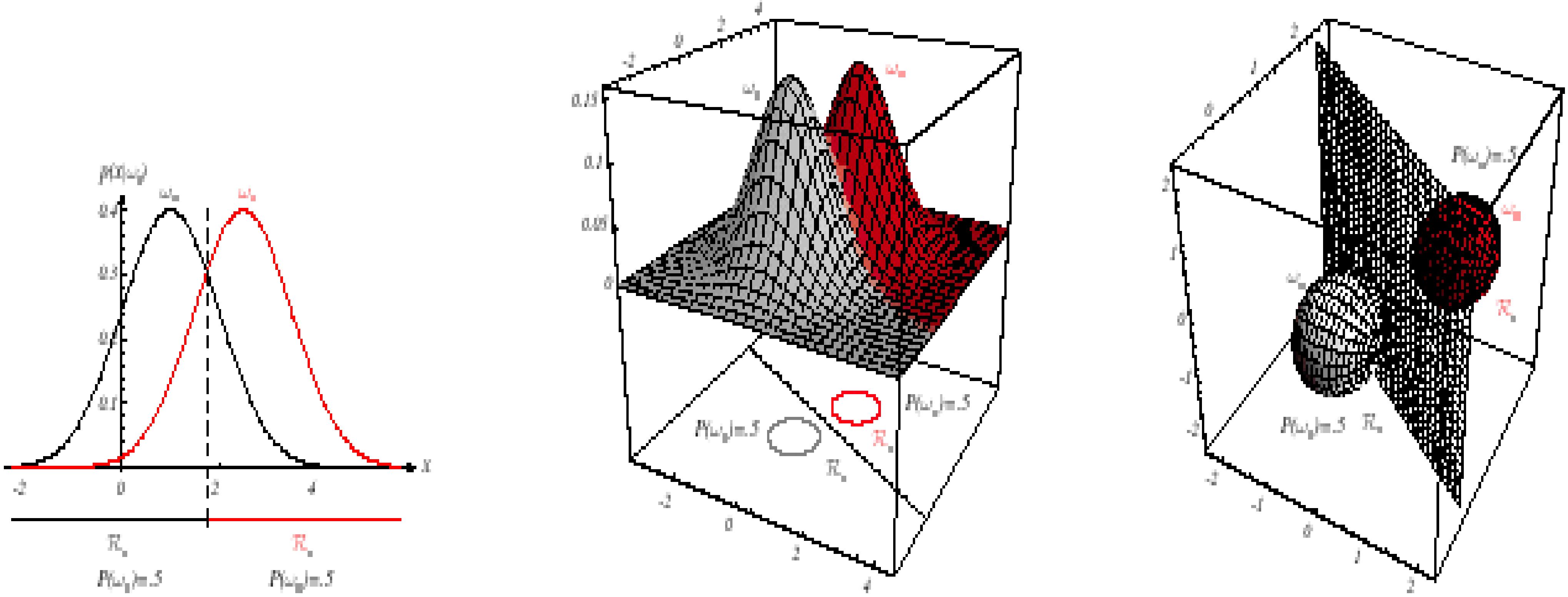


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates $\mathcal{R}_1$ from $\mathcal{R}_2$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayes (Maximum Likelihood) Decision Classifier:

Bayes Classifier can be shown to result with a minimum average error/risk, therefore considered to be optimal

Most general optimal solution

Can be used if the parametric models are known or properly estimated

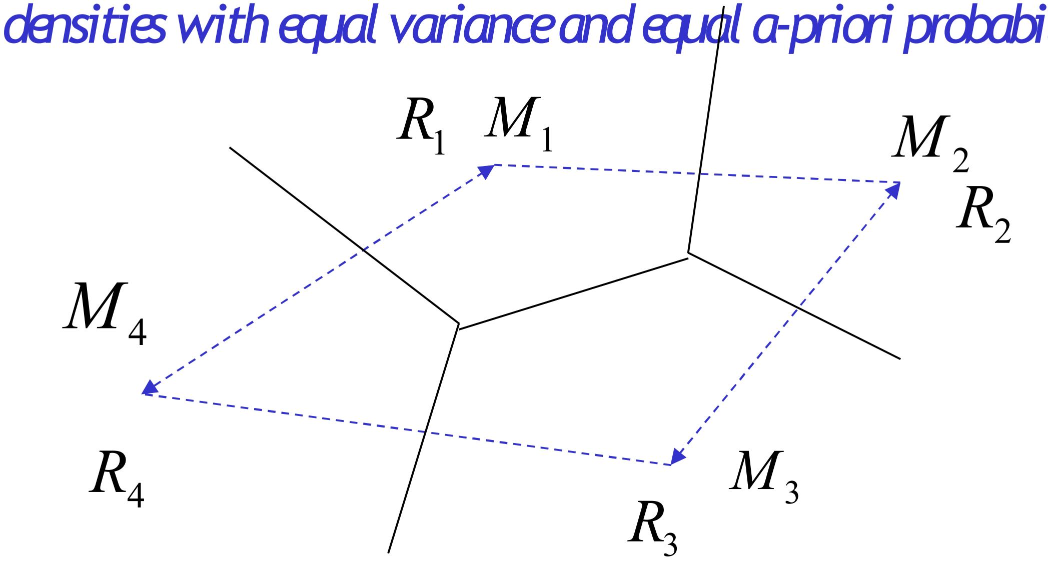
Provides an upper limit(you cannot do better with other rule)

Useful in comparing with other classifiers

Minimum Distance Classifier: A special case of Bayes

Classify an unknown sample X to the category with closest mean !

Optimum when gaussian densities with equal variance and equal a-priori probability.



Piecewise linear boundary in case of more than 2 categories.

Naive Bayes Classifier

Another special case

Assumes the features are statistically independent for a given category

Results with: $P(x1, x2, x3\ldots) = p(x1)p(x2)p(x3)\ldots$

Simplifies the decision rule

Often used in practice

Estimation of Parameters

Bayes Rule is great if you know the class-conditional densities, but not available in nature

If the parametric form of the densities are given or assumed, then, using the labeled samples, the parameters can be estimated. (supervised learning)

Maximum Likelihood Estimation of parameters

Use the sample set X1, X2, X3,.......

Find the parameters that will result with most likely combination

Features and Feature Selection

Curse of dimensionality: High number of features increase the correct classification ratio but require too much data!

We should remove unnecessary features

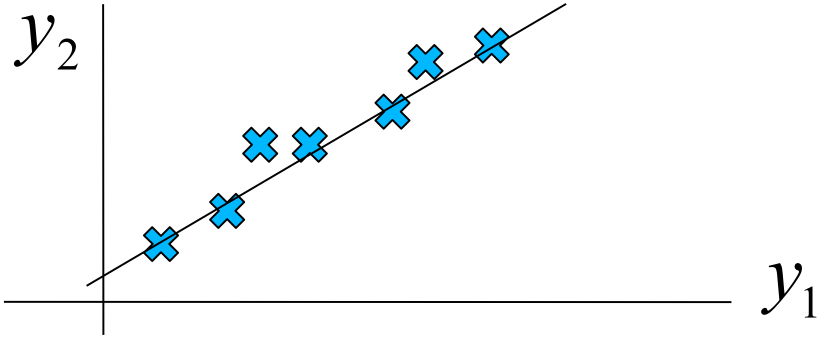
Are all features independent from each other? Can we reduce the size without loosing information, by eliminating redundancies?(Principal Component analysis)

Are the features class discriminating? Again, remove features that have no discriminating ability(Fishers etc)

Eliminating Redundant Features

$X = [x_1, \ldots\ldots\ldots, x_d]^T$ is to be found using a larger set

$Y = [y_1, \ldots\ldots\ldots, y_m]^T$



y_1, y_2

Features that are linearly dependent so they can be reduced to 1

So we either

Throw one away

Generate a new feature using and

(ex:projections of the points to a line)

Form a linear combination of features.

$$\left.\begin{aligned} x_1 &= f_1(y_1,\ldots\ldots,y_m)\\ x_2 &= f_2(y_1,\ldots\ldots,y_m)\\ \\ x_d &= f_d(y_1,\ldots\ldots,y_m)\end{aligned}\right\}\qquad \textit{Linear functions of } y$$

$$X = WY$$

A linear transformation

W? Can be found by:

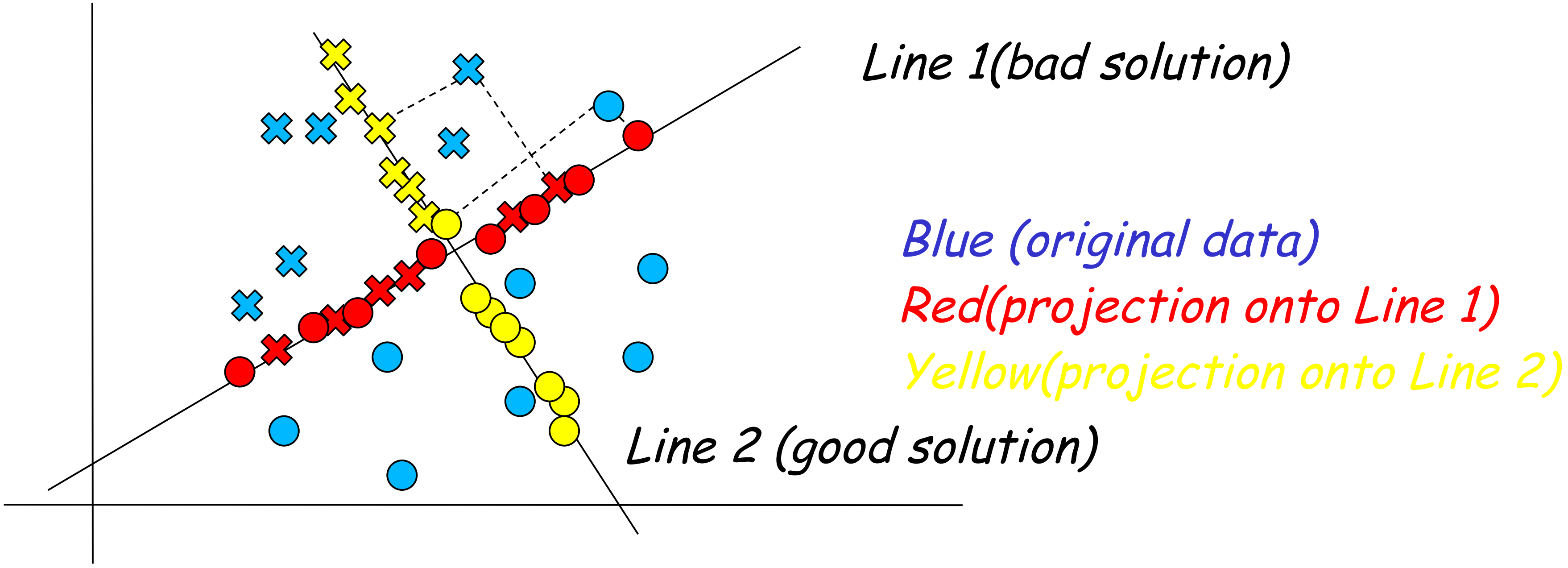
- **K-L expansion, Principal Component Analysis(PCA)**

PCA uses **eigenvalue** approach to result with an ordered set of features,
in order of statistical independence.

So pick first d 'most independent' features

Fisher’s Linear Discriminant

- Curse of dimensionality. More features, more samples needed.
- We would like to choose features with more discriminating ability.
- Reduces the dimension of the problem to one in its simplest form.
- Seperates samples from different categories.
- Consider samples from 2 different categories now.



-Find a line so that the projection separates the samples best. _

Same as: $y = W^T X$

Apply a transformation to samples X to result with a scalar such that Fisher’s criterion function is maximized, where

$$J(W) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

It can be shown that W that maximizes J can be found by solving the eigenvalue problem again:

$$S_W^{-1} S_B W = \lambda W$$

and the solution is given by

$$\boxed{W = S_W^{-1}(M_1 - M_2)}$$

Where SB and Sw are 'scatter matrices' defined as functions of data scattering.

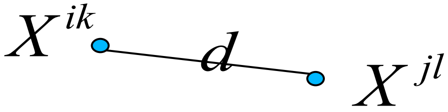
Multiple Discriminant Analysis: c category problem. A generalization of 2-category problem. Generalization to M dimensions is also possible.

k-Nearest Neighbor (k-NN) Rule

Non-parametric classification rules:
Linear and generalized discriminant functions
Nearest Neighbor & k-NN rules

Nearest Neighbor Classification Rule
1-NN: A direct classification using learning samples
Assume we have learning samples from different categories

$$X^{11}, X^{12}, \ldots\ldots, X^{jk}, \ldots\ldots\ldots\ldots$$



Assume a distance measure between samples such as euclidian

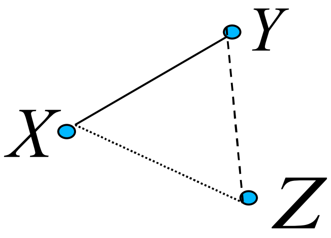
$$d(X^{ik}, X^{jl})$$

A general distance metric should obey the following rules:

$$d(X^{ij}, X^{ij}) = 0$$

$$d(X^{ij}, X^{jl}) = d(X^{jl}, X^{ij})$$

$$d(X,Y) \leq d(X,Z) + d(Z,Y)$$



Most standard: Euclidian Distance

$$d(X,Y) = \|X - Y\| = \left[\sum_{i=1}^{n} (x_i - y_i)^2\right]^{1/2} = \left[(X - Y)^T (X - Y)\right]^{1/2}$$

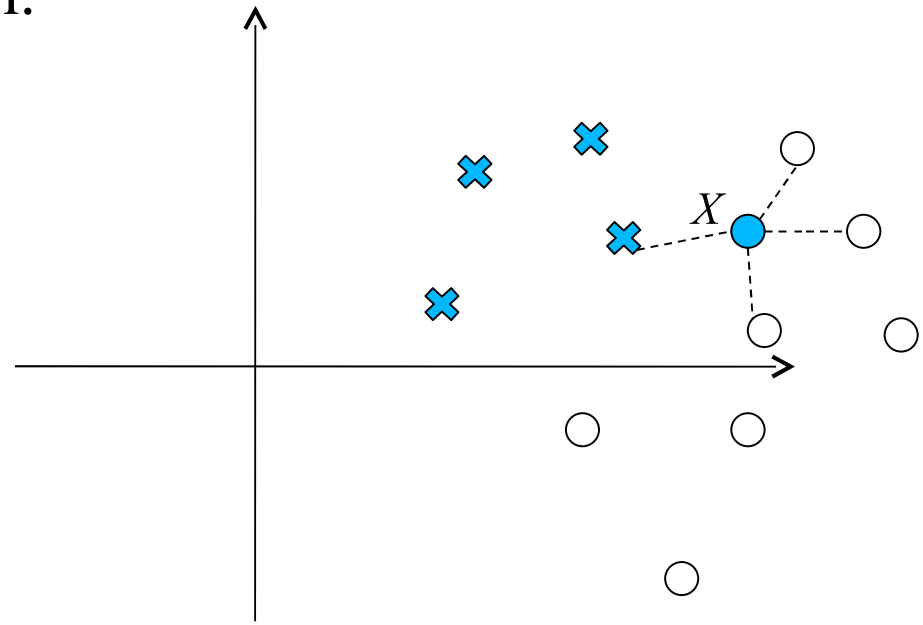
1-NN Rule: Given an unknown sample X

α_i if

$$d(X, X^{ik}) < d(X, X^{jl})$$

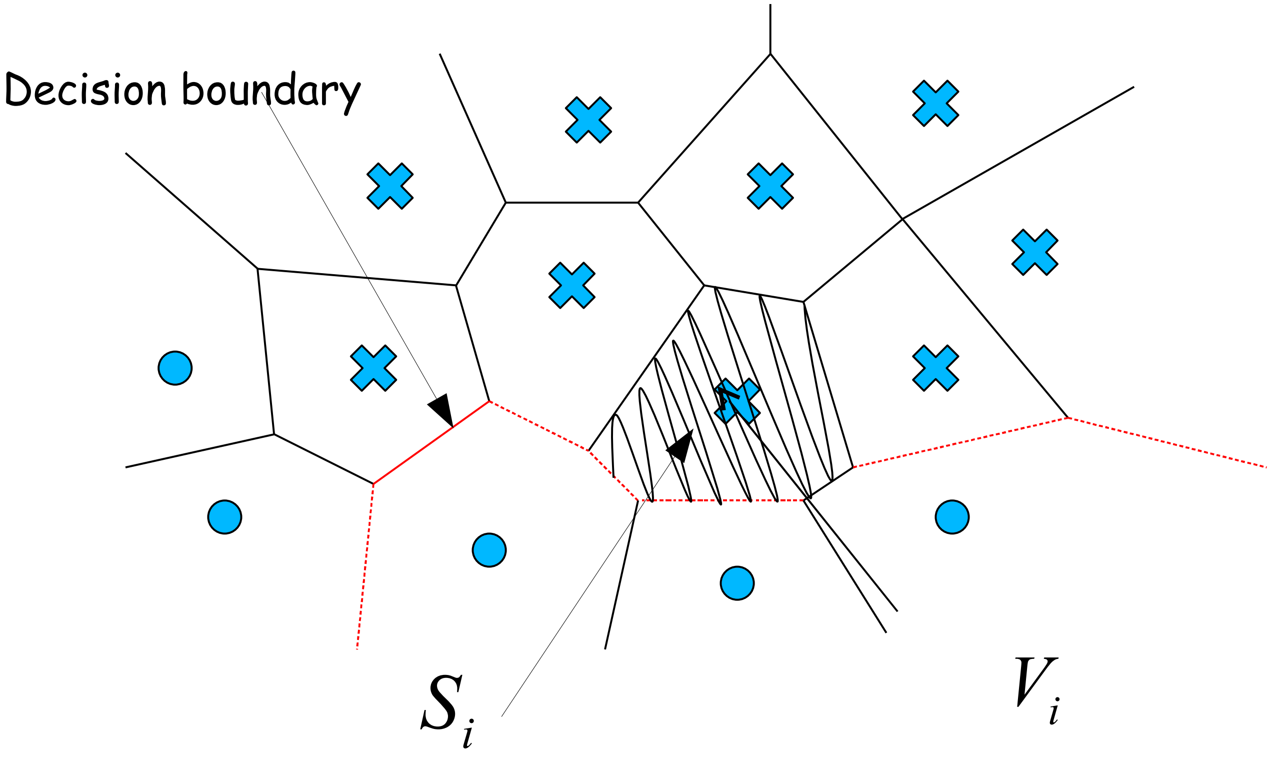
For $jl \neq ik$

That is, assign X to category ω_i if the closest neighbor of X is from category i.



Results with piecewise linear decision boundaries.

Voronoi Diagrams



V_i is a polygon such that any point that falls in V_i is closer to sample S_i than any other sample S_j .

k-NN rule: instead of looking at the closest sample, we look at k nearest neighbors to X and we take a vote. The largest vote wins. k is usually taken as an odd number so that no ties occur.

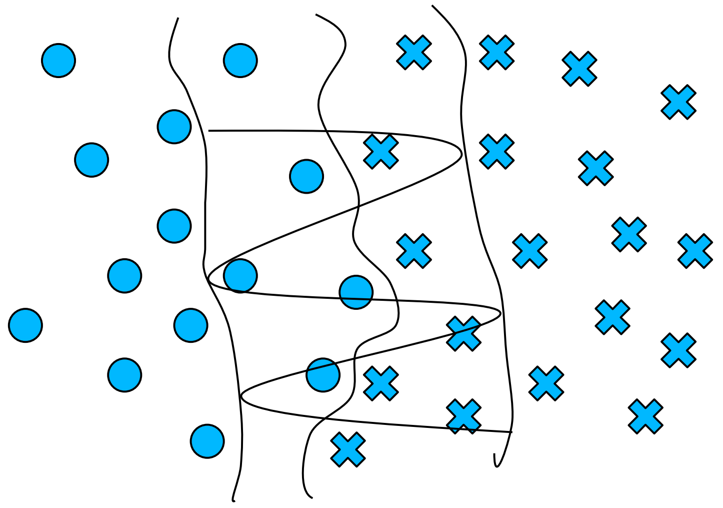
Analysis of NN rule is possible when $M \to \infty$ and it was shown that it is no worse than twice of the <u>minimum-error classification (in error rate).</u>

EDITING AND CONDENSING

NN rule becomes very attractive because of its simplicity and yet good performance.

So, it becomes important to reduce the computational costs involved.

Do an intelligent elimination of the samples.



Remove samples that do not contribute to the decision boundary.

Linear Discriminant Functions

Assume the discriminant functions are linear functions of X

$$g(X) = w_1 x_1 + w_2 x_2 + \ldots . + w_n x_n + w_o$$

$$= W^T X + w_0$$

$$W = [w_1 \ldots\ldots . w_n]^T$$

$$X = [x_1 \ldots\ldots\ldots . x_n]^T$$

We may also write g in more compact form

$$\boxed{g(X) = W_a^{\ T} X_a = W_a^{\ T} Y}$$

$$Y = X_a = [x_1 \ldots . . x_n 1]$$

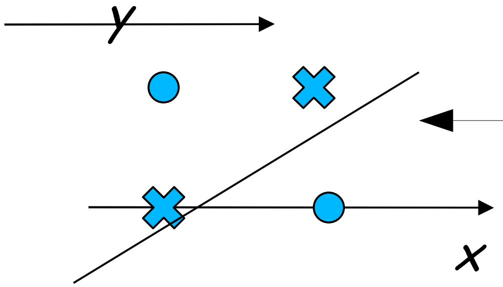
$$W_a = [w_1 w_2 \ldots\ldots w_n w_0]^T$$

Linear Discriminant Functions

- Its assumed that the discriminant functions are linear (boundaries are linear)
- The labeled learning samples are used to find best boundaries.

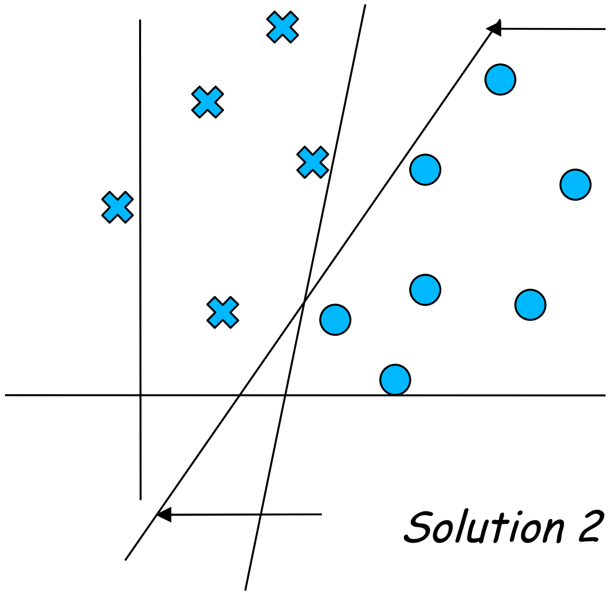
- Finding the g is the same as finding Wa.
- How do we define the 'best'?
- All learning samples are classified correctly?
- Does a solution exist?

Linear Separability



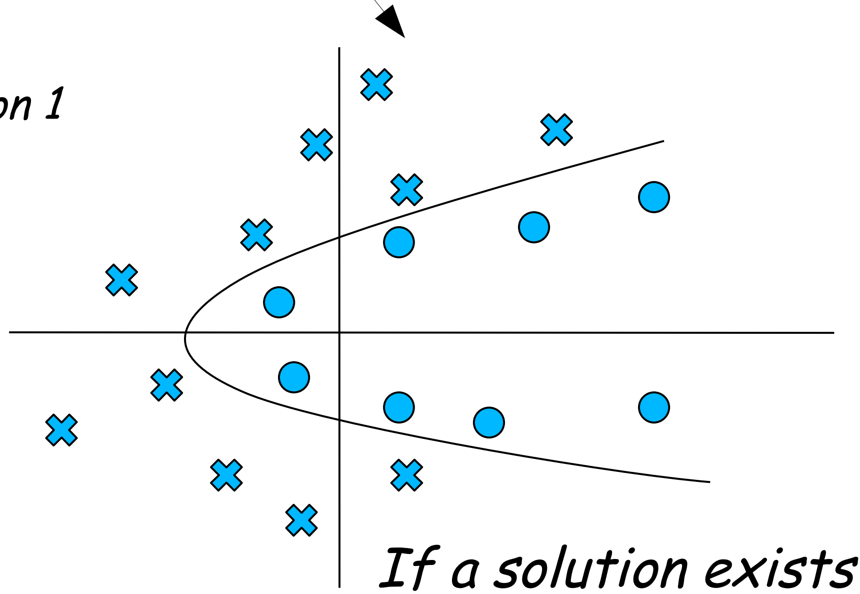
XOR Problem

Not linearly separable



Solution 1

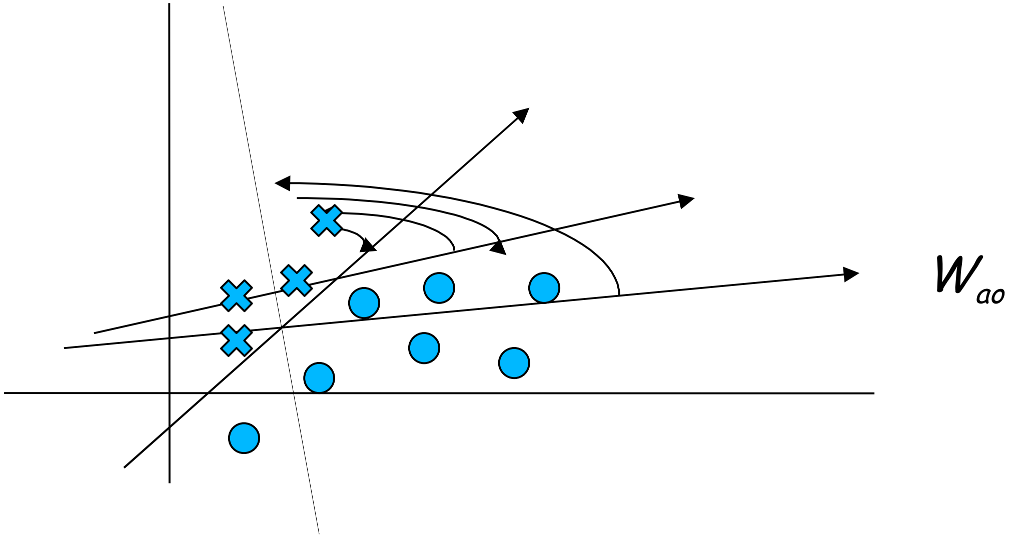
Solution 2



Seperable,many solutionsions possible

If a solution exists-the problem is called "linearly separable" and W_a is found iteratively. Otherwise "not linearly separable" piecewise or higher degree solutions are seeked.

Iterative Solution: start with an initial estimate and update it until a solution is found.

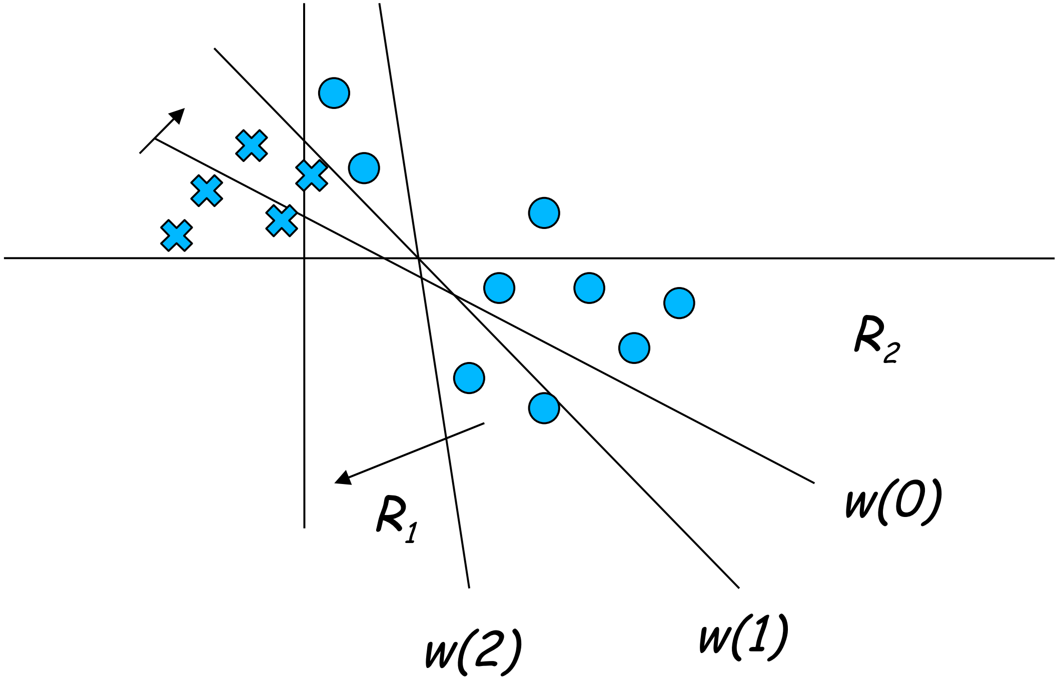


- **Gradient Descent Procedures**
- **Perceptron Criterion function**
- **One-layer Perceptron(Rosenblatt)**

Assume linear separability.

perceptron learning

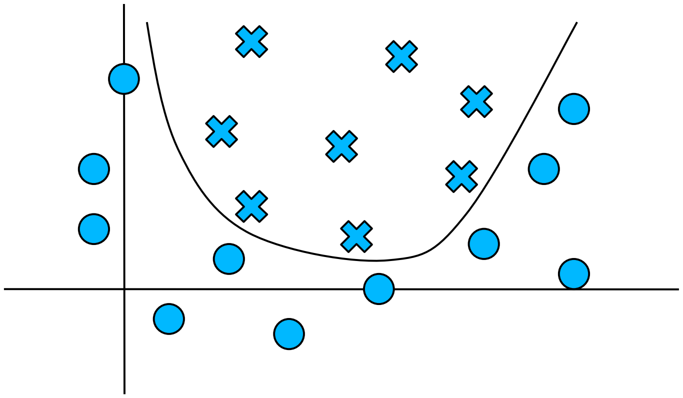
An iterative algorithm that starts with an initial weight vector and moves it back and forth until a solution is found.



Update of the weights are done using misclassified samples, towards reducing them

Generalized Discriminant Functions

When we have nonlinear problems as below:



Then we seek for a higher degree boundary.

<u>Ex:</u> quadratic boundary

$$g(X) = \sum \sum w_{ij} x_i x_j + \sum w_i x_i + w_0$$

will generate hyperquadratics boundaries.

g(X) still a linear function w's. $g(Y_a) = W_a^T Y_a$

–

NON-SEPARABLE CASE-What to do?

It was shown that we can increase the feature space dimension with a nonlinear transformation, the results are linearly separable. Then find an optimum solution.(Support Vector Machines)

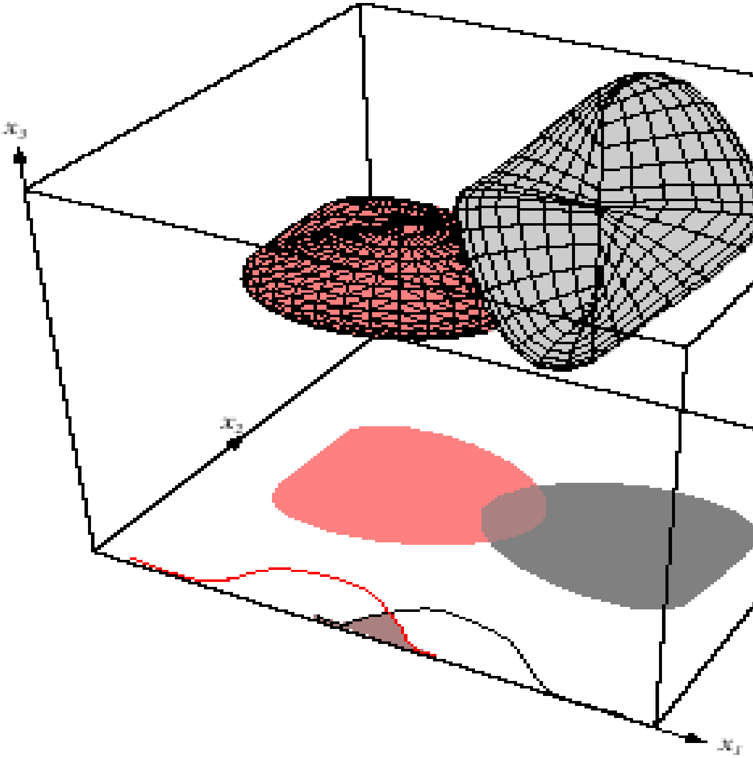


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

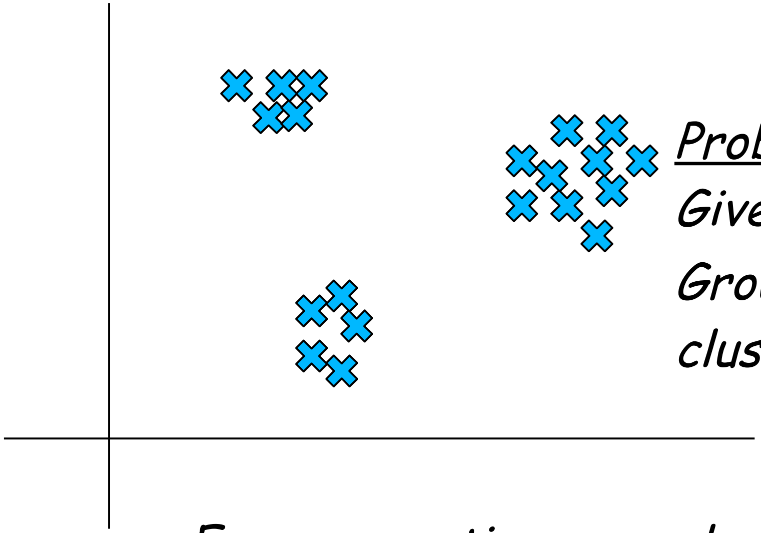
UNSUPERVISED LEARNING AND CLUSTERING

No class labels for learning samples.

We need additional means to label and classify – can be done separately (first label then classify) or together.

PARAMETRIC APPROACH- Estimation of class conditional densities

NONPARAMETRIC – CLUSTERING

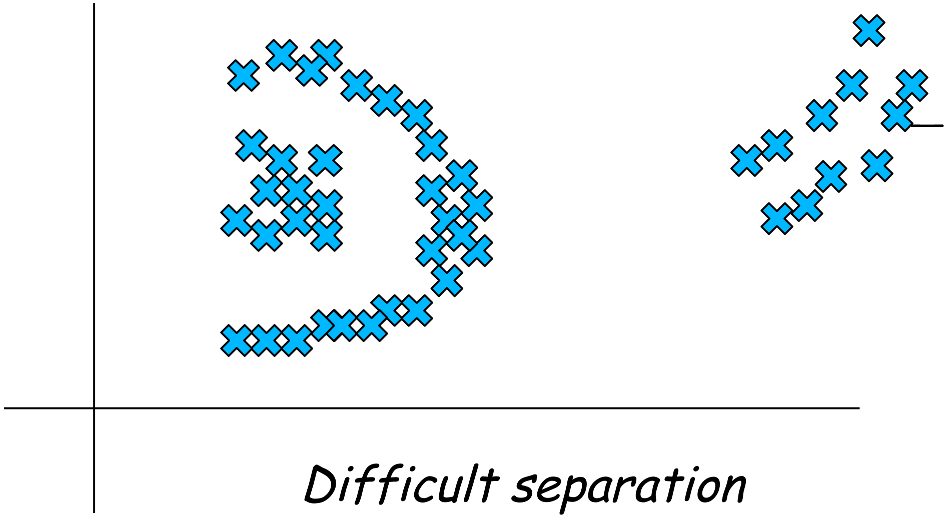


Problem:

Given samples $X_1, X_2, \ldots\ldots\ldots\ldots, X_n$

Group them into clusters so that samples in same cluster are "similar"

Easy separation example



Difficult separation

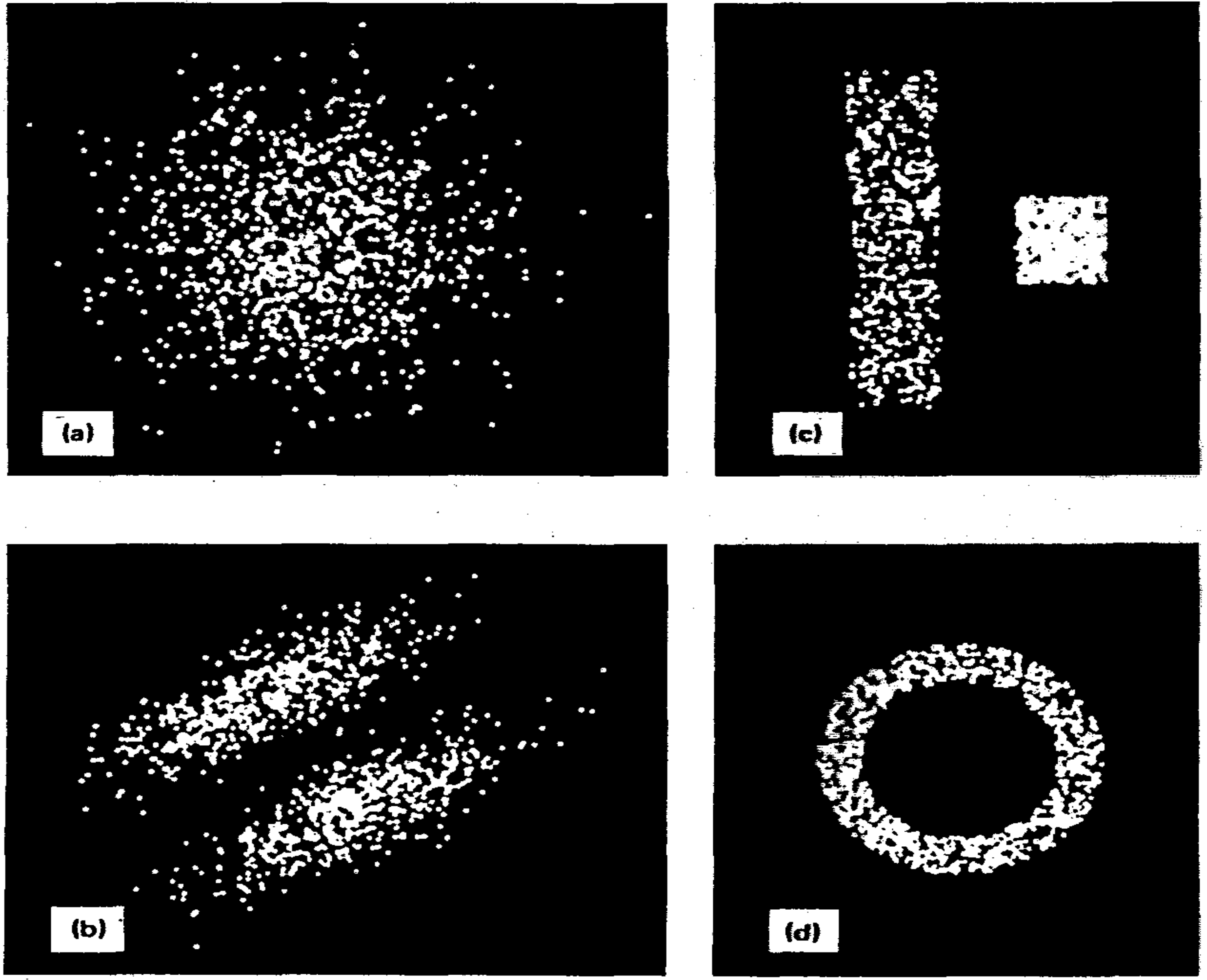


FIGURE 6.7. Data sets having identical second-order statistics.

Scaling Effects

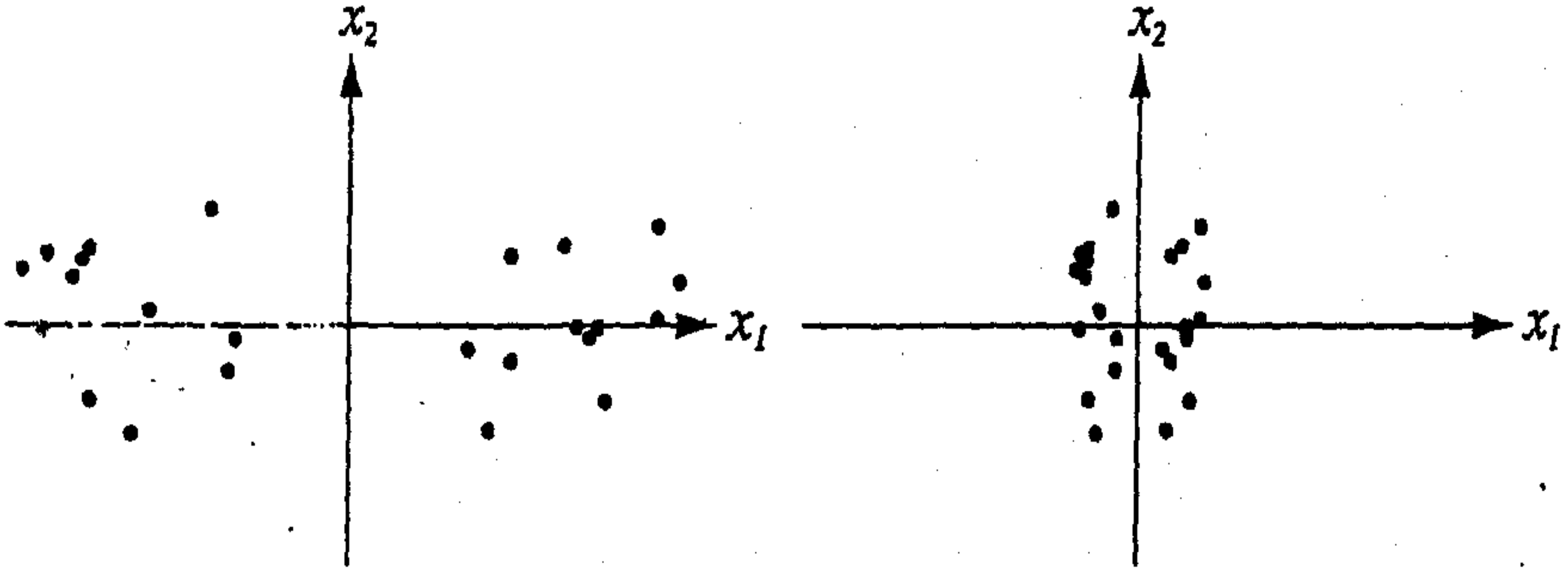


FIGURE 10.9. If the data fall into well-separated clusters (left), normalization by scaling for unit variance for the full data may reduce the separation, and hence be undesirable (right). Such a normalization may in fact be appropriate if the full data set arises from a single fundamental process (with noise), but inappropriate if there are several different processes, as shown here.

Criterion Functions for Clustering

$$J_e = \sum_{i=1}^{c} \left(\sum_{x \in C_i} \left\| x - M_i \right\|^2 \right) = \frac{1}{2} \sum_{i=1}^{c} n_i \bar{S}_i$$

$$\bar{s}_i = \frac{1}{n_i^2} \sum_{x \in C_i} \sum_{x' \in C_i} \left\| x - x' \right\|^2$$

c- number of clusters

M_i – mean of the samples in the same cluster

<u>Aim: </u>determine the partition that will minimize J.

Minimum variance partition (sum of squared error criterion)

Sum of squared error criterion

$$M_i = \frac{1}{n_i} \sum_{x \in D_i} x$$ *Sample mean for cluster D_i*

Sum of squared errors:

$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \left\| x - M_i \right\|^2$$

Using J_e results well for compact clusters.

Basic ISODATA Algorithm (k-means)

Assume that there are k categories

- Choose k arbitrary points in space as cluster centers.

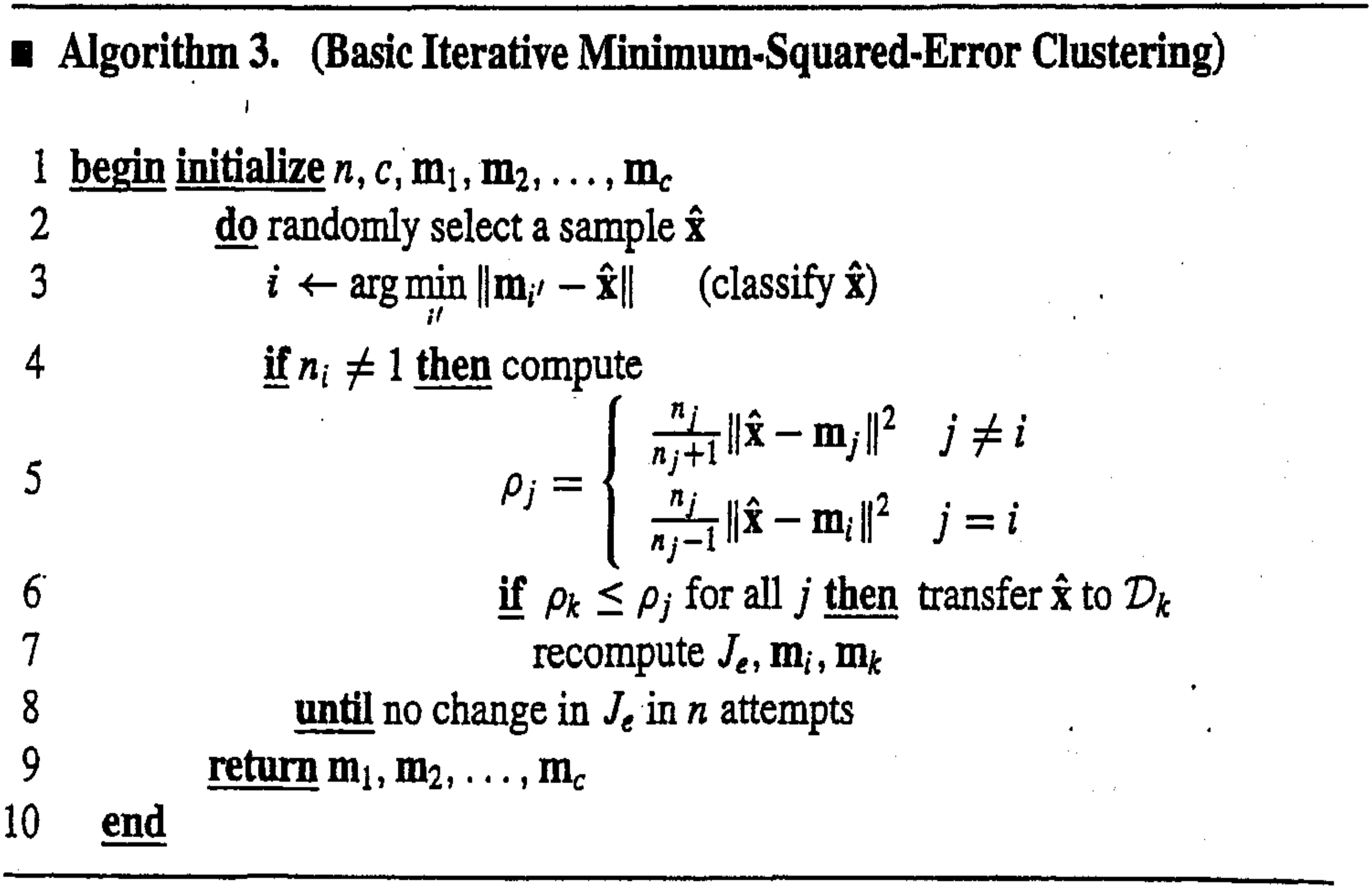
$$M_{10}, M_{20}, \ldots\ldots\ldots\ldots, M_{k0}$$

- Assign samples to their nearest cluster.

- Update M's. If any means changed value, go to 2. Otherwise stop.

May fall into local minimum.

Iterative Minimum Squared Error Clustering



Algorithm 3. (Basic Iterative Minimum-Squared-Error Clustering)

```
begin initialize $n, c, \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_c$
    do randomly select a sample $\hat{\mathbf{x}}$
        $i \leftarrow \arg\min_{i'} \|\mathbf{m}_{i'} - \hat{\mathbf{x}}\|$    (classify $\hat{\mathbf{x}}$)
        if $n_i \neq 1$ then compute
            $\rho_j = \begin{cases} \frac{n_j}{n_j+1}\|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 & j \neq i \\ \frac{n_j}{n_j-1}\|\hat{\mathbf{x}} - \mathbf{m}_i\|^2 & j = i \end{cases}$
            if $\rho_k \leq \rho_j$ for all $j$ then transfer $\hat{\mathbf{x}}$ to $\mathcal{D}_k$
                recompute $J_e, \mathbf{m}_i, \mathbf{m}_k$
    until no change in $J_e$ in $n$ attempts
    return $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_c$
end
```

Hierarchical Clustering

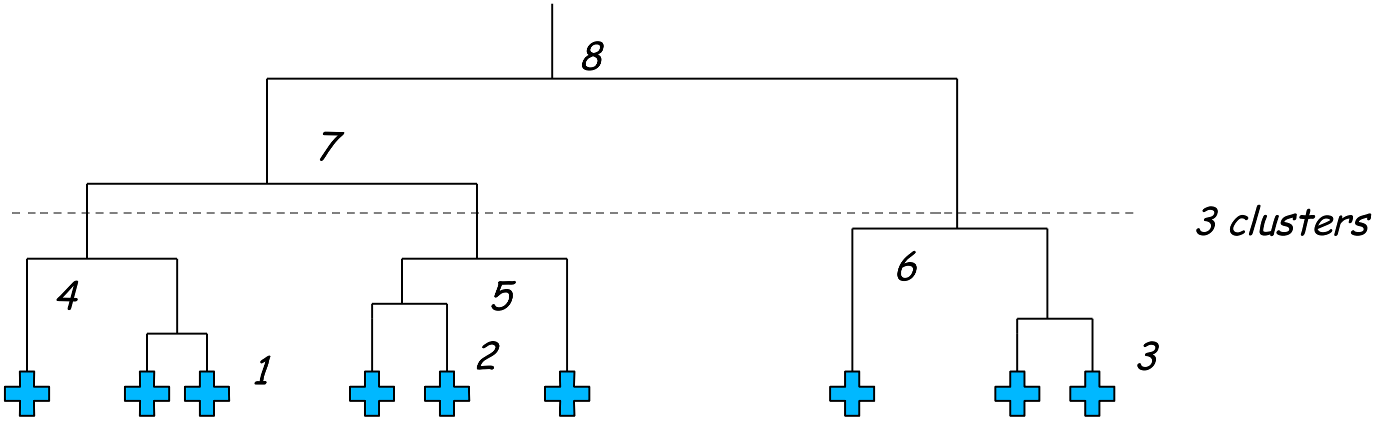
A different approach to clustering.

Hierarchy in living species

Each species is a class by itself.

Combine the ones that are closest

Continue combining until the number of clusters are what is desired or a criterion is satisfied.



Issues:

How do we measure distance between clusters?

When do we stop?

Do we start from bottom or from top?

Basic Bottom-up Hierarchical Clustering Algorithm for k clusters

1. Start taking each sample as a cluster. n=m (n of samples)

2. Measure $d_{i,j}$ – distance between clusters D_i and D_j. Join two clusters D_i and D_k for which $d_{i,k} = \min d_{i,j}$

n=n-1;

3. If n<k stop. k:number of desired clusters

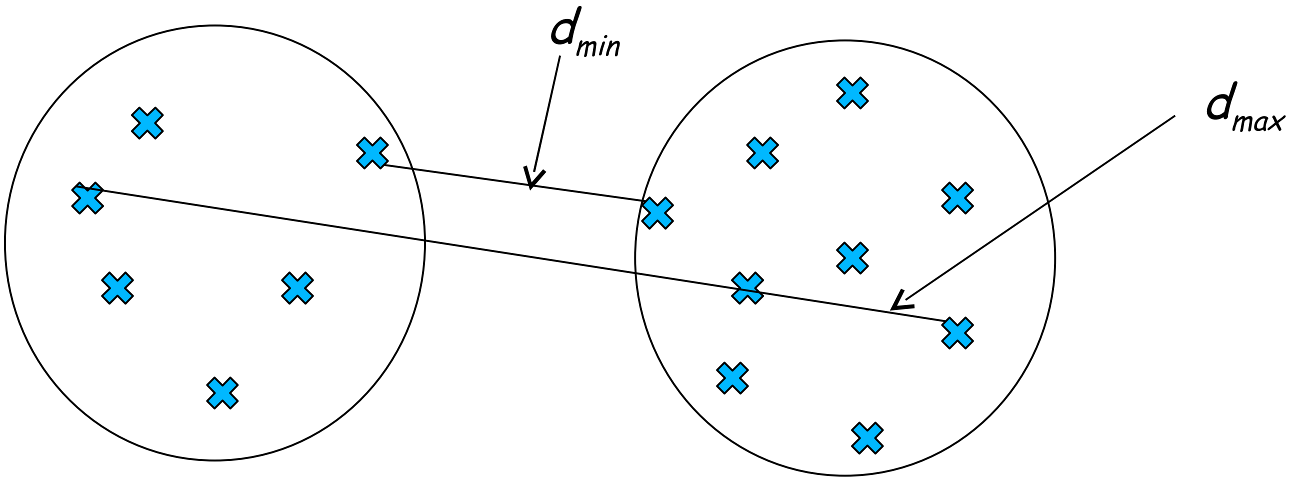
Else go to 2.

d_{ij} can be defined in many ways.(where s is in i and z is in j:)

$$d_{\min_{i,j}} = \min \|x_s - x_z\|$$

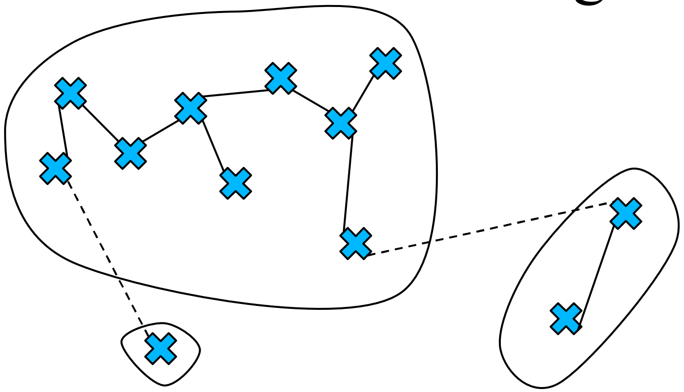
$$d_{avg_{i,j}} = \frac{1}{n_i n_j}\left(\sum \sum \|x_s - x_z\|\right)$$

$$d_{\max} = \max \|x_s - x_z\|$$



Example

Apply hierarchical clustering with d_{min} to below data where c=3.



will form elongated clusters!

Nearest Neighbor Clustering

Tree Classifiers

Consider the feature vector X = (x1, x2, x3....xn)

A tree classifier considers features one by one instead of as a whole and measures them one by one, following the leaves of a tree. The features are usually binary valued .

An optimum tree can be constructed using learning samples.

Leaves of the tree correspond to the classes.

Example will be seen in the following .

Tree Classifiers: Example

Classification of illness 'Osteoarthritis' to levels 0–4

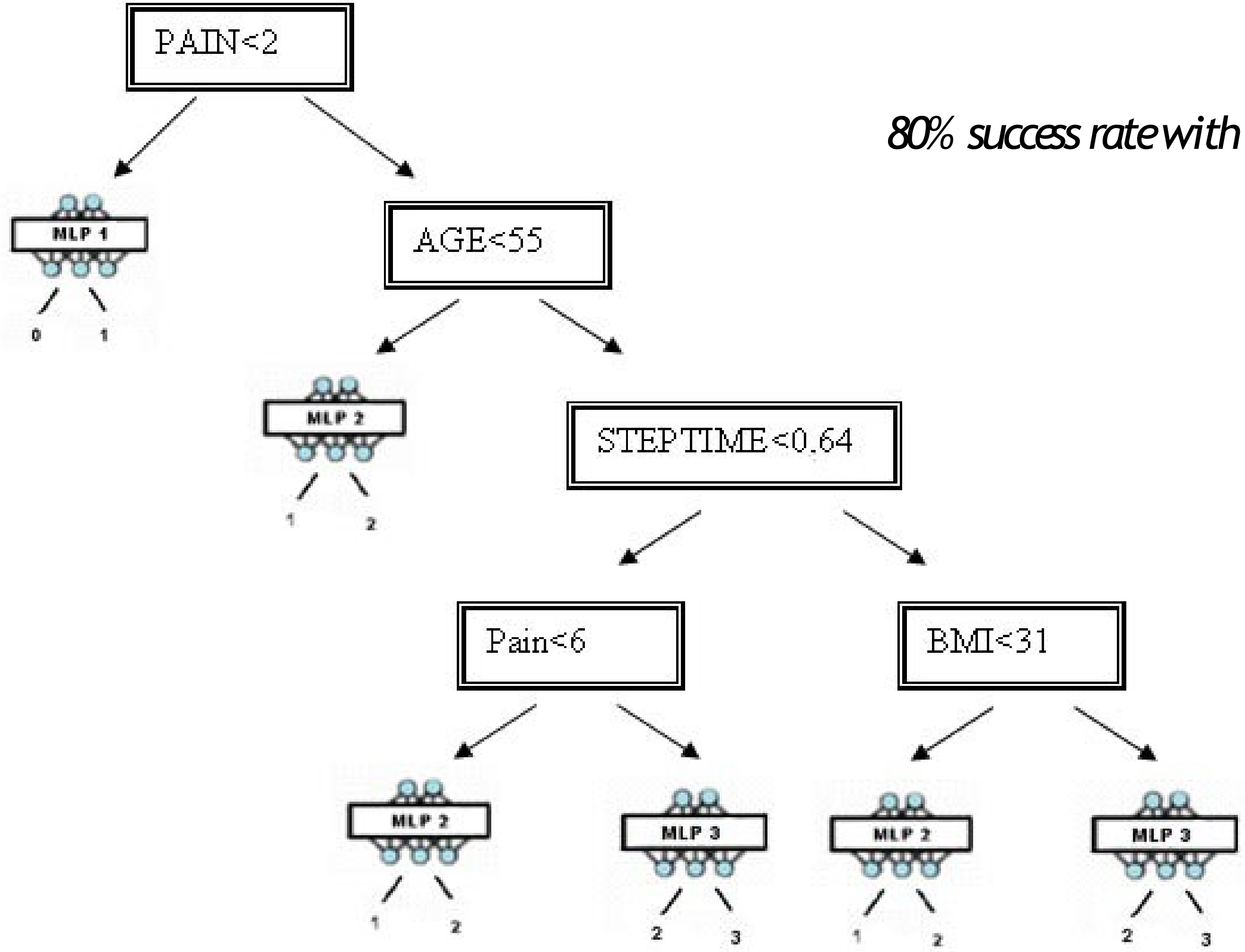
Using features obtained by gait (walking) analysis and patient history

The tree is generated by using 'gini' impurity index

'OAGAIT': A Decision Support System for Grading Knee Osteoarthritis using Gait Data' Pattern Recogniton Letters, July 2010

N. Köktaş, N. Yalabık, G. Yavuzer, P. Dunn, V. Atalay

Implementation and results



80% success rate with 100 test samples

Decision Tree Construction as Learning

Binary Tree (All trees can be converted to a binary tree)

'impurity' measure used to decide how to split a tree (which feature to start with); shows how samples are distributed to categories as a result of split

Split the tree so that impurity is lowest

'entropy impurity','variance impurity', 'Gini impurity'

Training and Performance Testing in PR

Whichever classifier is used, there is usually a training (learning) stage

How to train with available data?

Validation and Cross-Validation

How to test the performance?

Confusion Matrices

ROC Curves

Cross-Validation

Assume a set of labeled samples are available

A number of them will be used to train the classifier and others to test the results(do not use the same samples)

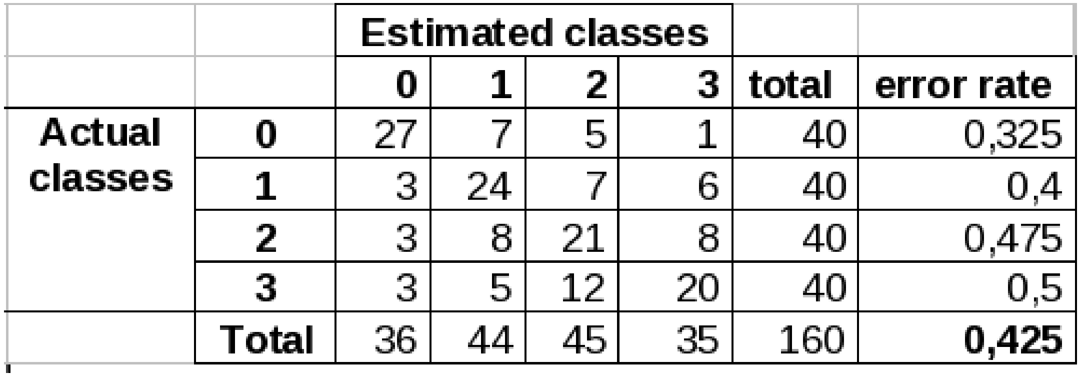
M-fold cross-validation:

- Divide the sample set into m disjoint sets of equal size
- Train m times, each time with a different set used in testing
- The performance is measured as the mean of errors each time

Performance Testing

Simplest method: Confusion Matrices

Shows which category is confused with which



		Estimated classes					
		0	1	2	3	total	error rate
Actual classes	0	27	7	5	1	40	0,325
	1	3	24	7	6	40	0,4
	2	3	8	21	8	40	0,475
	3	3	5	12	20	40	0,5
	Total	36	44	45	35	160	**0,425**

ROC (Receiver Operating Characteristic) Curves

Assume 2 categories, where our aim is to detect a single object against all others (a binary classifier)

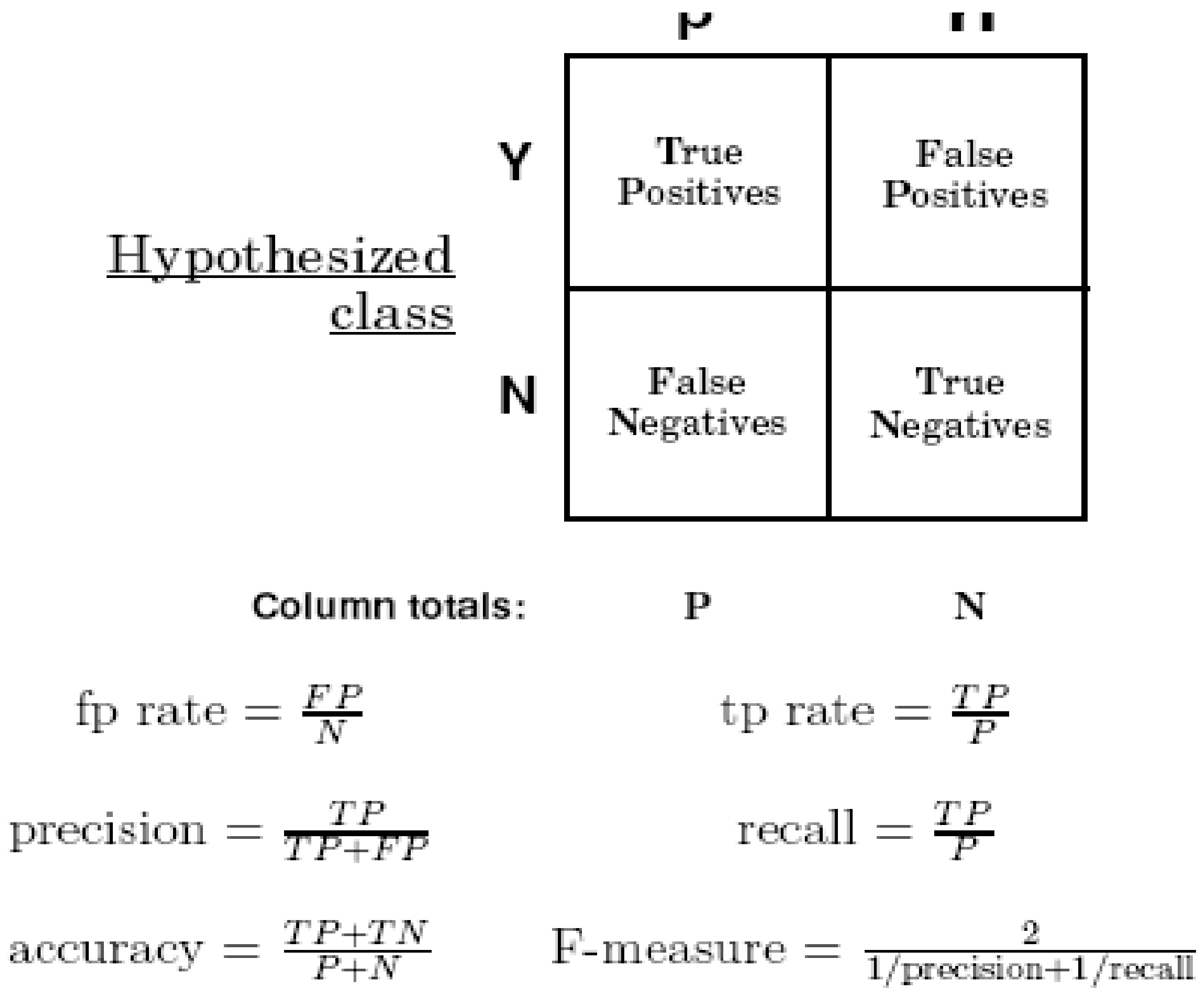
'hit' correctly classifying the object (true positive)

'false alarm' incorrectly finding that there is an object when it is not there (false positive)

'miss' finding no object when it is there (false negative)

Probability of 'hit' vs the probabilty of 'false alarm' is called a ROC Curve.

A ROC Curve is usually used to compare the classifiers.



		p	n
Hypothesized class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N} \qquad \text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

Measures of Performance

Sensitivity (recall rate)

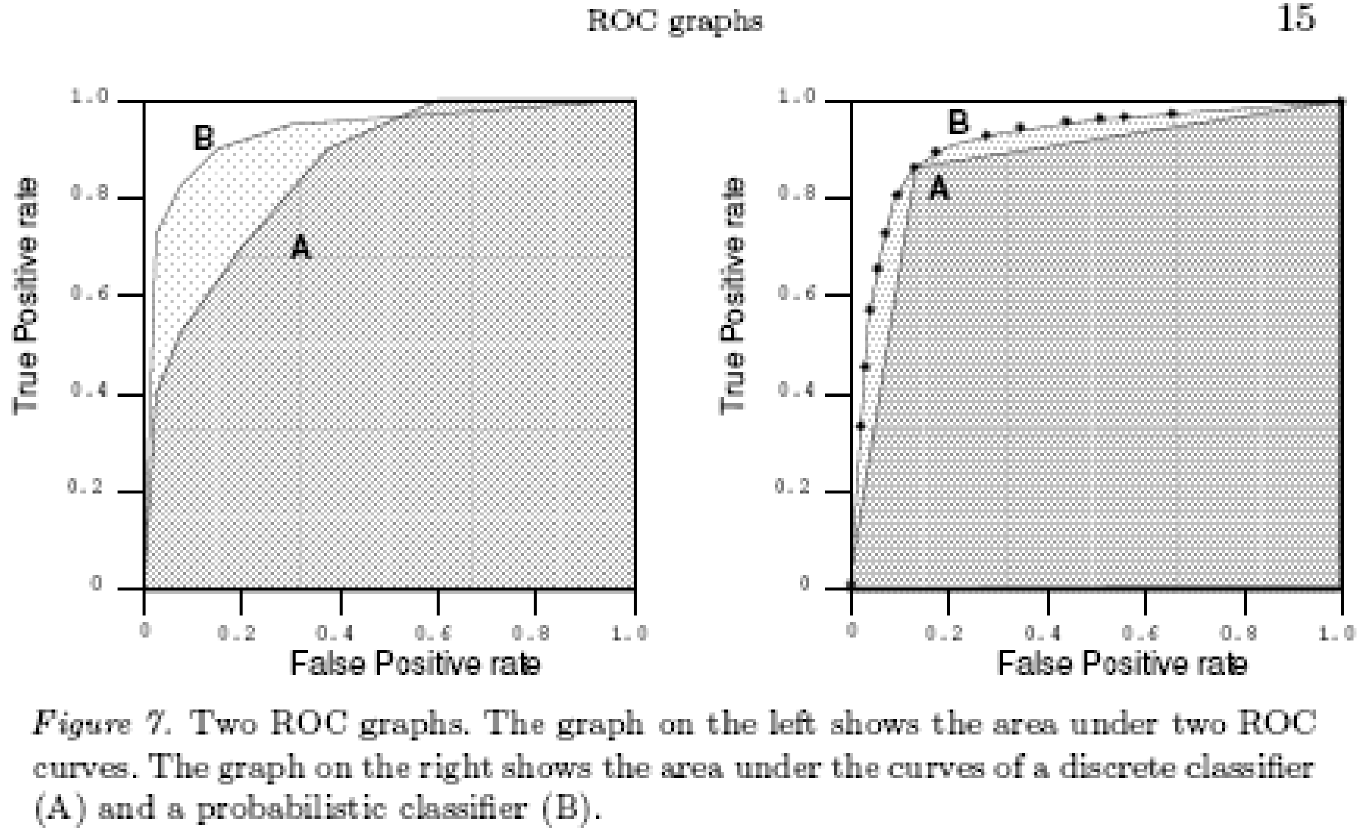
$$\text{sensitivity} = \frac{\text{number of True Positives}}{\text{number of True Positives} + \text{number of False Negatives}}$$

Specifity

$$\text{specificity} = \frac{\text{number of True Negatives}}{\text{number of True Negatives} + \text{number of False Positives}}$$

Examples of ROC Curves

(Taken from :Tom Fawcett 'ROC Graphs: Notes and Practical Considerations for researchers')



ROC graphs 15

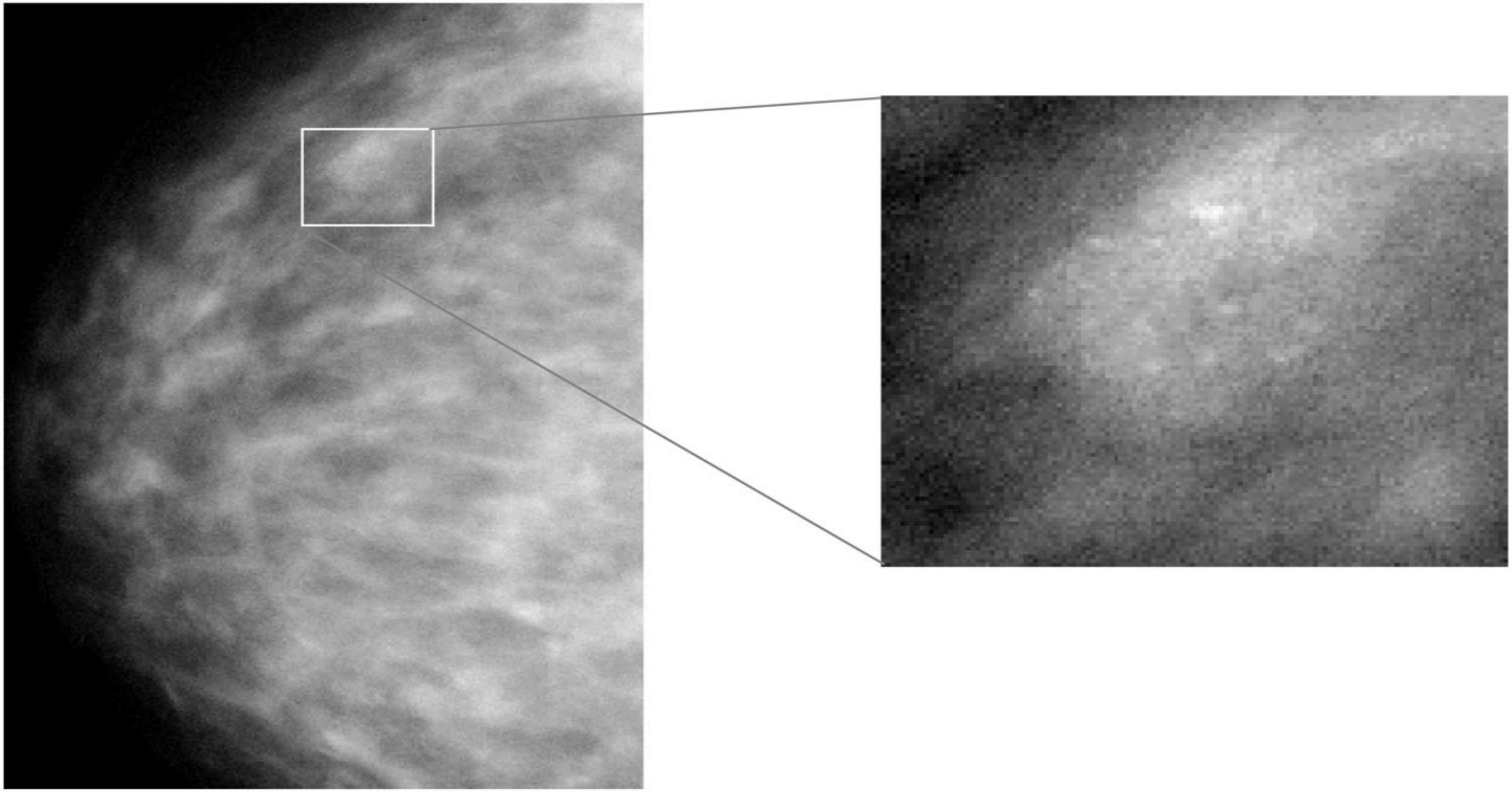
Figure 7. Two ROC graphs. The graph on the left shows the area under two ROC curves. The graph on the right shows the area under the curves of a discrete classifier (A) and a probabilistic classifier (B).

Example: Microcalcifications in a Mammogram

'A Support Vector Machine Approach for Detection of Microcalcifications'

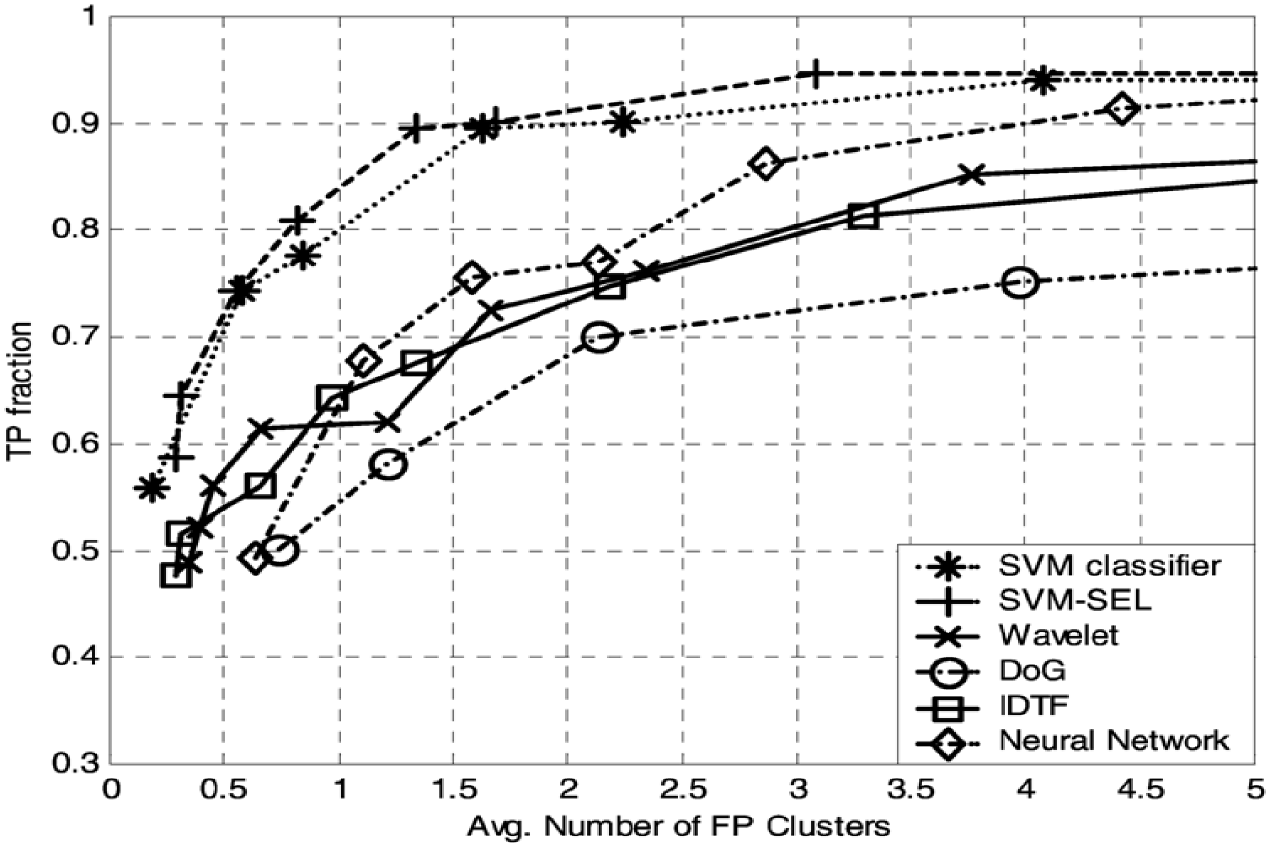
Issam El-Naqa et al

IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 21, NO. 12, DECEMBER 2002



Performance Comparison using a ROC curve

Higher the curve is, better the performance



References

'*Pattern Classification*' R. Duda, P. Hart, D. Stork 2nd ed. Wiley 2001

'*Pattern Recognition*' S.Theodoridis, K.Koutroumbas" , Elsevier, 2003

Min720 Lecture Notes, N. Yalabık, ODTÜ 2010

http://home.comcast.net/~tom.fawcett/public_html/pap ROC Curves)

Others in respective pages