ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

Τάσος ΠΟΥΛΙΕΖΟΣ
Καθηγητής Συστημάτων Αυτομάτου Ελέγχου
Τμήμα Μηχανικών Παραγωγής και Διοίκησης
Πολυτεχνείο Κρήτης

Το επιστημονικό μου πλαίσιο:

BSc. Mathematics and Computing, Polytechnic of North London (νυν London Metropolitan University)

MSc. Control Systems, Imperial College

Phd. Control Systems, Brunel University

Τι είναι η «αναγνώριση προτύπων» ;

- *Απλά*: ασχολείται με την απάντηση στο ερώτημα: «τι είναι αυτό;» - Morse
- *Μαθηματικά*: η εκτίμηση συναρτήσεων πυκνότητας σε χώρο πολλών διαστάσεων και η διαίρεση του χώρου σε περιοχές κλάσεων – Fukunaga
- *Λογοτεχνικά*: αποκάλυψη δομής στο χάος

- **Απαραίτητο συστατικό της μηχανικής ευφυΐας**

Είναι σημαντικό;

TU Delft

PR Sys Design

Pattern Recognition course for Industry

2 March, 2012

Dear researcher,

are you working on practical classification problems but miss deeper understanding of advanced statistical techniques to analyze and learn from your data?

Our 5-days Advanced Pattern Recognition Course will help you to extend your competence and practical capabilities.

Advanced Pattern Recognition Course for Industrial Applications
23-27 April 2012

The Delft Pattern Recognition research at Delft Technical University in collaboration with PR Sys Design is happy to announce the Pattern Recognition Course for Industry at TUDelft, The Netherlands.

This course offers our broad experience and expertise in the research of applied and fundamental pattern recognition. It provides a balance between theory (lectures) and practical exercises. It has constantly been refreshed, improved and updated over the last nine years. The course is based on the following Matlab Toolboxes: PRTools and DD_Tools (developed at TUDelft) and perClass (developed by PR Sys Design). These toolboxes provide a great flexibility and can be adopted in industrial circumstances, as many of our participants have already done.

Professionals from all over the world participated in the APR course during the past ten years. Their various backgrounds (e.g. mining, industrial inspection, agriculture, medical diagnosis, education and governmental research) highly stimulated interesting discussions during the lectures and exercises.
For more information see the website or send an email message to prcourse@prtools.org

We have contacted you due to your interest in PRTools. If you do not wish to receive further communications from us, please, click on the unsubscribe link below. We won't contact you in the future.

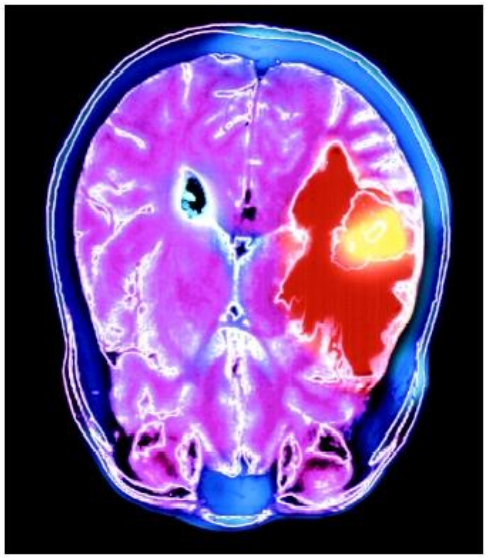
Sincerely,

Carmen Lai
APR Course Organization
--
Dr Carmen Lai, PR Sys Design
http://perclass.com

Copyright © 2012 PR Sys Design, All rights reserved.
You are receiving this email because you have downloaded PRTools
Our mailing address is:
PR Sys Design, Molengraaffsingel 12,
2629JD Delft, The Netherlands

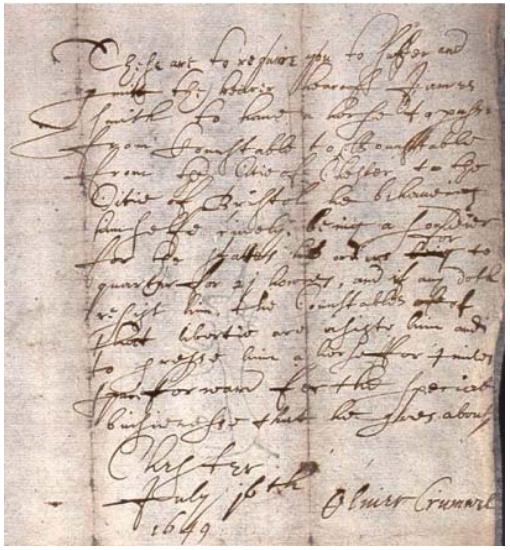
ΕΙΣΑΓΩΓΗ

π.χ.



MRI εγκεφάλου
(κίτρινο: όγκος)

Ιατρική διάγνωση



Χειρόγραφο κείμενο

Διαλογή φακέλων, OCR

Ταυτοποίηση
δακτυλικών
αποτυπωμάτων

Υπηρεσίες ασφάλειας

ΕΙΣΑΓΩΓΗ

Μηχανική όραση

Έλεγχος ποιότητας

Μηχανική όραση

Ρομποτική

Εξόρυξη δεδομένων

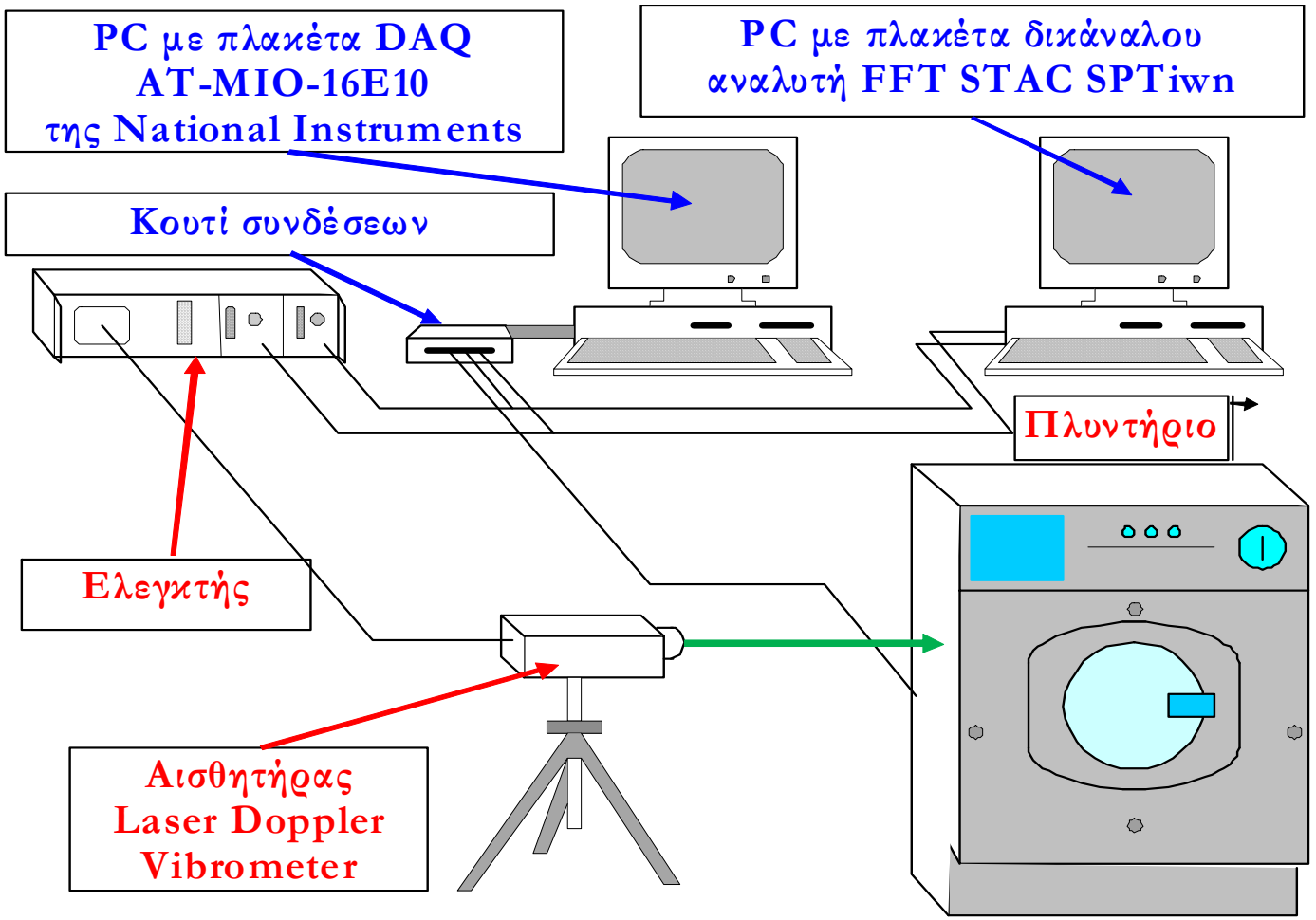
Βάσεις δεδομένων εικόνων

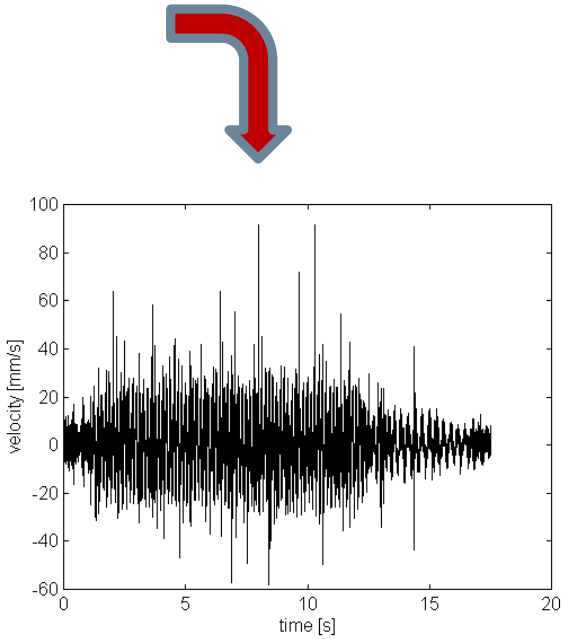
Πληροφορική

ΕΙΣΑΓΩΓΗ

Ευρωπαϊκό έργο MEDEA

Έλεγχος ποιότητας σε γραμμή παραγωγής πλυντηρίων





ΕΙΣΑΓΩΓΗ

Ο άνθρωπος παραμένει ο καλύτερος «αναγνωριστής προτύπων» (;)

completely automated public Turing test to tell computers and humans apart

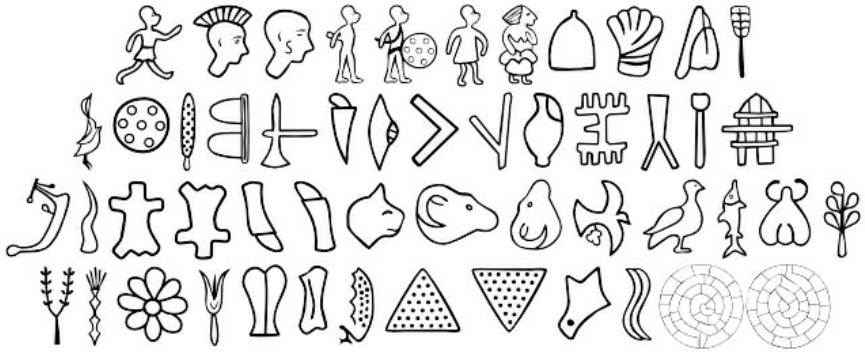
Τμήμα Γ4, ΒΠΣ, 1969
ΕΓΩ;

Ο τρόπος που το καταφέρνει δεν έχει κατανοηθεί πλήρως ακόμα

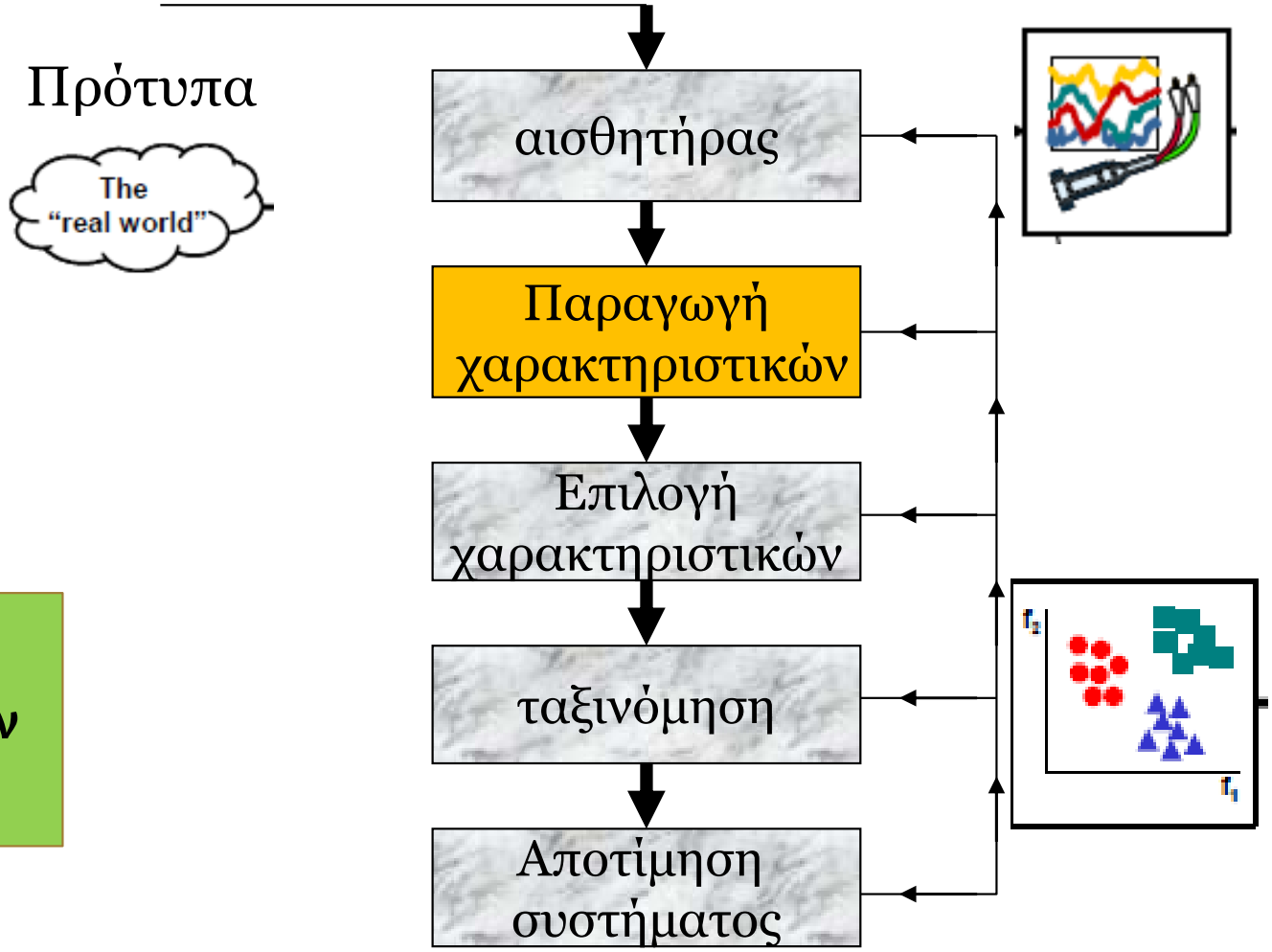
ΕΙΣΑΓΩΓΗ

Ο δίσκος της Φαιστού

Διάμετρος ~ 16 cm, πηλός με έντυπα σχήματα και στις δύο πλευρές (122+119, 45 μοναδικά).
Ανακαλύφθηκε το 1908 από τον Luigi Pernier.



Διαδικασία μηχανικής αναγνώρισης προτύπων



Η διαδικασία αυτή μιμείται εν πολλοίς την ανθρώπινη

ΕΙΣΑΓΩΓΗ

Κατηγοριοποίηση μεθόδων

Με επίβλεψη **Γνωστός αριθμός κλάσεων και ικανό πλήθος παραδειγμάτων από κάθε κλάση.**
Χωρίς επίβλεψη **Το είδος και πλήθος των κλάσεων είναι άγνωστο. Παραδείγματα από άγνωστες καταστάσεις.**

Παραδείγματα συνόλων δεδομένων

MEDEA

IRIS

Μαθηματική τυποποίηση του προβλήματος

Έστω n αντικείμενα (πρότυπα)

$$[x_1, x_2, \ldots, x_n]^T$$

(μετρήσεις, εικόνες, ήχοι κλπ)

με q χαρακτηριστικά (features) το καθένα σε μορφή μητρώου X διάστασης $n \times q$.

Στόχος είναι η ομαδοποίηση των αντικειμένων σε m ομάδες, κλάσεις ή συστάδες (clusters).

Αναγνώριση με επίβλεψη

Συλλέγουμε n δεδομένα (αντικείμενα) που ανήκουν σε m, γνωστές, κλάσεις $\omega_1, \ldots, \omega_m$.

Ορίζουμε το σύνολο εκμάθησης ως,

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

όπου $y_i \in \{1, 2, \ldots, m\}$

Αναζητούμε έναν κανόνα απόφασης d τέτοιον ώστε κάθε νέο αντικείμενο x να ταξινομείται σε μία από τις κλάσεις m με «βέλτιστο» τρόπο.

ΕΙΣΑΓΩΓΗ

Δομή παραδόσεων

1. Εισαγωγή.
2. Στατιστικές μέθοδοι: ταξινομητές Bayes.
3. Στατιστικές μέθοδοι: εκτιμητές κατανομών.
4. Μη στατιστικές μέθοδοι: γραμμικοί ταξινομητές.
5. «Ευφυείς» μέθοδοι.
6. Ταξινόμηση χωρίς επίβλεψη.

Επιθυμητές γνώσεις

1. Θεωρία πιθανοτήτων.
2. Άλγεβρα και λογισμός πινάκων.
3. Γραμμικός και μη γραμμικός προγραμματισμός.

Βιβλιογραφία

Σαγκριώτης Ε., Θεοδωρίδης Σ. (2003). «Σήματα και επεξεργασία εικόνας: Τόμος Γ' Ανάλυση εικόνας και αναγνώριση προτύπων», ΕΑΠ.

Theodoridis S. and Koutroumbas K. (1998). "Pattern recognition", Academic Press. (και στα Ελληνικά)

Bishop C. M. (1995). "Neural networks for pattern recognition", Oxford University Press.

Duda R., Hart P., Stork D. (2008). "Pattern classification", Wiley.

Fukunaga K. (1990). "Statistical pattern recognition", Academic Press.

Άσκηση 1

Μεταφράστε «κατάλληλα» τους όρους:

Pattern **Feature** **Cluster**

«Κατάλληλα»:

1. **Να μην υπάρχει άλλη αγγλική λέξη που να μεταφράζεται σαν την ελληνική, δηλαδή η συνάρτηση να είναι 1-1 (π.χ. «χαρακτηριστικό» μεταφράζεται το "characteristic", άρα αποκλείεται ως μετάφραση του feature, standard: πρότυπο …).**
2. **Μεταφράζουμε με βάση την ετυμολογία και όχι τη σημασία.**
3. **Μονολεκτικά.**

Bayes - υποθέσεις

- Χειριζόμαστε τα αντικείμενα ως τυχαίες μεταβλητές, ανεξάρτητες δεδομένης της κλάσης.
- Θεωρούνται γνωστά, ή εκ θεωρίας ή από πληροφορία:

$P(\omega_i)$: πρότερη πιθανότητα κλάσης ω_i
(η πιθανότητα το αντικείμενο x ν' ανήκει στη κλάση ω_i)

$P(\omega_i|x)$: υπό συνθήκη ή ύστερη πιθανότητα της κλάσης ω_i δεδομένου του x
(η πιθανότητα το νέο αντικείμενο να ανήκει στη κλάση ω_i, δεδομένου ότι το αντίστοιχο διάνυσμα χαρακτηριστικών είναι x)

Ταξινομητές Bayes

- **Κριτήριο βέλτιστου:** ελαχιστοποίηση **πιθανότητας σφάλματος** (σφάλματος Bayes) P_ε

$$P_\varepsilon = \min_{R_i}\left\{1-\sum_{i=1}^{L}\left\{P(\omega_i)\int_{R_i} p(x|\omega_i)\,dx\right\}\right\}$$

$p(x|\omega_i)$: συνάρτηση πυκνότητας πιθανότητας υπό συνθήκη για την κλάση ω_i

R_i: περιοχές όπου το x ταξινομείται στη κλάση ω_i σύμφωνα με τον κανόνα d

$p(x)$: συνάρτηση πυκνότητας πιθανότητας του $x(s)$ (ή **συνάρτηση πυκνότητας μίγματος**)

P_ε=1-(πιθανότητα σωστής ταξινόμησης): ο i-οστός όρος παριστάνει τη πιθανότητα σωστής ταξινόμησης της κλάσης i επί τη πιθανότητα εμφάνισής της.

Ταξινομητές Bayes

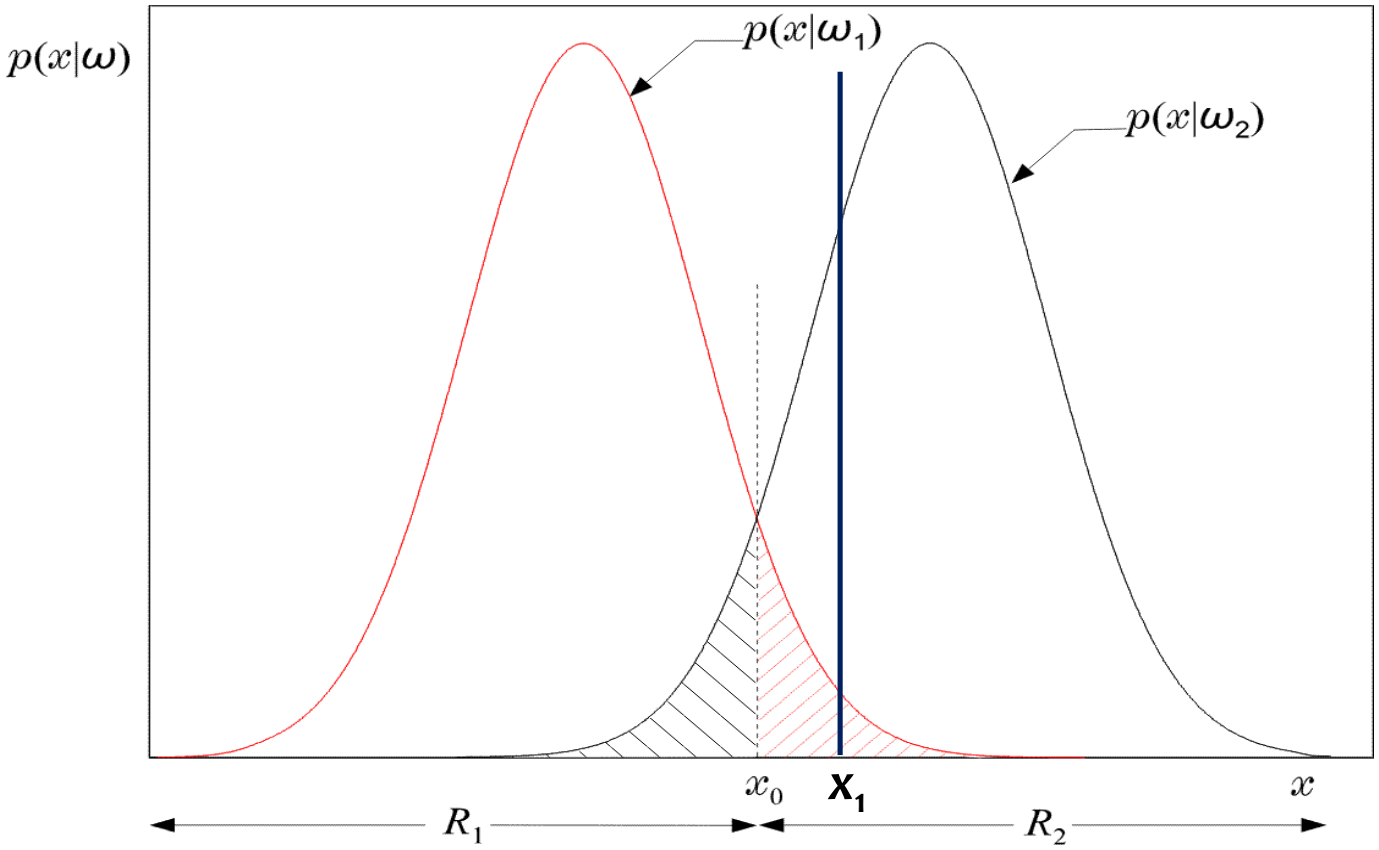
- Κανόνας Bayes:

  Το νέο αντικείμενο με διάνυσμα χαρακτηριστικών x, ταξινομείται στην πλέον **πιθανή** κλάση:

$$x \rightarrow \omega_i : P(\omega_i | x) \quad \max$$

ύστερη πιθανότητα

Ταξινομητές Bayes



Οι περιοχές R_1, R_2 του ταξινομητή Bayes για ισοπίθανες κλάσεις. Το σημείο x_1 ταξινομείται στη κλάση 2 αφού $p(x|\omega_2) > p(x|\omega_1)$ αλλά υπάρχει και μικρή πιθανότητα ν' ανήκει στην 1. Το σημείο x_0 στο οποίο $p(x|\omega_2) = p(x|\omega_1)$ καλείται **σημείο διάκρισης** ή **διαχωρισμού**.

$$P_\varepsilon = \frac{1}{2}\int_{-\infty}^{x_0} p(x|\omega_2)\,dx + \frac{1}{2}\int_{x_0}^{+\infty} p(x|\omega_1)\,dx$$

Ταξινομητές Bayes

Απόδειξη

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

$$P_e = P(\boldsymbol{x} \in R_2|\omega_1)P(\omega_1) + P(\boldsymbol{x} \in R_1|\omega_2)P(\omega_2)$$ από κοινού πιθανότητα

$$= P(\omega_1) \int_{R_2} p(\boldsymbol{x}|\omega_1)\, d\boldsymbol{x} + P(\omega_2) \int_{R_1} p(\boldsymbol{x}|\omega_2)\, d\boldsymbol{x}$$

$$P_e = \int_{R_2} P(\omega_1|\boldsymbol{x})p(\boldsymbol{x})\, d\boldsymbol{x} + \int_{R_1} P(\omega_2|\boldsymbol{x})p(\boldsymbol{x})\, d\boldsymbol{x}$$ Bayes

Επειδή R_1, R_2 καλύπτουν όλο το χώρο:

$$\int_{R_1} P(\omega_1|\boldsymbol{x})p(\boldsymbol{x})\, d\boldsymbol{x} + \int_{R_2} P(\omega_1|\boldsymbol{x})p(\boldsymbol{x})\, d\boldsymbol{x} = P(\omega_1)$$

Ταξινομητές Bayes

έπεται $$P_e = P(\omega_1) - \int_{R_1} (P(\omega_1|\boldsymbol{x}) - P(\omega_2|\boldsymbol{x}))\, p(\boldsymbol{x})\, d\boldsymbol{x}$$

Άρα η περιοχή R_1 επιλέγεται έτσι ώστε σ' αυτή: $P(\omega_1|x) > P(\omega_2|x)$

Ταξινομητές Bayes

Παράδειγμα (Θεοδωρίδης)

Σε μια αίθουσα υπάρχουν 100 **αθλήτριες χορού** (κλάση ω_1) και 200 **καλαθοσφαιριστές** (κλάση ω_2). Έστω ότι έχουμε πληροφορία για το βάρος τους (x) που συνοψίζεται μέσω των κατανομών πυκνότητας πιθανότητας,

$$P(x|\omega_1) = \frac{e^{-\frac{(x-50)^2}{300}}}{\sqrt{300\pi}}, \quad P(x|\omega_2) = \frac{e^{-\frac{(x-90)^2}{300}}}{\sqrt{300\pi}}$$

(δηλαδή τα βάρη ακολουθούν κανονική κατανομή με μέση τιμή μ_1= 50 για τις αθλήτριες χορού και μ_2=90 για τους καλαθοσφαιριστές και ίση διασπορά σ^2=300).

Ένα νέο άτομο εισέρχεται στην αίθουσα με βάρος 67 κιλά. Που θα το κατατάξουμε;

Ταξινομητές Bayes

Σύμφωνα με τον ταξινομητή Bayes θα ταξινομηθεί στη κλάση για την οποία $P(\omega_i|x)$ max. Ξέρουμε,

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$$

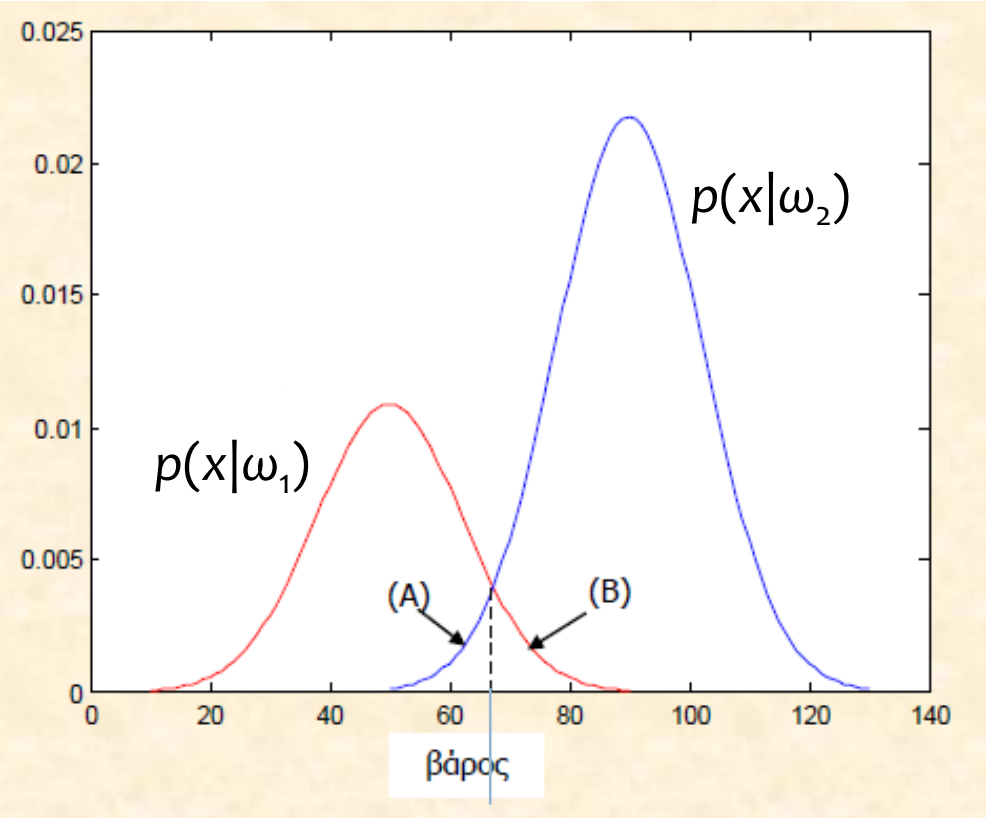
Επομένως θα υπολογίσουμε $p(x|\omega_i)P(\omega_i)$.
Από τα δεδομένα, $P(\omega_1)=100/300=1/3$, $P(\omega_2)=200/300=2/3$

$$i=1:\ p(x|\omega_1)P(\omega_1)= \frac{e^{-\frac{(67-50)^2}{300}}}{\sqrt{300\pi}}\frac{1}{3} = \mathbf{0,0041}$$

$$i=2:\ p(x|\omega_2)P(\omega_2)= \frac{e^{-\frac{(67-90)^2}{300}}}{\sqrt{300\pi}}\frac{1}{3} = 0,0037$$

Άρα κατατάσσεται στις χορεύτριες.

Ταξινομητές Bayes



Μπορούμε να υπολογίσουμε τη διαχωριστική τιμή λύνοντας την,

$$\frac{1}{3}\frac{1}{\sqrt{300\pi}}e^{-\frac{(x-50)^2}{300}} = \frac{2}{3}\frac{1}{\sqrt{300\pi}}e^{-\frac{(x-90)^2}{300}} \Leftrightarrow$$

$$-(x-50)^2 = -(x-90)^2 + 300\ln 2 \Leftrightarrow x \approx 67.4$$

όπως επίσης και τη πιθανότητα λάθους, υπολογίζοντας τα εμβαδά Α, Β

$$A+B = \int_{R_2} p(x|\omega_1)p(\omega_1)dx + \int_{R_1} p(x|\omega_2)p(\omega_2)dx$$

$$= \frac{1}{3}\int_{67,4}^{\infty} p(x|\omega_1)dx + \frac{2}{3}\int_{-\infty}^{67,4} p(x|\omega_2)dx$$

Ταξινομητές Bayes

Χρειάζεται τρόπος υπολογισμού της ύστερης πιθανότητας !

Κανόνας του Bayes (για δύο κλάσεις):

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$$

$$p(x) = \sum_{i=1}^{2} p(x|\omega_i)P(\omega_i)$$

Επομένως πρέπει να ξέρουμε τις **πρότερες** πιθανότητες $P(\omega_i)$ και τις **πιθανοφάνειες** $p(x/\omega_i)$

Bayes – κανονικές κατανομές

Στη περίπτωση που οι εμπλεκόμενες συναρτήσεις πυκνότητας πιθανότητας είναι κανονικές, τα πράγματα απλουστεύονται (;), αφού:

$$p(x|\omega_i)=\frac{1}{(2\pi)^{\frac{q}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-m_i)^T\Sigma_i^{-1}(x-m_i)\right\},\ i=1,\ldots,q$$

(η μέση τιμή και η συνδιασπορά συνήθως εκτιμώνται).

Έτσι ο ταξινομητής Bayes έχει τη λογική:

$$d=\arg\left\{\max_i\left\{\frac{1}{|\Sigma_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(x-m_i)^T\Sigma_i^{-1}(x-m_i)\right\}P(\omega_i)\right\}\right\}$$

Ταξινομητές Bayes

Η εκθετική μορφή της *d* παραπέμπει στη δυνατότητα μετασχηματισμού της για ευκολότερο χειρισμό. Έτσι παίρνοντας λογαρίθμους (μονοτονική συνάρτηση):

$$g_i(x) = -\frac{1}{2}(x-m_i)^T \Sigma_i^{-1}(x-m_i) + \ln P(\omega_i) + c_i \cdot i=1,\ldots,q$$

$$c_i = -\left(\frac{q}{2}\right)\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i|$$

Η μορφή αυτή είναι γενικά **μη γραμμική τετραγωνική**. Οι **επιφάνειες διαχωρισμού** που ορίζονται από τις εξισώσεις,

$$g_i(x) - g_j(x) = 0$$

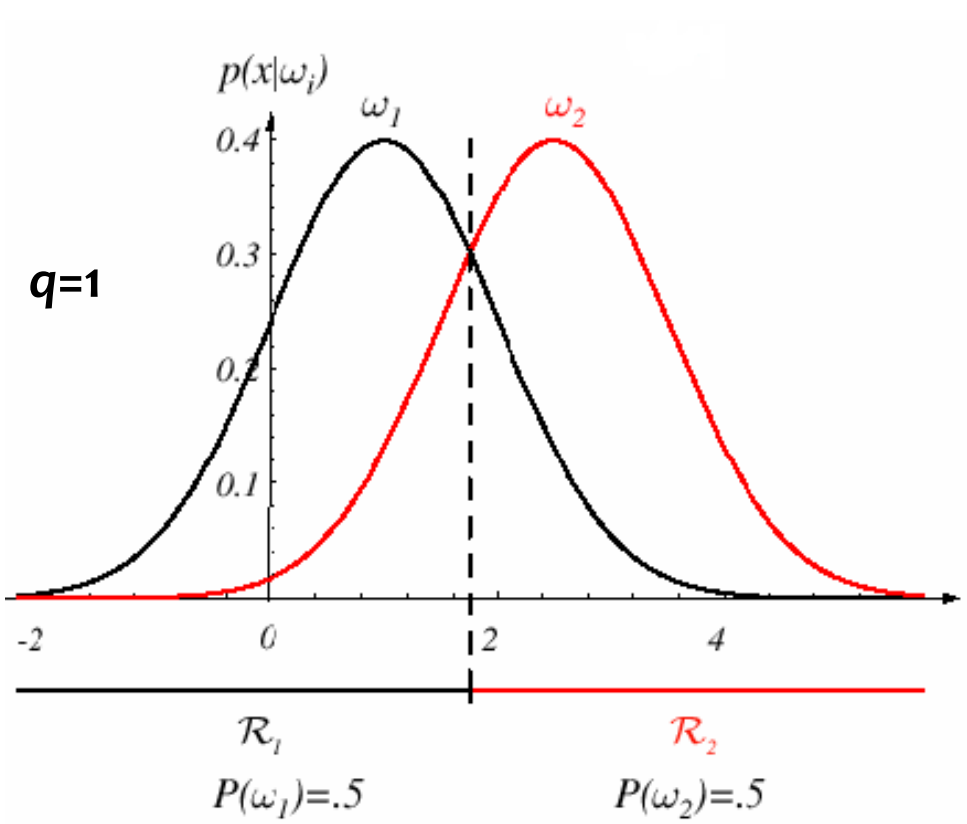
(**όχι όμως για όλα τα ζεύγη *i, j***) είναι γενικά **υπερτετραγωνικές** μορφές (**υπερελλειψοειδή** κλπ).

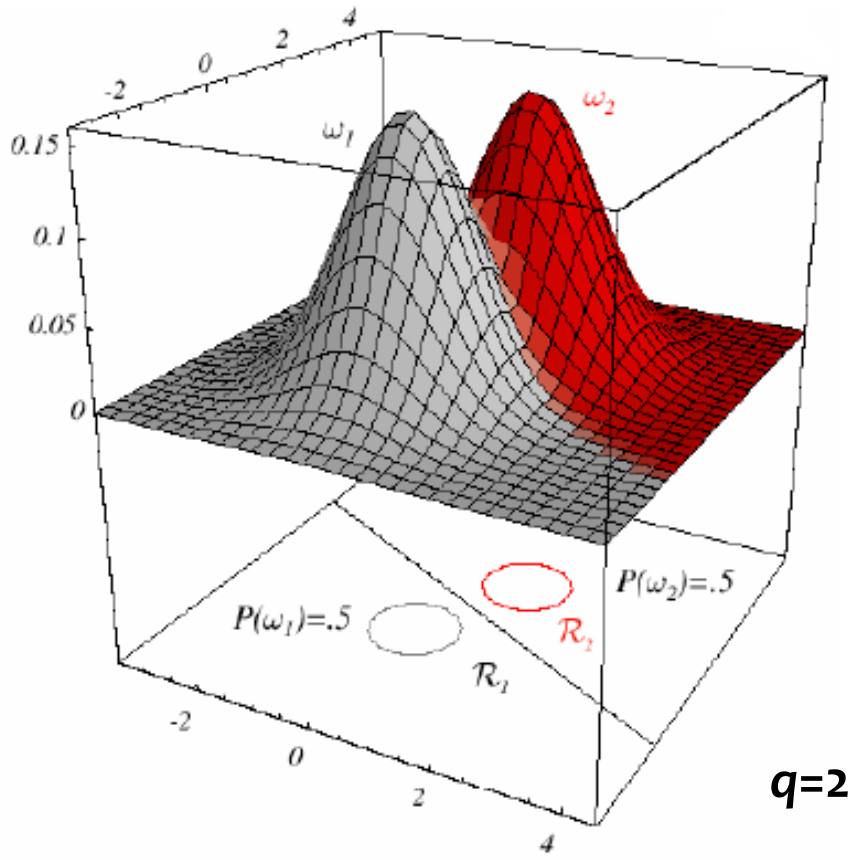
Για καλύτερη κατανόηση ας δούμε τις περιπτώσεις κατά σειρά πολυπλοκότητας.

Ταξινομητές Bayes

1. $\Sigma_i=\sigma^2 I$ (στατιστική ανεξαρτησία χαρακτηριστικών, ίση διασπορά)

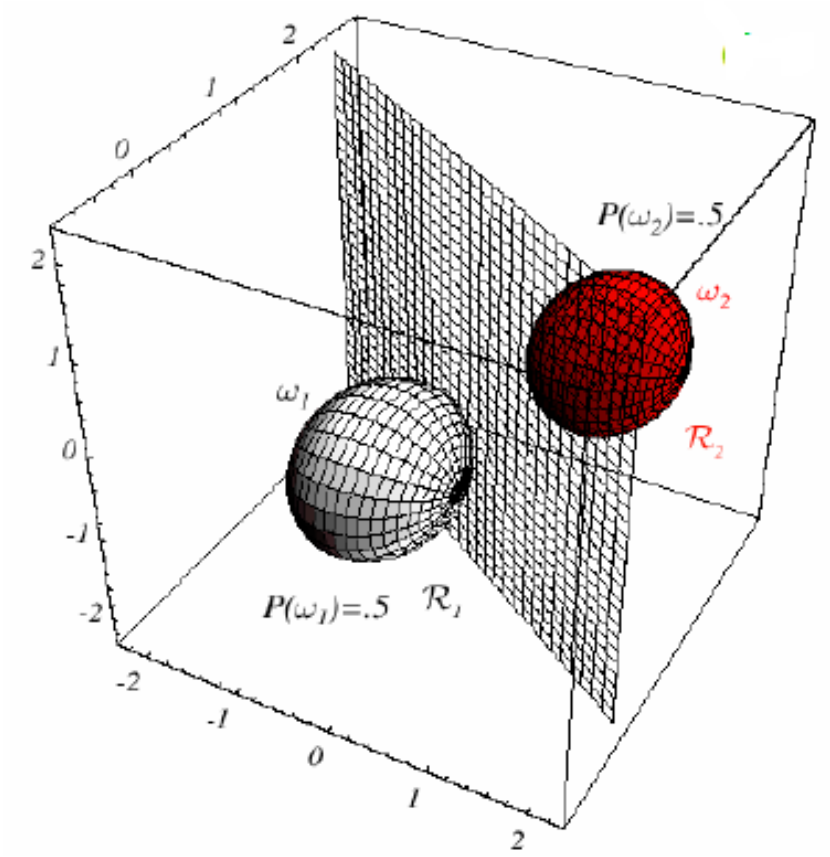
Τα δείγματα κείνται σε ίσες υπερσφαίρες και οι επιφάνειες απόφασης είναι υπερεπίπεδα διάστασης q-1





Ταξινομητές Bayes

q=3



Π. Τσακαλίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ
ΥΠΟΛΟΓΙΣΤΩΝ

Η συνάρτηση πυκνότητας πιθανότητας είναι 4-διάστατη εδώ.

Ταξινομητές Bayes

Στη περίπτωση αυτή η $g_i(x)$ για,

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

γράφεται,

$$g_i(x) = w_i^T x + w_{io}$$

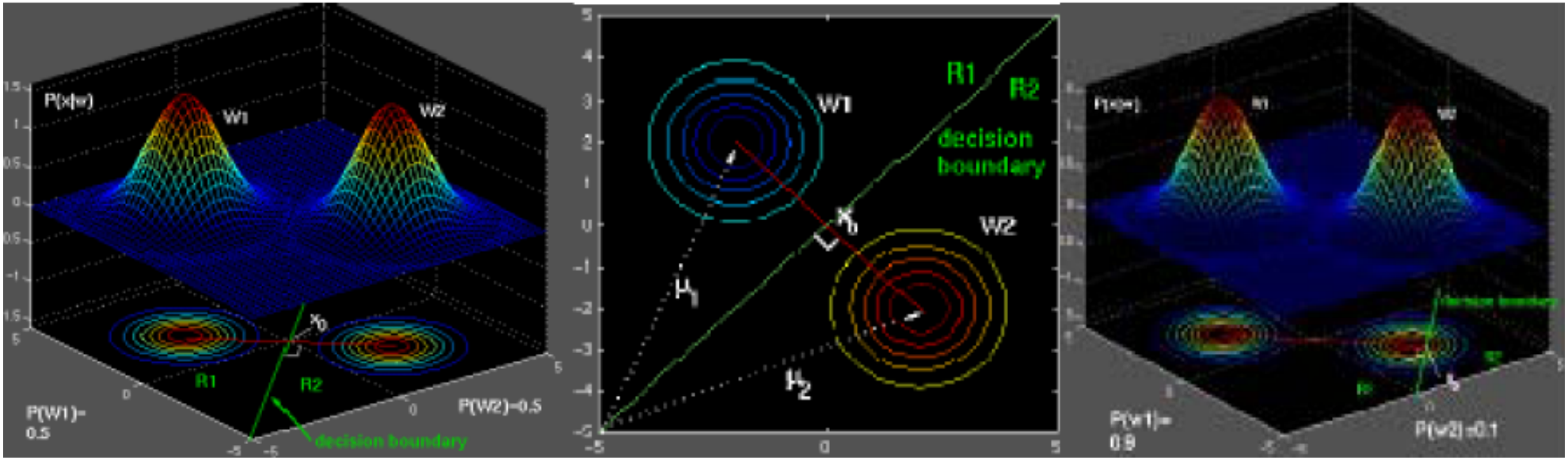
$$w_i = \frac{1}{\sigma^2} m_i, \quad w_{io} = -\frac{1}{2\sigma^2} m_i^T m_i + \ln P(\omega_i)$$

Ενώ $g_i(x) - g_j(x) = w^T(x - x_o)$

$$w = m_i - m_j, \quad x_o = \frac{1}{2}(m_i + m_j) - \frac{\sigma^2}{\|m_i + m_j\|^2} + \ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)(m_i - m_j)$$

Ταξινομητές Bayes

Επιφάνεια απόφασης: Υπερ-επίπεδο που περνά από το σημείο x_0 και είναι κάθετο στο διάνυσμα w που ενώνει τις μέσες τιμές μ_i και μ_j.



Π. Τσακαλίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ
ΥΠΟΛΟΓΙΣΤΩΝ

2. $\Sigma_i=\Sigma$

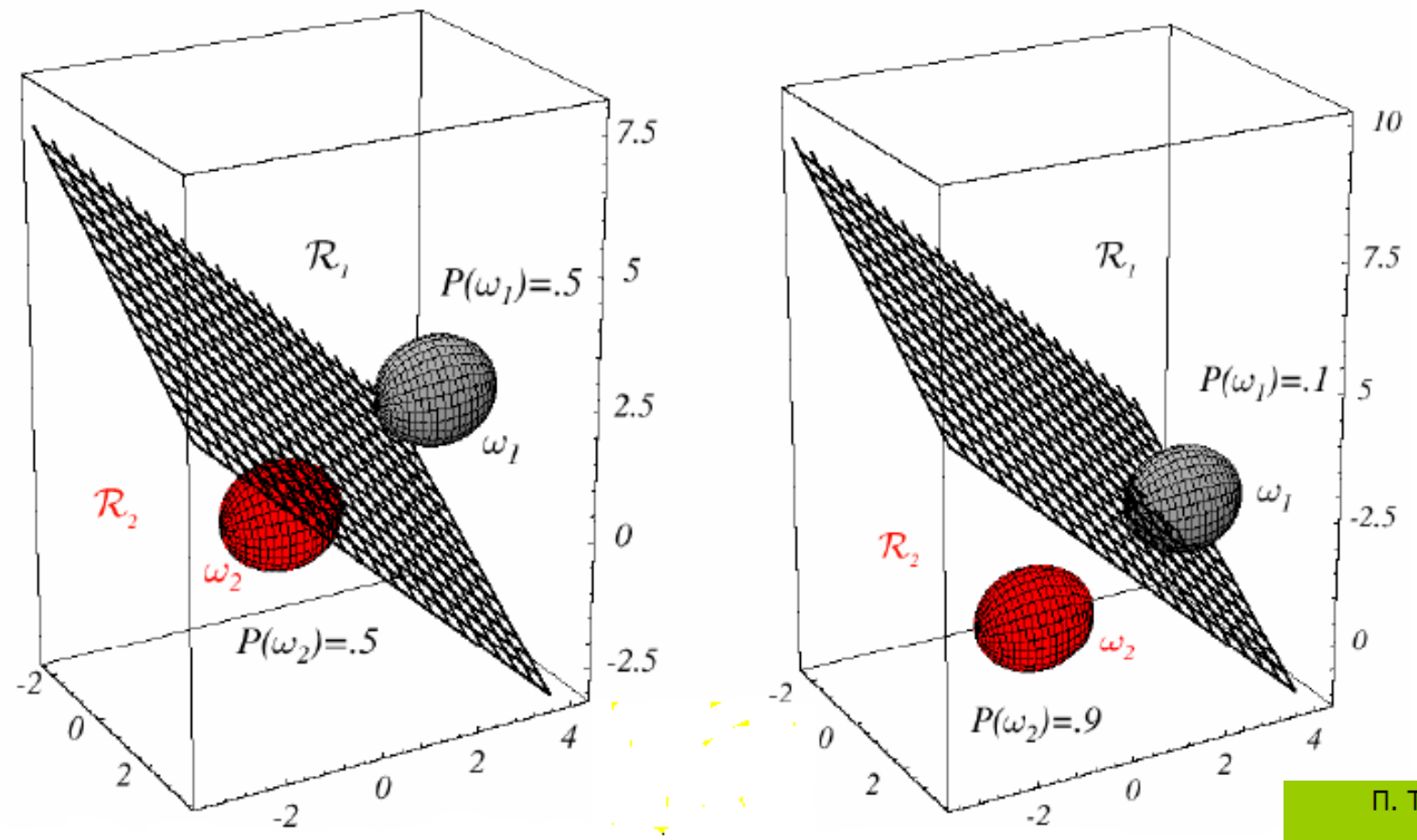
Τα χαρακτηριστικά σε υπερελλειψοειδή ίδιου μεγέθους.
Οι επιφάνειες απόφασης υπερεπίπεδα:

$$g_i(x) = w_i^T x + w_{io}$$

$$w_i = \Sigma^{-1} m_i, \quad w_{io} = -\frac{1}{2} m_i^T \Sigma^{-1} m_i + \ln P(\omega_i)$$

Συγκρίνετε με: $w_i = \frac{1}{\sigma^2} m_i, \quad w_{io} = -\frac{1}{2\sigma^2} m_i^T m_i + \ln P(\omega_i)$

Ταξινομητές Bayes



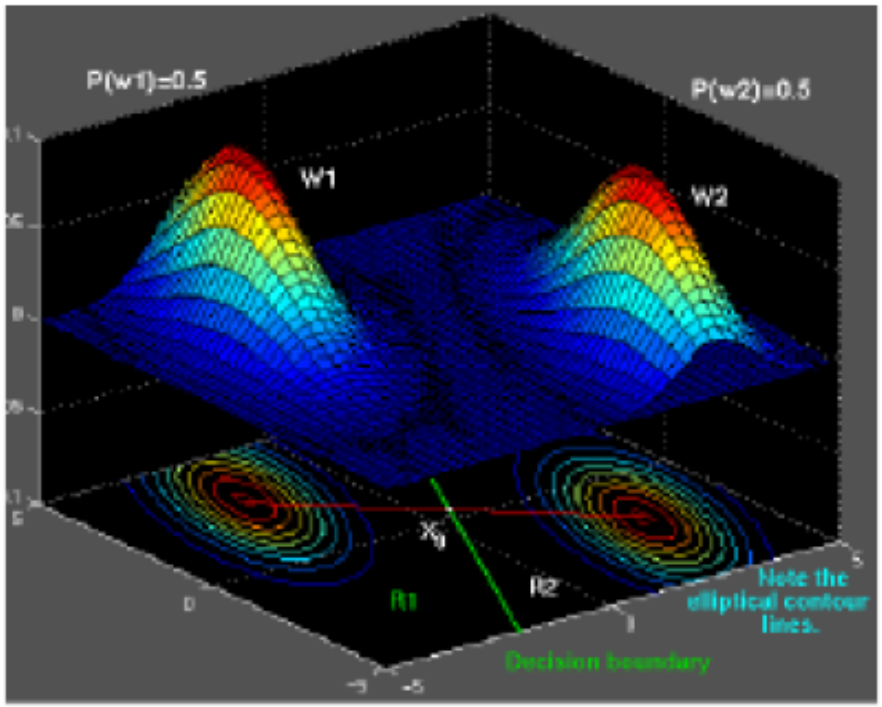
Π. Τσακαλίδης

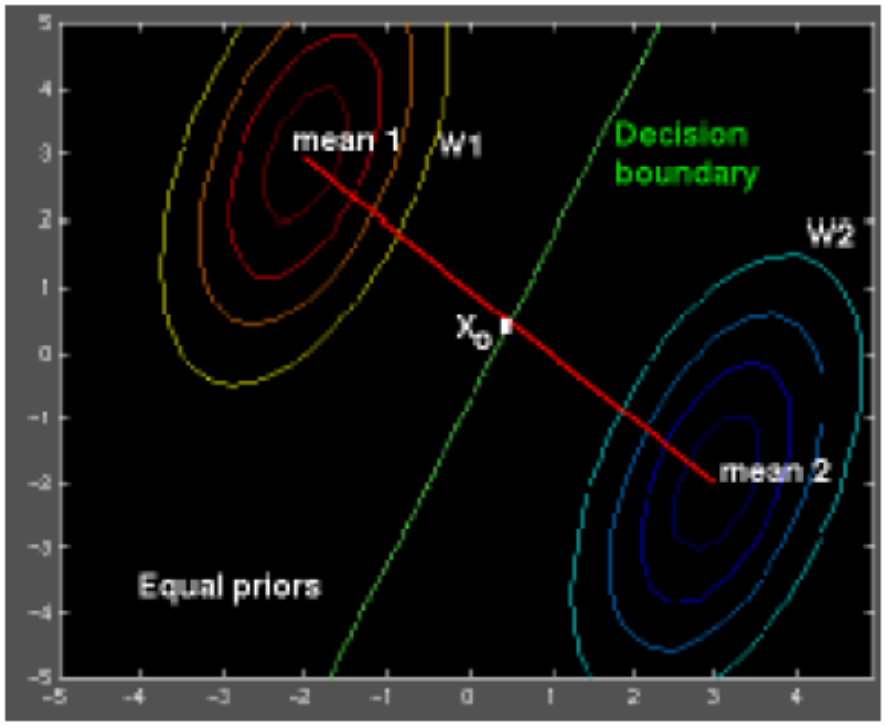
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ
ΥΠΟΛΟΓΙΣΤΩΝ

Ταξινομητές Bayes

$$g_i(x) - g_j(x) = w^T(x - x_o)$$

$$w = \Sigma^{-1}(m_i - m_j), \quad x_o = \frac{1}{2}(m_i + m_j) - \frac{1}{(m_i - m_j)^T \Sigma^{-1}(m_i - m_j)} + \ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right)(m_i - m_j)$$





Εφόσον $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$ το υπερεπίπεδο απόφασης ***δεν*** είναι κάθετο στο διάνυσμα $\boldsymbol{w}$ που ενώνει τις μέσες τιμές $\boldsymbol{\mu}_i$ και $\boldsymbol{\mu}_j$.

3. Σ_i αυθαίρετο

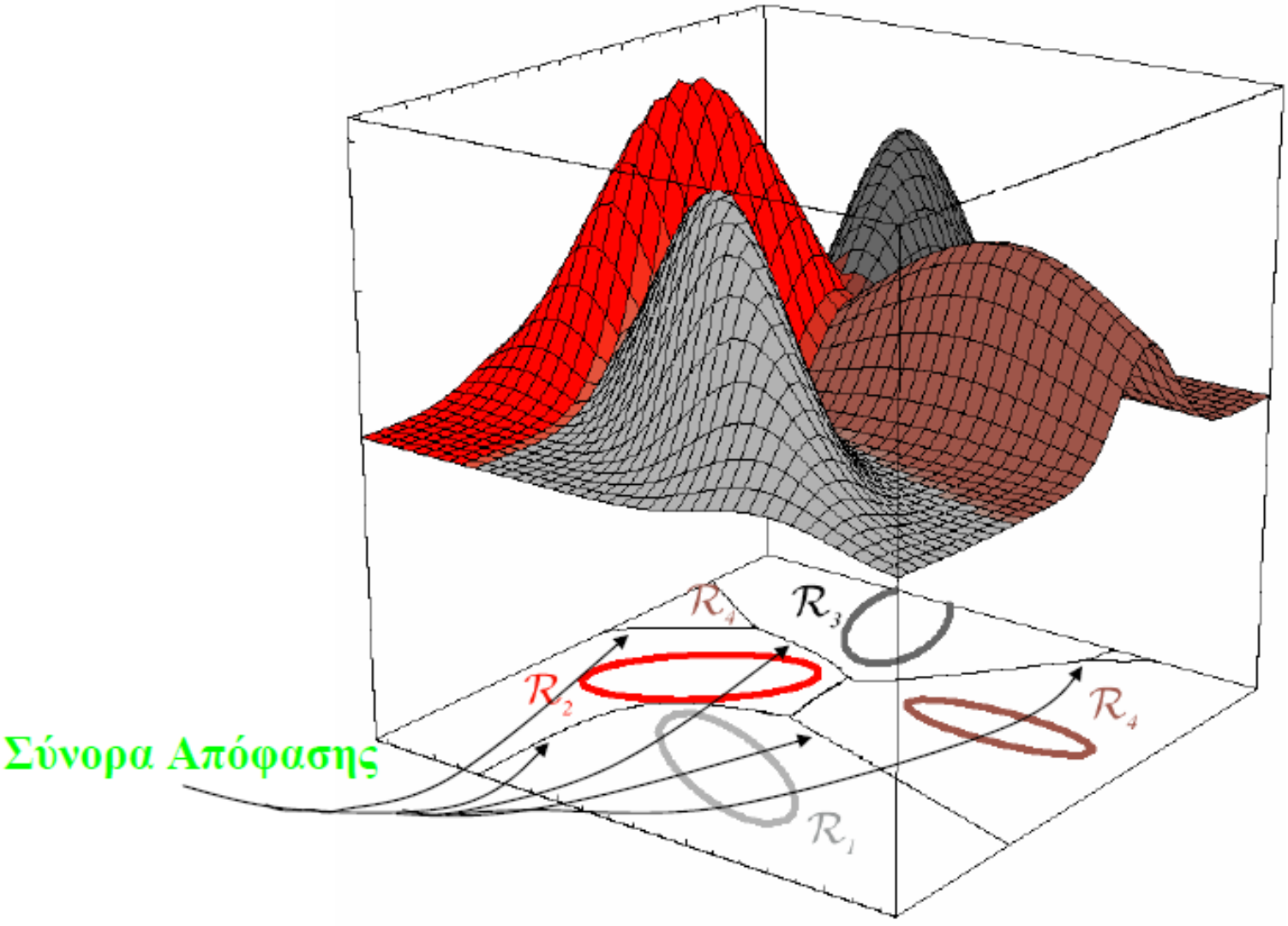
Οι επιφάνειες απόφασης μη γραμμικές, τετραγωνικές:

$$g_i(x) = x^T W_i x + w_i^T x + w_{i0}$$

$$W_i = -\frac{1}{2}\Sigma_i^{-1}, \quad w_i = \Sigma_i^{-1} m_i, \quad w_{i0} = -\frac{1}{2} m_i^T \Sigma_i^{-1} m_i - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

Συγκρίνετε με: $w_i = \frac{1}{\sigma^2} m_i, \quad w_{i0} = -\frac{1}{2\sigma^2} m_i^T m_i + \ln P(\omega_i)$

Ταξινομητές Bayes

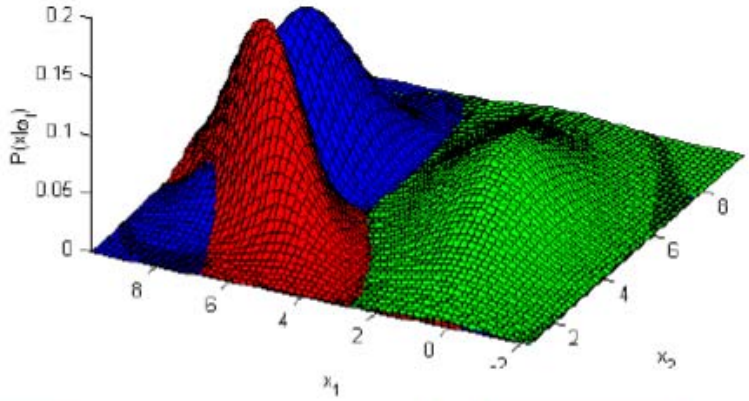


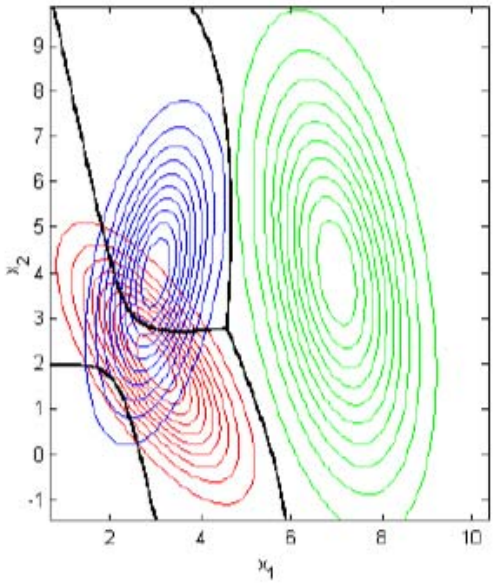
Ταξινομητές Bayes

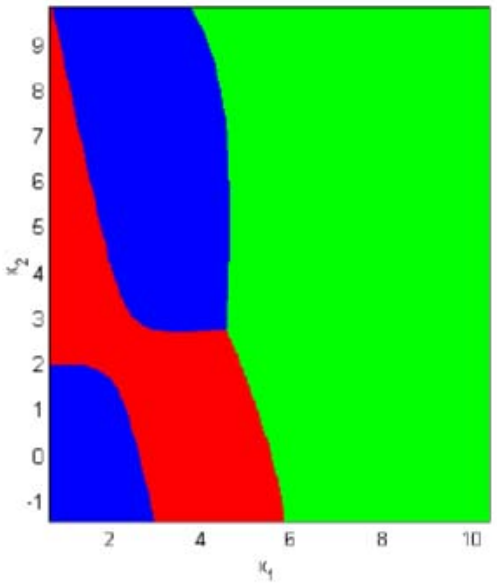
3 κλάσεις, 2 χαρακτηριστικά
ίσες πρότερες πιθανότητες
διαφορετικές, μη διαγώνιες Σ

$$\mu_1 = [3\ 2]^T \qquad \mu_2 = [5\ 4]^T \qquad \mu_3 = [3\ 4]^T$$

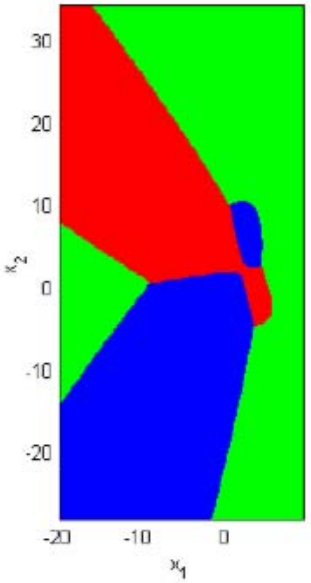
$$\Sigma_1 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1 & -1 \\ -1 & 7 \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} .5 & .5 \\ .5 & 3 \end{bmatrix}$$







Zoom
out



CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

Bayes ελαχίστου ρίσκου

Η ελαχιστοποίηση της πιθανότητας λάθους μπορεί να μην είναι το κατάλληλο κριτήριο σε κάποιες περιπτώσεις, αφού υπονοεί ότι όλα τα λάθη έχουν την **ίδια βαρύτητα**.

Αν αυτό δεν ισχύει, η θεωρία που παρουσιάστηκε μπορεί εύκολα να επεκταθεί για να καλύψει περιπτώσεις όπου κάποια λάθη έχουν σοβαρότερες επιπτώσεις από άλλα.

Έστω c_{ki} το κόστος (ζημία) της κατάταξης ενός αντικειμένου στη (λάθος) κλάση i, ενώ η σωστή είναι k. Ο πίνακας των στοιχείων c_{ij} καλείται **πίνακας κόστους**.

Ταξινομητές Bayes

Το συνολικό κόστος σ' αυτή τη περίπτωση είναι,

$$r = \sum_{i=1}^{q} \int_{R_i} \left(\sum_{k=1}^{q} c_{ki} p\left(x \middle| \omega_k\right) P(\omega_k) \right) \mathrm{d}x$$

ενώ ο βέλτιστος κανόνας:

$$d : x \rightarrow \omega_i : \sum_{k=1}^{q} c_{ki} P(\omega_k) p(x|\omega_k) < \sum_{k=1}^{q} c_{kj} P(\omega_k) p(x|\omega_k), \quad \forall j \neq i$$

(η περίπτωση αυτή περιλαμβάνει και τα προηγούμενα).

Πρακτικά θέματα

Η πραγματικότητα είναι λίγο πιο πολύπλοκη από τη θεωρία. Για να εφαρμόσουμε σωστά τα προηγούμενα πρέπει κατ' αρχήν να εξακριβώσουμε αν τα δεδομένα μας ακολουθούν κανονική κατανομή και στη συνέχεια να εκτιμήσουμε τις παραμέτρους της.

Αν αυτό δεν ισχύει, μπορούμε είτε να χρησιμοποιήσουμε άλλη μεθοδολογία (που θα δούμε στη συνέχεια) είτε να προχωρήσουμε, αγνοώντας το γεγονός αυτό.

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Δοκιμασία κανονικότητας

Η δοκιμασία κανονικότητας έχει μελετηθεί εξονυχιστικά και στη βιβλιογραφία αναφέρονται πλήθος δοκιμασιών, τόσο για μονοδιάστατες όσο και για πολυδιάστατες κατανομές.

Το γεγονός αυτό δείχνει ότι το πρόβλημα, αν και εκ πρώτης όψεως φαίνεται απλό, δεν έχει λυθεί ικανοποιητικά. Μια καλή ανασκόπηση των μεθόδων υπάρχει στη δημοσίευση:

Henze N., "Invariant tests for multivariate normality: a critical review", *Statistical Papers*, **43**:467-506, (2002).

Από τη πληθώρα των δοκιμασιών επιλέγω αυτό που βασίζεται στην **ασυμμετρία** (=0) και **κύρτωση** (=3) της κανονικής κατανομής.

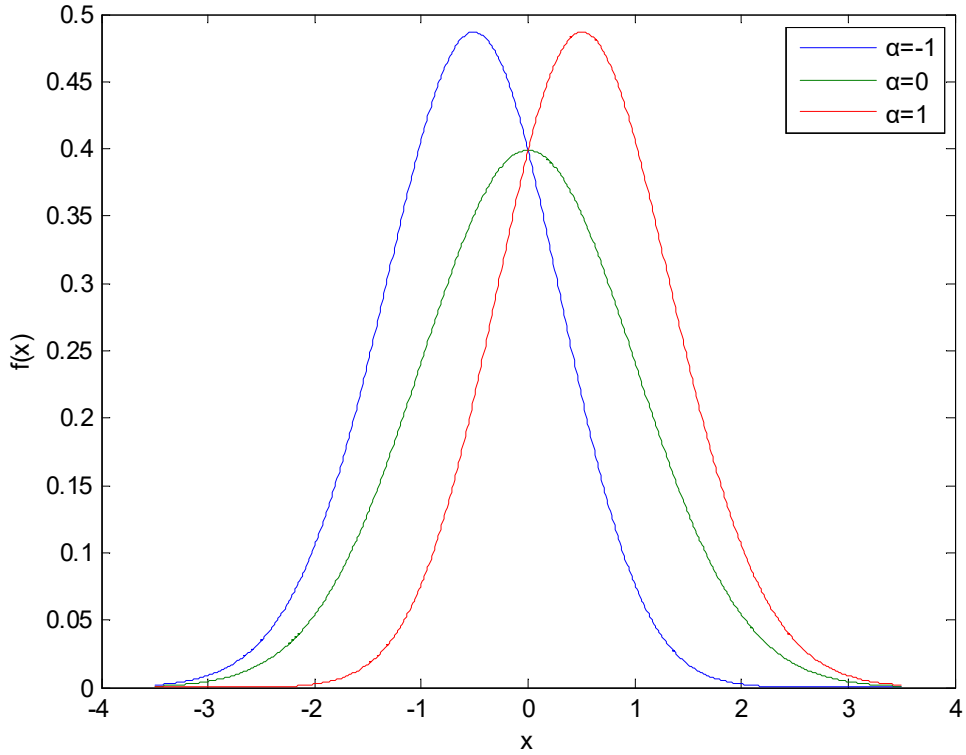
Ταξινομητές Bayes/εκτίμηση παραμέτρων

Η **ασυμμετρία** ορίζεται ως,

$$\gamma_1 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\kappa_3}{\kappa_2^{\frac{3}{2}}}$$

και όπως είναι προφανές είναι μέτρο της ασυμμετρίας μίας κατανομής.

Ταξινομητές Bayes/εκτίμηση παραμέτρων



$$f(x,\alpha)=\frac{1}{\sigma\pi}e^{-\frac{(X-\mu)^2}{2\sigma^2}}\int_{-\infty}^{\alpha\left(\frac{X-\mu}{\sigma}\right)}e^{-\frac{t^2}{2}}dt$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Εκτίμηση παραμέτρων

Εφ' όσον αποφασίστηκε ότι οι εμπλεκόμενες κατανομές είναι κανονικές, το επόμενο βήμα είναι η εκτίμηση των παραμέτρων τους, δηλαδή της μέσης τιμής μ_i και συνδιασποράς Σ_i για κάθε κλάση. Η πλέον ενδεδειγμένη μέθοδος για τη διαδικασία αυτή είναι της **μέγιστης πιθανοφάνειας**. Έστω,

$X = \{x_1, x_2, ..., x_N\}$ N ανεξάρτητα δείγματα από μία κλάση, και

$$p(X) \equiv p(X;\vartheta) = p(x_1, x_2, ..., x_N;\vartheta) = \prod_{k=1}^{N} p(x_k;\vartheta)$$

η συνάρτηση πιθανοφάνειας του ϑ ως προς το X. Ορίζουμε ως εκτιμήτρια μέγιστης πιθανοφάνειας, την,

Ταξινομητές Bayes/εκτίμηση παραμέτρων

$$\hat{\vartheta}_{ML}: \ \arg\max_{\vartheta} \prod_{k=1}^{N} p(x_k;\vartheta)$$

$$L(\vartheta) \equiv \ln p(X;\vartheta) = \sum_{k=1}^{N} \ln p(x_k;\vartheta)$$

$$\hat{\vartheta}_{ML}: \ \frac{\partial L(\vartheta)}{\partial(\vartheta)} = 0 \Rightarrow \sum_{k=1}^{N} \frac{1}{p(x_k;\vartheta)} \frac{\partial p(x_k;\vartheta)}{\partial(\vartheta)} = 0$$

$$p(x|m,\Sigma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{q}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x_i - m)^{\mathrm{T}} \Sigma^{-1} (x_i - m)\right\} \Rightarrow$$

$$L(m,\Sigma) = \ln p(x|m,\Sigma) = c - \frac{N}{2} \ln|\Sigma| - \frac{1}{2} \mathrm{tr}\left\{\Sigma^{-1} \sum_{i=1}^{N} (x_i - m)(x_i - m)^{\mathrm{T}}\right\}$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Χρησιμοποιήθηκε η ιδιότητα των συμμετρικών πινάκων:

$$x^{\mathrm{T}}Ax = \mathrm{tr}(x^{\mathrm{T}}Ax) = \mathrm{tr}(Axx^{\mathrm{T}})$$

Για ελάχιστο: $$\frac{\partial L}{\partial m} = \Sigma^{-1}\sum_{i=1}^{N}(x_i - m)^{\mathrm{T}} = 0$$

$$\frac{\partial L}{\partial(\Sigma^{-1})} = \frac{N}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{N}(x_i - m)(x_i - m)^{\mathrm{T}} = 0$$

Από τη πρώτη:

$$\sum_{i=1}^{N}(x_i - m)^{\mathrm{T}} = 0 \Rightarrow Nm = \sum_{i=1}^{N}x_i \Rightarrow \hat{m}_{ML} = \frac{1}{N}\sum_{i=1}^{N}x_i = \bar{x}$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Ενώ από τη δεύτερη:

$$\frac{N}{2}\Sigma-\frac{1}{2}\sum_{i=1}^{N}\left(x_i-\hat{m}_{ML}\right)\left(x_i-\hat{m}_{ML}\right)^{\mathrm{T}}=0\Rightarrow$$

$$\hat{\Sigma}_{ML}=\frac{1}{N}\sum_{i=1}^{N}\left(x_i-\hat{m}_{ML}\right)\left(x_i-\hat{m}_{ML}\right)^{\mathrm{T}}$$

Οι εκτιμήτριες μέγιστης πιθανοφάνειας έχουν επιθυμητές ιδιότητες, αλλά ασυμπτωτικά ($N\rightarrow\infty$), πχ **αμεροληψία**.

Υποδείγματα μίξης

Η προσέγγιση αυτή, που χρησιμοποιείται επίσης και σε προβλήματα χωρίς επίβλεψη, συνίσταται στην υπόθεση ότι τα δεδομένα παράγονται από ένα άθροισμα (μίγμα) άγνωστων, ως προς τις παραμέτρους (αλλά γνωστών ως προς το τύπο, π.χ. κανονικές) κατανομών με τυχαία ποσόστωση. Δηλαδή, υποθέτουμε,

$$p(x) = \sum_{j=1}^{J} p(x|j) P_j$$

$$\sum_{j=1}^{J} P_j = 1, \quad \int_R p(x|j)\mathrm{d}x = 1$$

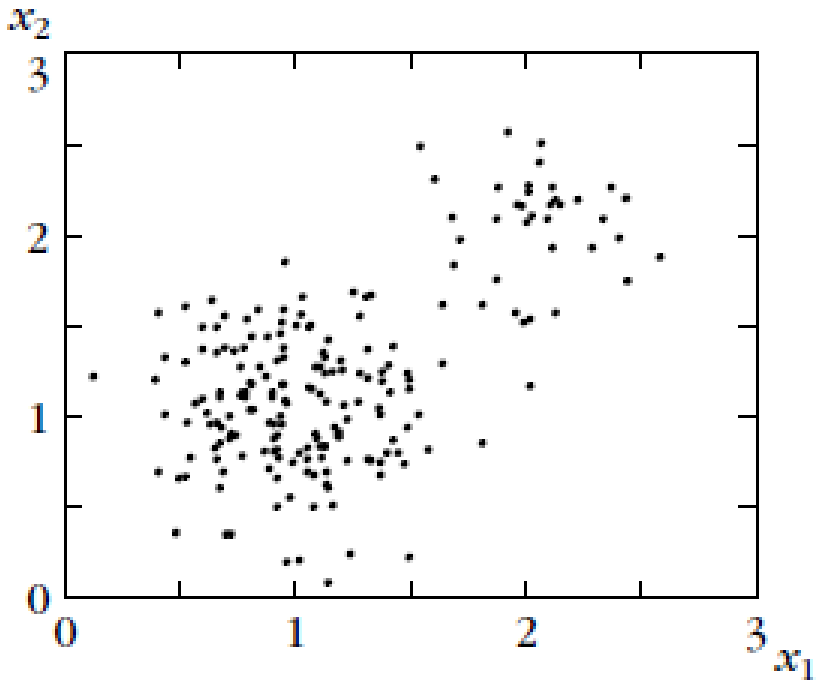
Ταξινομητές Bayes/εκτίμηση παραμέτρων

Παράδειγμα: παραγωγή 100 σημείων από μίγμα των παρακάτω κανονικών, με P_1=0,8 P_2=0,2. Δηλαδή,

$$p(x) = 0,8 \cdot \mathcal{N}(x|\mu_1, \Sigma_1) + 0,2 \cdot \mathcal{N}(x|\mu_2, \Sigma_2)$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}$$

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$$



Ταξινομητές Bayes/εκτίμηση παραμέτρων

Ας εκφράσουμε το πρόβλημα σε όρους μέγιστης πιθανοφάνειας. Δοθέντων δεδομένων,

$$X = \{x_1, x_2, \ldots, x_N\}$$

ζητείται η εύρεση των,

$$\vartheta \text{ και } P_1, P_2, \ldots, P_j$$

μέσω της λύσης της,

$$\max_{\vartheta, P_i, \ldots, P_j} \prod_{k=1}^{N} P(x_k; \vartheta, P_i, \ldots, P_j)$$

Η δυσκολία έγκειται στο ότι οι «ετικέτες» j δεν είναι γνωστές.

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Το πρόβλημα λύνεται μέσω του αναδρομικού αλγορίθμου Αναμενόμενης τιμής-Μεγιστοποίησης (ΕΜ).

Ο αλγόριθμος πρωτοχρησιμοποιήθηκε για προβλήματα με ελλιπή δεδομένα, και προσαρμόστηκε στο συγκεκριμένο ζήτημα.

Πλήρη δεδομένα: $(x_k, j_k),\ k = 1,2,...,N$

Ελλιπή δεδομένα: $x_k,\ k = 1,2,...,N$

$$p(x_k, j_k; \vartheta) = p(x_k | j_k; \vartheta) P_{jk}$$

$$L(\vartheta) = \sum_{k=1}^{N} \ln(p(x_k | j_k; \vartheta) P_{jk})$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Διάνυσμα αγνώστων: $\Theta = \begin{bmatrix} \vartheta^T & P^T \end{bmatrix}, \; P = \begin{bmatrix} P_1 & P_2 & \cdots & P_J \end{bmatrix}^T$

Βήμα Ε: $$Q(\Theta;\hat{\Theta}(t)) = E\left[\sum_{k=1}^{N} \ln(p(x_k|j_k;\vartheta)P_{jk})\right] =$$

$$\sum_{k=1}^{N} E[\;\;] =$$

$$\sum_{k=1}^{N}\sum_{j_k=1}^{J} P\left(j_k|x_k;\vartheta(t)\right) \ln\left(p(x_k|j_k;\vartheta)P_{jk}\right)$$

Βήμα Μ: $$\frac{\partial Q}{\partial \vartheta} = 0, \; \frac{\partial Q}{\partial P_{jk}} = 0, \quad j_k = 1,2,\ldots,J$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

$$P(j|x_k;\hat{\Theta}(t)) = \frac{p(x_k|j;\vartheta(t))P_j}{P(x_k;\hat{\Theta}(t))}$$

$$p(x_k;\hat{\Theta}(t)) = \sum_{j=1}^{J} p(x_k|j;\vartheta(t))P_j$$

Για παράδειγμα, αν οι εμπλεκόμενες κατανομές είναι κανονικές:

$$p(\boldsymbol{x}_k|j;\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_j^2)^{l/2}} \exp\left(-\frac{\|\boldsymbol{x}_k - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right)$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

Βήμα Ε:

$$Q(\Theta;\Theta(t))=\sum_{k=1}^{N}\sum_{j=1}^{J}P(j|\boldsymbol{x}_k;\Theta(t))\left(-\frac{l}{2}\ln\sigma_j^2-\frac{1}{2\sigma_j^2}\|\boldsymbol{x}_k-\boldsymbol{\mu}_j\|^2+\ln P_j\right)$$

Βήμα Μ:

$$\boldsymbol{\mu}_j(t+1)=\frac{\sum_{k=1}^{N}P(j|\boldsymbol{x}_k;\Theta(t))\boldsymbol{x}_k}{\sum_{k=1}^{N}P(j|\boldsymbol{x}_k;\Theta(t))}$$

$$\sigma_j^2(t+1)=\frac{\sum_{k=1}^{N}P(j|\boldsymbol{x}_k;\Theta(t))\|\boldsymbol{x}_k-\boldsymbol{\mu}_j(t+1)\|^2}{l\sum_{k=1}^{N}P(j|\boldsymbol{x}_k;\Theta(t))}$$

$$P_j(t+1)=\frac{1}{N}\sum_{k=1}^{N}P(j|\boldsymbol{x}_k;\Theta(t))$$

Ταξινομητές Bayes/εκτίμηση παραμέτρων

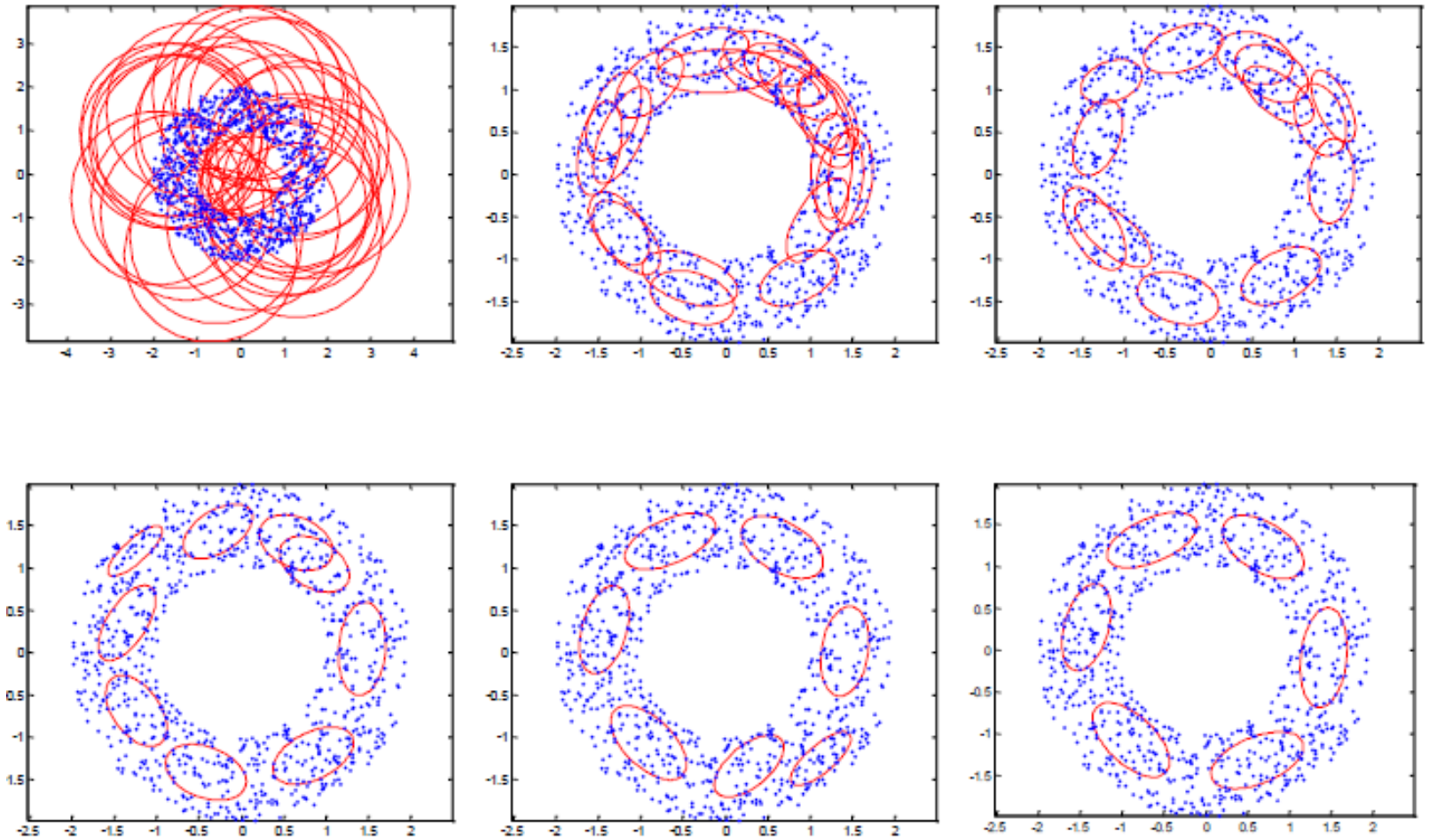
Πρακτικά:

- Το j δεν το ξέρουμε. Στην αναγνώριση με επίβλεψη, το σύνολο εκμάθησης μπορεί να μας βοηθήσει.
- Ξεκινάμε την επανάληψη με κατάλληλη αρχική εκτίμηση $\vartheta(0)$ $(\sum P_j = 1)$.
- Σταματάμε την επανάληψη όταν $\|\vartheta(t) - \vartheta(t+1)\| \leq \varepsilon$, για κατάλληλη νόρμα και ε.
- Ο αλγόριθμος συγκλίνει **πάντα** σε (τοπικό;) μέγιστο της συνάρτησης πιθανοφάνειας.
- Συνιστώσες με $P_i < \delta$ μπορούν να παραλειφθούν σε επόμενα βήματα.

Ταξινομητές Bayes/εκτίμηση παραμέτρων

900 σημεία από ομοιόμορφη κατανομή εντός δακτυλίου ακτινών 1, 2.

30 κανονικές με μέσες τιμές τυχαία σημεία του δείγματος και διαγώνιες συνδιασπορές με διασπορά πολύ μεγαλύτερη του δείγματος



CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

Ταξινομητές Bayes/εκτίμηση παραμέτρων

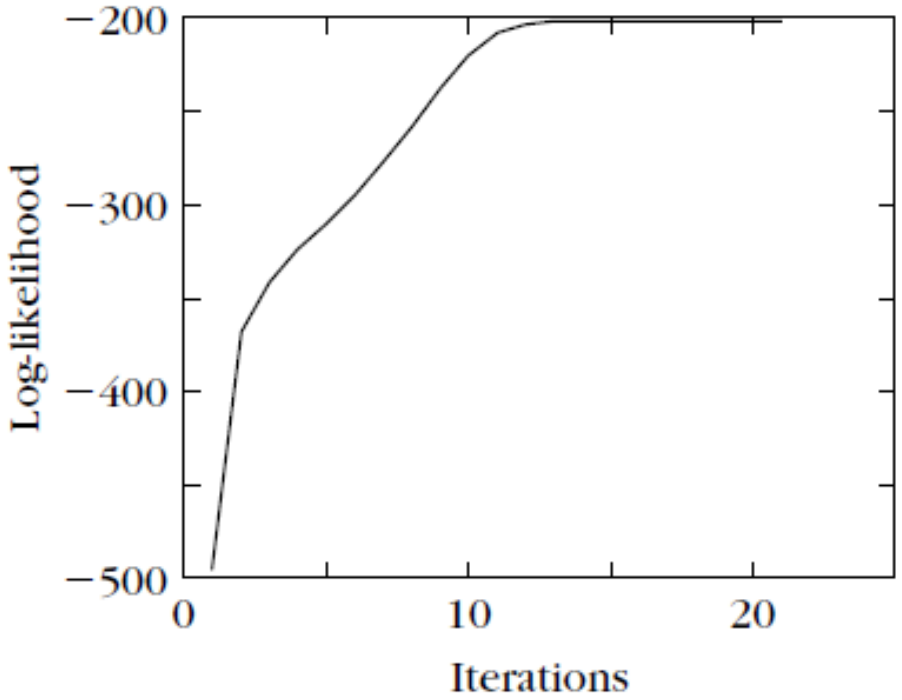
Αρχική: $\boldsymbol{\mu}_1(0) = [1.37,\ 1.20]^T, \quad \boldsymbol{\mu}_2(0) = [1.81,\ 1.62]^T, \quad \sigma_1^2 = \sigma_2^2 = 0.44, \quad P = 0.5$

Τελική: $\boldsymbol{\mu}_1 = [1.05,\ 1.03]^T,\ \boldsymbol{\mu}_2 = [1.90,\ 2.08]^T,\ \sigma_1^2 = 0.10,\ \sigma_2^2 = 0.06,\ P = 0.844$

Πραγματική: P=0,8

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}$$

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$$



©Theodoridis, Koutroumbas

Εισαγωγή

Αν τα δεδομένα μας δεν ακολουθούν κανονική κατανομή μπορούμε να χρησιμοποιήσουμε μη παραμετρικές μεθόδους για την εκτίμηση των εμπλεκομένων συναρτήσεων.

(μπορούμε βέβαια να χρησιμοποιήσουμε τις μεθόδους που ακολουθούν ακόμη και για κανονικές κατανομές).

Οι τεχνικές αυτές βασίζονται εν πολλοίς στην προσέγγιση άγνωστων συναρτήσεων πυκνότητας κατανομής από **ιστογράμματα**.

4. Μη παραμετρικές μέθοδοι

Η ιδέα στην οποία βασίζονται όλοι οι αλγόριθμοι είναι:
Η πιθανότητα του διανύσματος x, που ακολουθεί τη κατανομή $p(x)$, να ανήκει στο σύνολο R, είναι,

$$P(x \in R) = \int_R p(x)\mathrm{d}x \approx p(x) \cdot V$$

όπου V ο όγκος που περιβάλλει το R.

Επίσης, η πιθανότητα το δείγμα x να περιέχεται σε κάποια περιοχή προσεγγίζεται από το λόγο συχνότητας,

$$P(x \in \mathrm{R}) \approx k / N$$

όπου k είναι το πλήθος των δειγμάτων που περιέχονται στην περιοχή και N το συνολικό πλήθος. Επομένως,

$$\boldsymbol{p(x)V \approx k/N \text{ ή } p(x) \approx k/NV}$$

4. Μη παραμετρικές μέθοδοι

Η προσέγγιση βελτιώνεται όσο μεγαλώνει το πλήθος N και συρρικνώνεται ο όγκος V. Στη πράξη το N είναι δεδομένο, άρα θεωρητικά θα μπορούσαμε να μειώσουμε το V. Όμως αυτό δε μπορεί να γίνει γιατί στο τέλος το R θα είναι τόσο μικρό που δεν θα περιέχει δείγματα της κατανομής. Επομένως, το V πρέπει να είναι,

1. Αρκετά μεγάλο ώστε να περιέχει αρκετά παραδείγματα από τη κατανομή.
2. Αρκετά μικρό ώστε η $p(x)$ να είναι σταθερή στο R.

4. Μη παραμετρικές μέθοδοι

Συνοψίζοντας,

$$\hat{p}(x) \approx \frac{k}{NV}$$

όπου,

V ο όγκος της περιοχής γύρω από το x
N το συνολικό πλήθος δειγμάτων
k το πλήθος των δειγμάτων εντός του V.

Δύο προσεγγίσεις:

1. Προκαθορίζουμε το V και εκτιμάμε το k από τα δεδομένα: **εκτίμηση πυκνότητας πυρήνα** (KDE).
2. Προκαθορίζουμε το k και εκτιμάμε το V από τα δεδομένα: **k πλησιέστερος γείτονας** (k-NN).

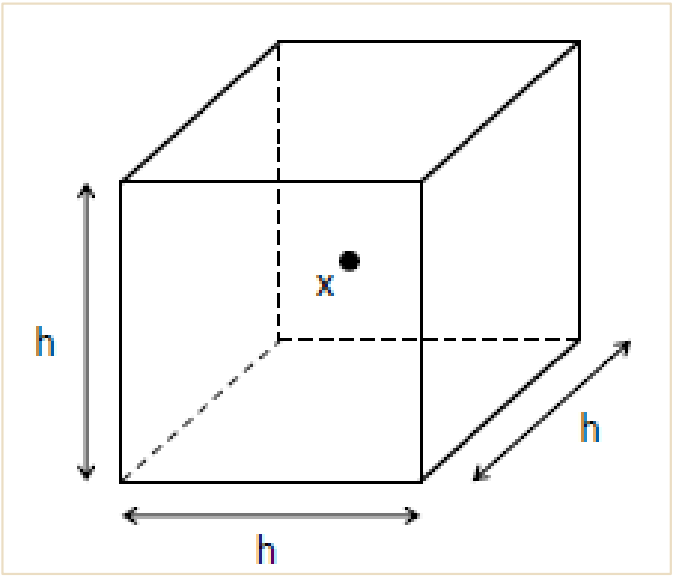
Αποδεικνύεται ότι και στις δύο περιπτώσεις η πραγματική σππ προσεγγίζεται καθώς το $N \to \infty$, αν ο όγκος V συρρικνώνεται και το k αυξάνεται.

1. KDE

Ορίζουμε τη **συνάρτηση πυρήνα** ως,

$$\varphi(x) = \begin{cases} 1 & |x| \le \frac{1}{2} \\ 0 & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

γνωστή και ως **παράθυρο Parzen**.



Η $\varphi\left(\frac{x_i - x}{h}\right)$ είναι 1 για κάθε σημείο x_i μέσα στον υπερκύβο πλευράς h και κέντρο το x και μηδέν αλλού.

4. Μη παραμετρικές μέθοδοι

Το πλήθος των σημείων εντός του υπερκύβου είναι,

$$k = \sum_{i=1}^{N} \varphi\left(\frac{x_i - x}{h}\right)$$

Άρα,

$$\hat{p}(x) \approx \frac{k}{NV} = \frac{\sum_{i=1}^{N} \varphi\left(\frac{x_i - x}{h}\right)}{Nh^q}$$

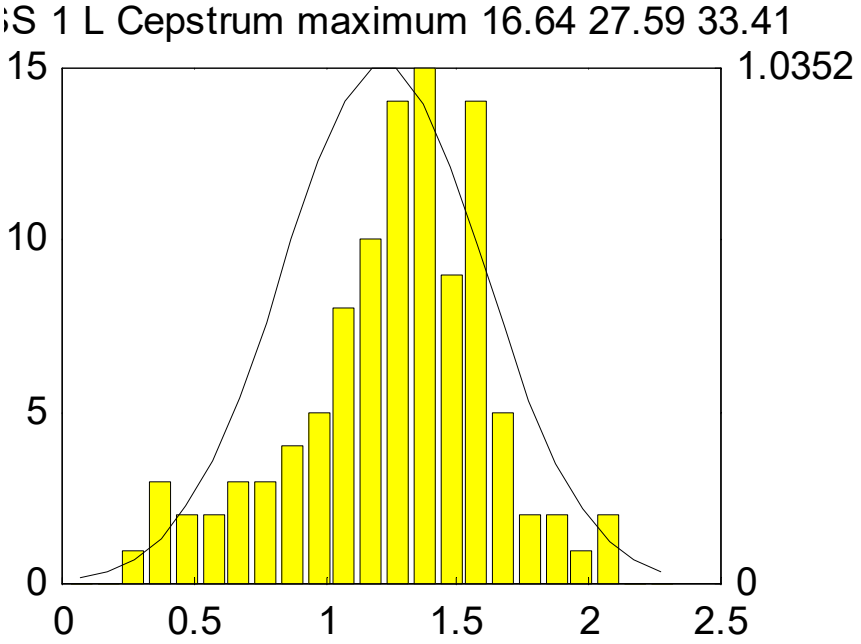
q = διάσταση υπερχώρου (χαρακτηριστικών).

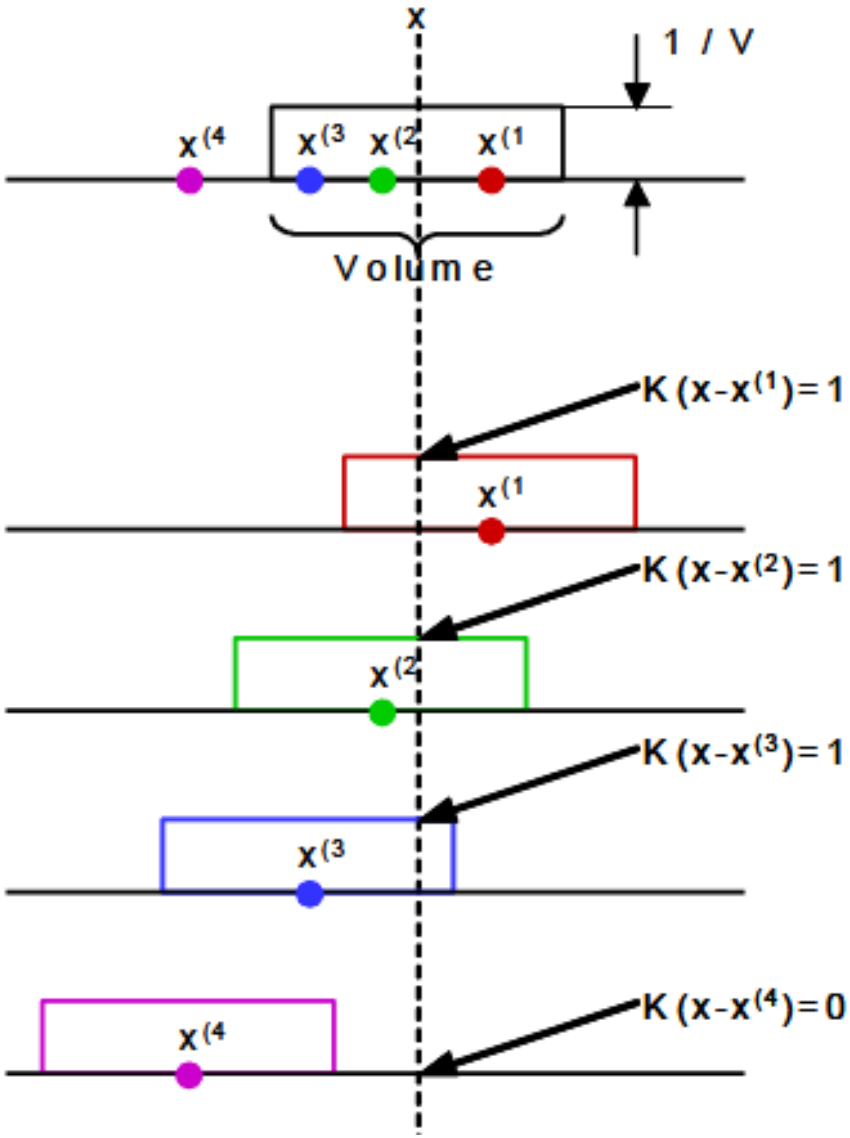
4. Μη παραμετρικές μέθοδοι

Μονοδιάστατα χαρακτηριστικά: χωρίζουμε τον άξονα σε περιοχές (bins) εύρους ***h***. Τότε,

$$\hat{p}(x) \approx \frac{1}{h}\frac{k}{N}, \left|x - \hat{x}\right| \le \frac{h}{2}$$

όπου $\hat{x}$ είναι το μέσο της περιοχής.





4. Μη παραμετρικές μέθοδοι

Πρόβλημα: η $p(x)$ πρέπει να είναι **συνεχής** ενώ η $\varphi(x)$ όπως έχει ορισθεί **ασυνεχής**.

Λύση: συναρτήσεις $\varphi(x)$ γνωστές ως **παράθυρα Parzen** που να ικανοποιούν

$$\varphi(x) \geq 0, \quad \int_x \varphi(x)\mathrm{d}x = 1$$

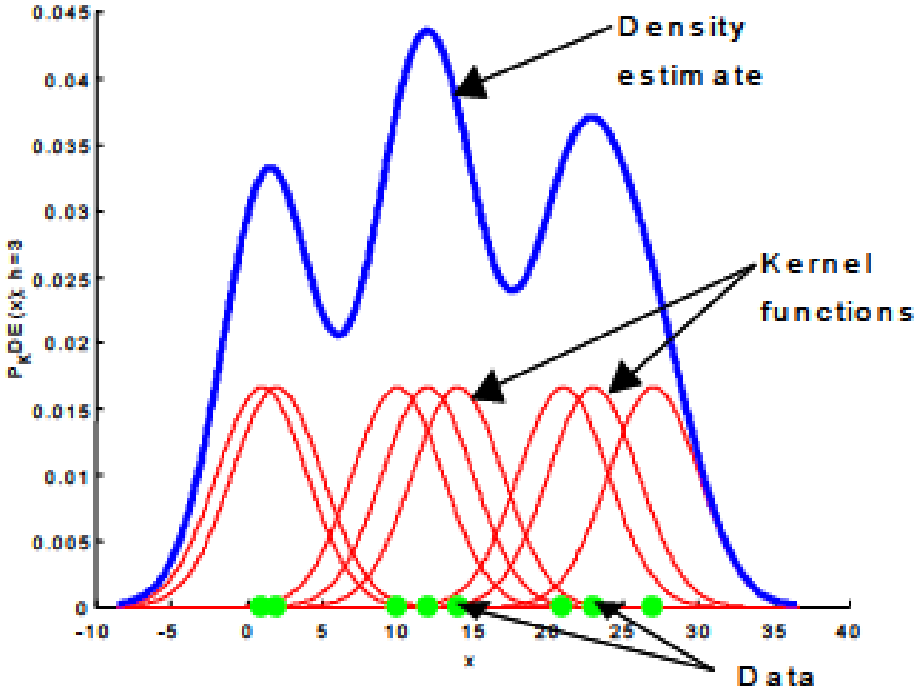
Συνηθέστερη περίπτωση είναι συμμετρικές, unimodal κατανομές όπως η πρότυπη κανονική,

$$\varphi(x) = \frac{1}{(2\pi)^{q/2}} \exp\left\{-\frac{x^T x}{2}\right\}$$

$$\Rightarrow \hat{p}(x) = \frac{1}{h^q}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{(2\pi)^{q/2}} \exp\left\{\frac{(x-x_i)^T(x-x_i)}{2h^2}\right\}$$

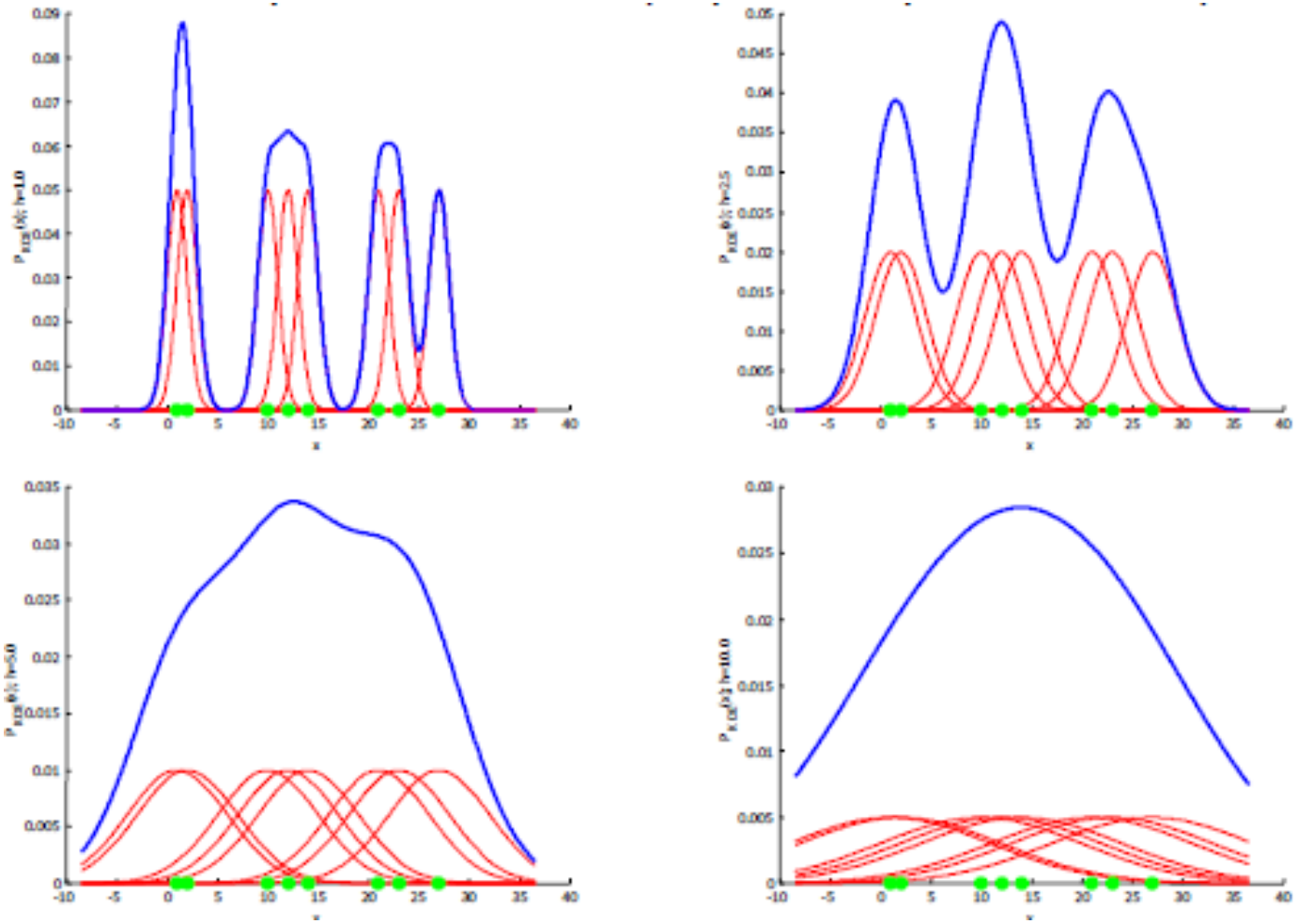
4. Μη παραμετρικές μέθοδοι

Δηλαδή, η άγνωστη σππ προσεγγίζεται ως ο μέσος όρος *N* κανονικών συναρτήσεων, που κάθε μία από αυτές έχει ως κέντρο ένα διαφορετικό σημείο του συνόλου εκπαίδευσης.



Η άγνωστη σππ προσεγγίζεται ως ένα σύνολο «λοφίσκων». Το *h* παίζει το ρόλο παράγοντα εξομάλυνσης.

4. Μη παραμετρικές μέθοδοι



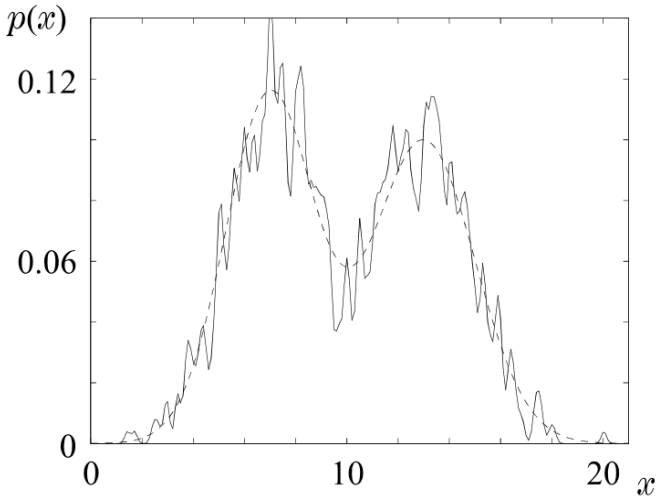
Μικρό *h*: μυτερές κατανομές, δύσκολη ερμηνεία

Μεγάλο *h*: υπερεξομάλυνση

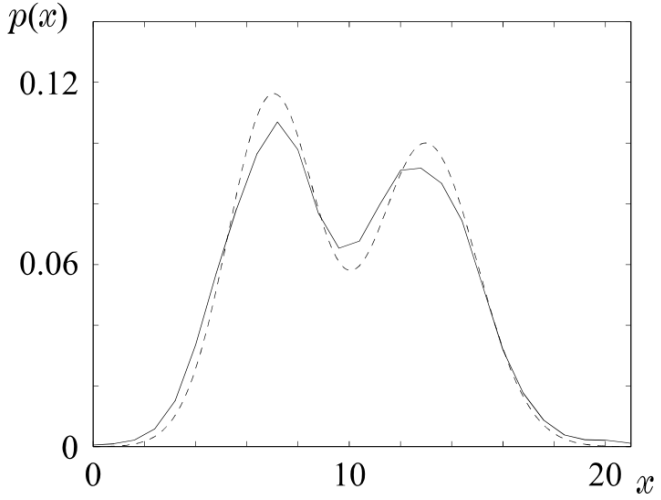
4. Μη παραμετρικές μέθοδοι

Μειώνοντας το h (με N σταθερό), **αυξάνεται** η διασπορά!

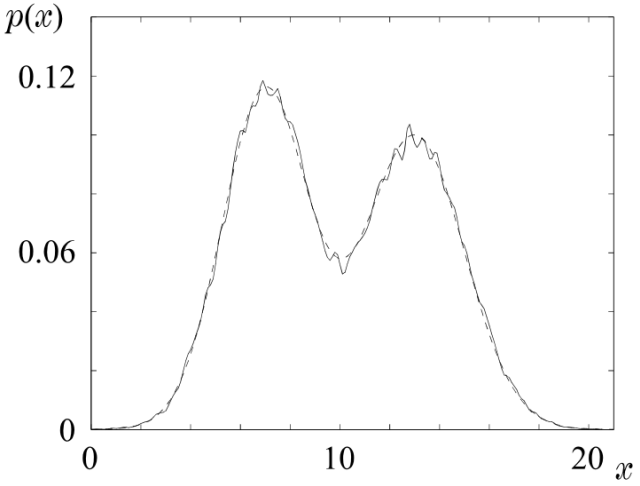
h=0,1· N=1000



h=0,8· N=1000



h=0,1· N=10000



Αυξάνοντας το N (με h σταθερό), **μειώνεται η διασπορά και αυξάνει η ακρίβεια**.

4. Μη παραμετρικές μέθοδοι

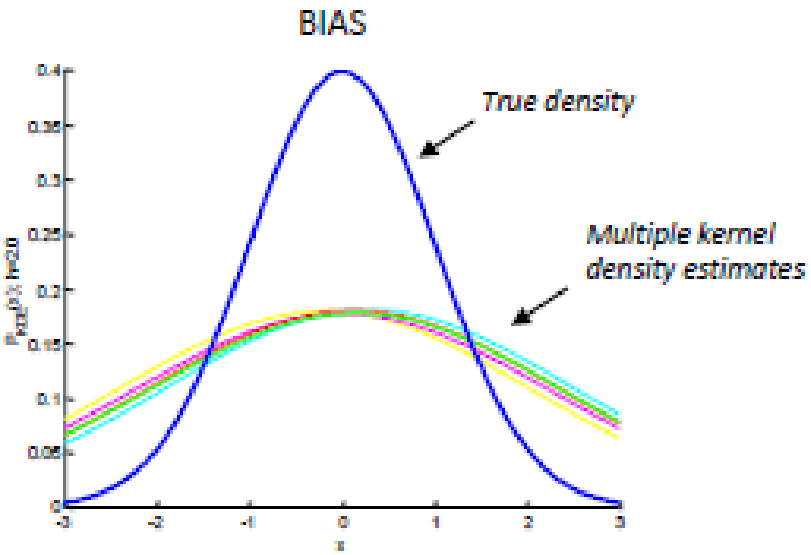
Πως εκτιμούμε το «βέλτιστο» *h*;
Θεωρώντας ως κριτήριο το μέσο τετραγωνικό σφάλμα εκτίμησης της σππ,

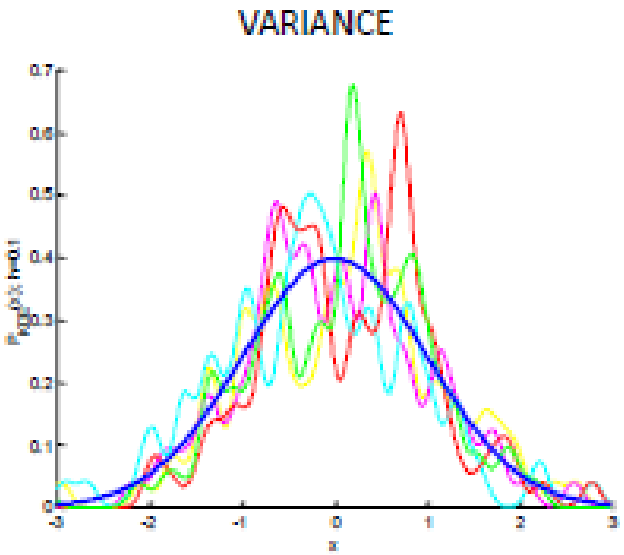
$$E\left[\left(\hat{p}(x)-p(x)\right)^2\right]=\underbrace{E\left[\hat{p}(x)-p(x)\right]^2}_{bias}+\underbrace{\mathrm{var}(\hat{p}(x))}_{\text{διασπορά}}$$

Η **μεροληψία** είναι το συστηματικό σφάλμα της εκτίμησης.
Η **διασπορά** είναι το τυχαίο σφάλμα.

4. Μη παραμετρικές μέθοδοι

Μεγάλο *h* ελαττώνει τις διαφορές μεταξύ των εκτιμήσεων σε διαφορετικά σύνολα εκμάθησης (διασπορά), αλλά αυξάνει την μεροληψία.
Μικρό *h*, έχει το αντίθετο αποτέλεσμα.





4. Μη παραμετρικές μέθοδοι

Η «εύλογη» προσέγγιση της προσομοίωσης δεν αποδίδει σε προβλήματα πολλών διαστάσεων. Αν υποθέσουμε ότι η πραγματική κατανομή είναι κανονική, και χρησιμοποιήσουμε κανονικούς πυρήνες τότε αποδεικνύεται ότι,

$$h^* = 1{,}06\ \sigma\ N^{-1/5}$$

4. Μη παραμετρικές μέθοδοι

Υπάρχουν κάποια «προβληματάκια» σε χαρακτηριστικά πολλών διαστάσεων, καθώς το *h* είναι κοινό σε κάθε διάσταση. Μία λύση είναι η «προλεύκανση» (pre-whitening) των δεδομένων, δηλαδή ο μετασχηματισμός τους έτσι ώστε Σ=I. Ο μετασχηματισμός αυτός είναι,

$$y=\Lambda^{-1/2}M^{T}x$$

όπου Λ και *M* είναι οι πίνακες ιδιοτιμών και ιδιοδιανυσμάτων του Σ.

Εναλλακτικά μπορούμε να χρησιμοποιήσουμε, γινόμενο πυρήνων

$$\hat{p}(x)\approx\frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{h_1\cdots h_q}\prod_{j=1}^{q}\varphi\left(\frac{x_i-x}{h_j}\right)\right\}$$

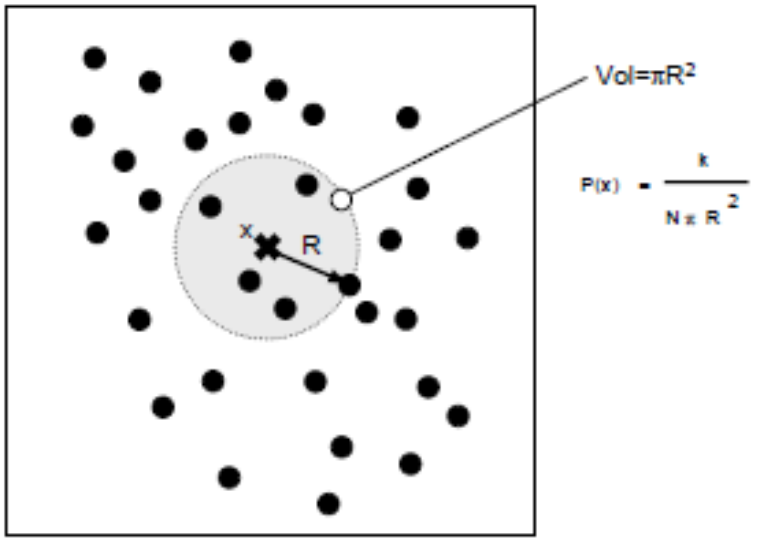
Λευκή διαδικασία: έχει την ίδια ισχύ σε όλες τις συχνότητες όπως το λευκό φως.

Η μέθοδος των πλησιέστερων γειτόνων

1. "Ογκομετρικός» ταξινομητής

Στη μέθοδο αυτή, επιλέγουμε μία τιμή για το k (πλήθος σημείων) και αυξάνουμε την περιοχή γύρω από το σημείο x μέχρι να περικλείσει k δεδομένα.

$$\hat{p}(x) = \frac{k}{NV(x)} = \frac{k}{Nc_q R_k^q(x)}$$



$\mathbf{c_q}$: όγκος υπερσφαίρας ακτίνας 1 σε q διαστάσεις ($c_3=4\pi/3$)

$R_k(x)$: απόσταση του σημείου x από τον k-οστό πλησιέστερο γείτονα του

Όγκος υπερσφαίρας q διαστάσεων, ακτίνας r :

$$V_q = \begin{cases} r^q \dfrac{\pi^{\frac{q}{2}}}{\left(\frac{q}{2}\right)!} & q \text{ άρτιος} \\ r^q 2^q \pi^{\frac{q-1}{2}} \dfrac{\left(\frac{q-1}{2}\right)!}{q!} & q \text{ περιττός} \end{cases}$$

4. Μη παραμετρικές μέθοδοι

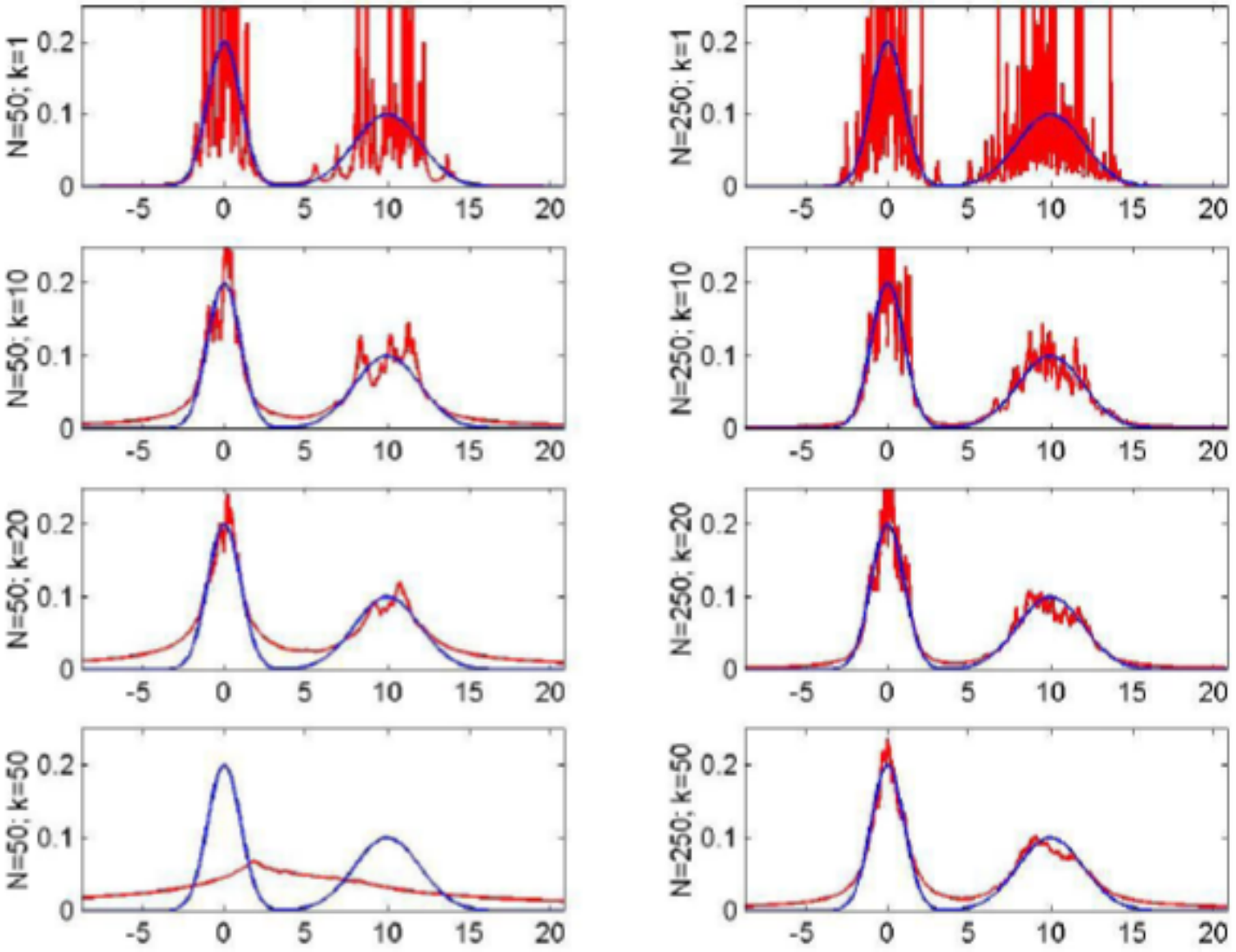
Η μέθοδος αυτή έχει διάφορα μειονεκτήματα:

- Η εκτίμηση έχει ασυνέχειες (λόγω της $R(x)$)
- Δεν είναι αυστηρά κατανομή γιατί το ολοκλήρωμα της αποκλίνει.

Παρ' όλ' αυτά, αποδεικνύεται ότι συγκλίνει στη πραγματική καθώς το $k \to \infty$, $N \to \infty$, $k/N \to 0$.

4. Μη παραμετρικές μέθοδοι

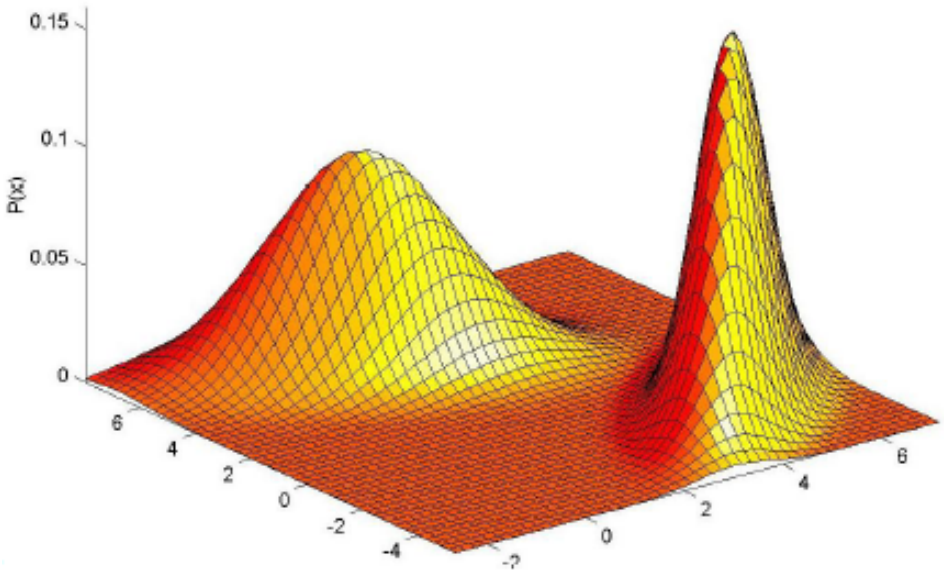
$p(x) = ½N(0,1) + ½N(10,4)$ (διτροπική κανονική)

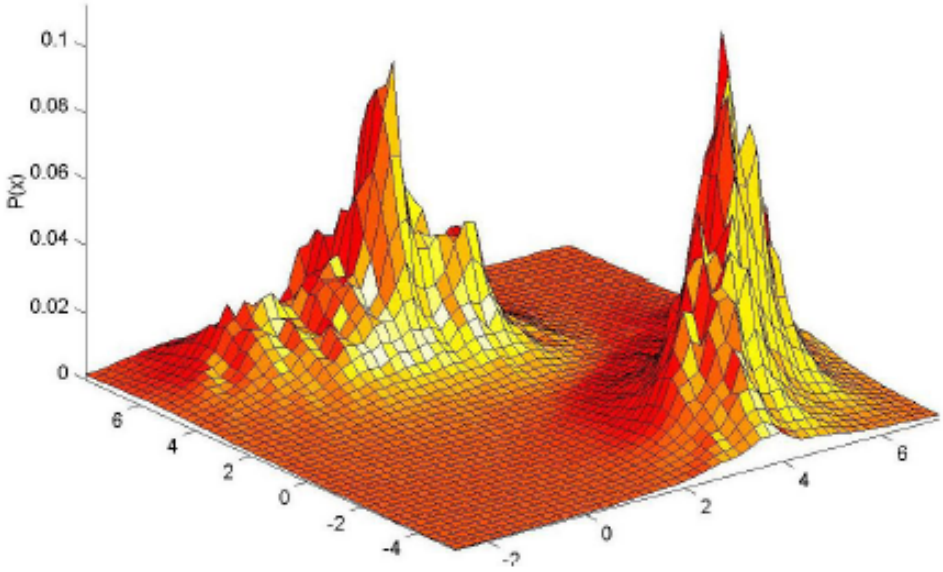


4. Μη παραμετρικές μέθοδοι

$$p(x) = \frac{1}{2}N(\mu_1, \Sigma_1) + \frac{1}{2}N(\mu_2, \Sigma_2)$$

$$with \begin{cases} \mu_1 = [0\ 5]^T & \Sigma_1 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \\ \mu_2 = [5\ 0]^T & \Sigma_2 = \begin{bmatrix} 1 & -1 \\ -1 & 4 \end{bmatrix} \end{cases}$$





k=10, *N*=200

Χρήση των k-NN, στην αναγνώριση προτύπων

$$x \rightarrow \omega_i : P(\omega_i | x) \quad \max$$

$$\hat{p}(x) = \frac{k}{NV(x)} = \frac{k}{Nc_q R_k^q(x)} \qquad P(\omega_i | x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$$

$$P(\omega_i | x) \propto \hat{p}(x|\omega_i)\hat{P}(\omega_i) = \frac{k_i}{N_i c_q R_{k_i}^q(x)} \frac{N_i}{N}$$

$$x \rightarrow \omega_i : \frac{k_i}{R_{k_i}^q(x)} \quad \max, \text{ή} \quad x \rightarrow \omega_i : \frac{1}{r_k(x)} \quad \max$$

4. Μη παραμετρικές μέθοδοι

Δηλαδή:

- υπολογίζουμε την απόσταση d του υπό ταξινόμηση χαρακτηριστικού x από όλα τα σημεία του συνόλου εκμάθησης
- Υπολογίζουμε τους όγκους $V_i(x)$ του υπερστερεού με κέντρο το x που περιέχει τα εγγύτερα k σημεία από την ω_i
- Ταξινομούμε το x στη κλάση i με τη μικρότερη ακτίνα

(ισχύουν για ίδια k_i – δεν είναι απαραίτητο)

Η μέθοδος των πλησιέστερων γειτόνων

2. «Ψηφοφορικός» ταξινομητής

Η μέθοδος αυτή εμπνέεται από την έννοια των πλησιέστερων γειτόνων, αλλά δεν μπορεί να συμπεριληφθεί στην κατηγορία των προσεγγίσεων κατανομών:

- **Δεδομένου νέου *x*, προσδιορίζουμε τα *k* «πλησιέστερα»**
- **Από αυτά τα *k* προσδιορίζουμε τα k_i που ανήκουν στις ω_i**
- **Ταξινομούμε το *x* στην κλάση για την οποία k_i max**

• **Απλούστερη περίπτωση *k*=1 (1NN): κανόνας του πλησιέστερου γείτονα**

4. Μη παραμετρικές μέθοδοι

Ο απλός αυτός ταξινομητής έχει εντυπωσιακή απόδοση:

- Σχεδόν βέλτιστος καθώς $N \to \infty$

$$P_B(e) < P_{1NN}(e) < 2 - \frac{q}{q-1} P_B(e) < 2P_B(e)$$

$P_B(e)$: πιθανότητα σφάλματος βέλτιστου Bayes

- Αυξανομένου του k, βελτιώνεται η απόδοση

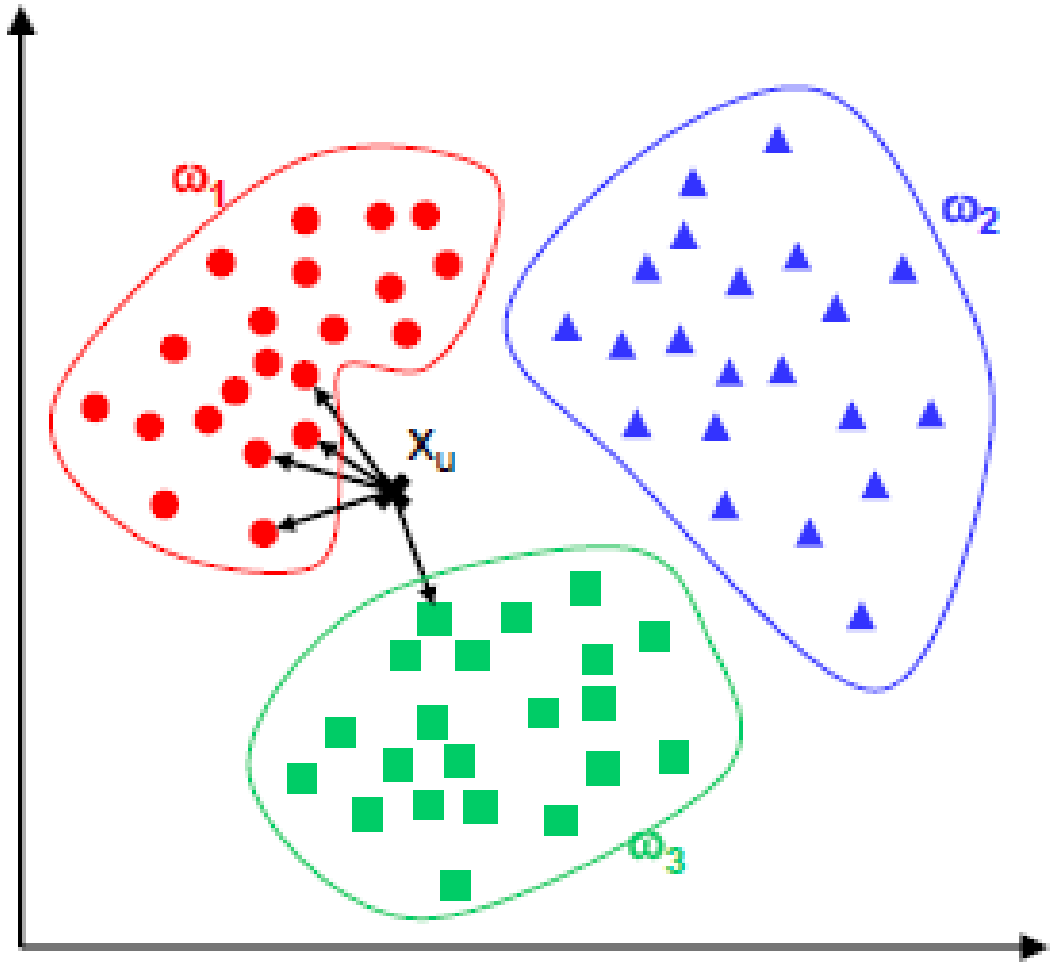
$$P_B(e) \leq P_{kNN}(e) \leq P_B(e) + \sqrt{\frac{2P_{NN}(e)}{k}}$$

4. Μη παραμετρικές μέθοδοι

Σχόλια:

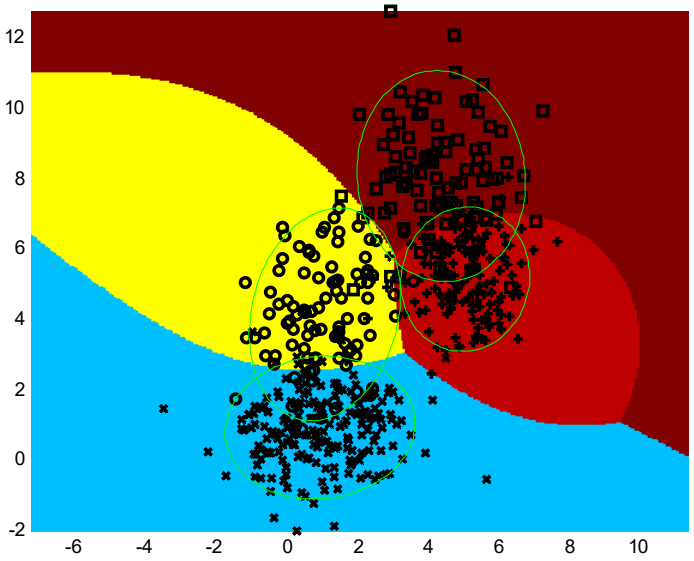
- Σε μικρά δείγματα μεγάλα k πιθανόν να συμπεριφέρονται χειρότερα από μικρότερα
- Η εξαντλητική αναζήτηση γειτόνων είναι της τάξης $(kN)^2$. Πιθανή αντιμετώπιση με την υιοθέτηση αποτελεσματικών μεθόδων αναζήτησης
- Η απόδοση φθίνει δραματικά μειούμενου του N. Πιθανή αντιμετώπιση η χρήση βελτιστοποιημένων μέτρων απόστασης.

4. Μη παραμετρικές μέθοδοι

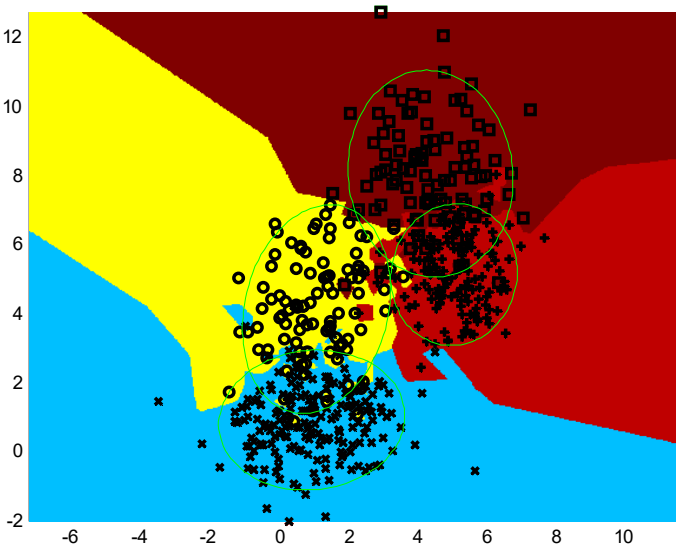


Ευκλείδεια απόσταση, k=5, i_{max}=1

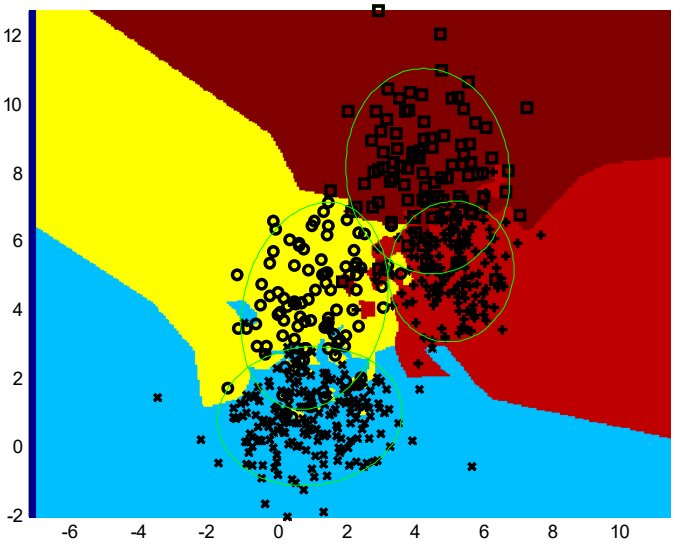
4. Μη παραμετρικές μέθοδοι



Κανονική κατανομή



1-NN



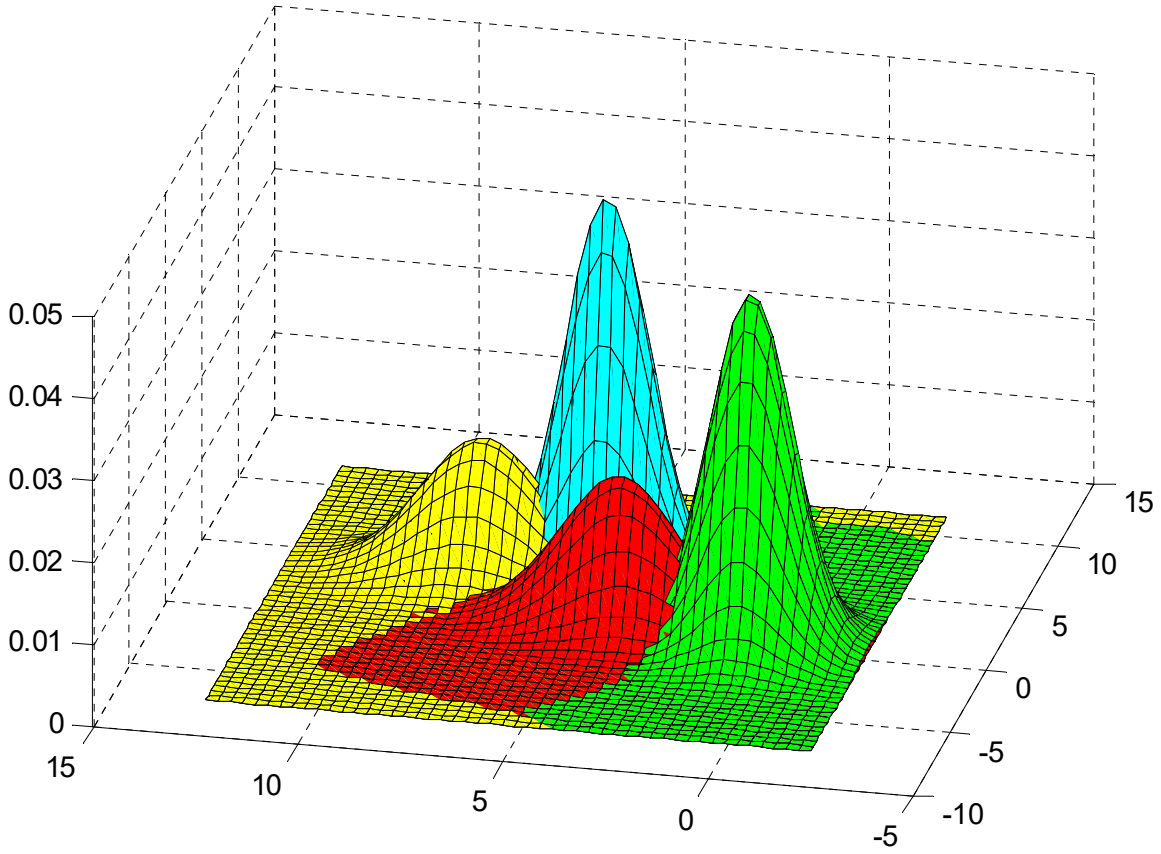
5-NN

$$N_1 = 200$$
$$N_2 = 10$$
$$N_3 = 150$$
$$N_4 = 100$$

$$m_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, m_2 = \begin{bmatrix} 1,2 \\ 4 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

$$m_3 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, m_4 = \begin{bmatrix} 4,5 \\ 8 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

4. Μη παραμετρικές μέθοδοι



$p(x|\omega_i)$

5. Γραμμικοί ταξινομητές

Στους ταξινομητές Bayes συναντήσαμε επιφάνειες (συναρτήσεις) διάκρισης της μορφής,

$$g_i(x) = w_i^T x + w_{io}$$

$$w_i = \Sigma^{-1} m_i, \quad w_{io} = -\frac{1}{2} m_i^T \Sigma^{-1} m_i + \ln P(\omega_i)$$

στη περίπτωση που $\Sigma = \Sigma_i$. Δηλαδή **γραμμικές.**

Το αποτέλεσμα αυτό μπορούμε να το χρησιμοποιήσουμε ανεξάρτητα από τις υποκείμενες κατανομές, με σκοπό τη κατασκευή **υποβέλτιστων** μεν, αλλά **απλών** και υπολογιστικά ελκυστικών ταξινομητών.

5. Γραμμικοί ταξινομητές

Το πρόβλημα:

Δοθέντος ενός *συνόλου εκπαίδευσης*,

$$S=\{x(i),\ y_i\}, i=1,\ \ldots,\ N$$

να βρεθούν *βάρη*,

$$w_i,\ i=0,\ \ldots,\ q$$

της (γραμμικής ως προς x) συνάρτησης,

$$g(x)=w^{\mathrm{T}}x+w_0$$

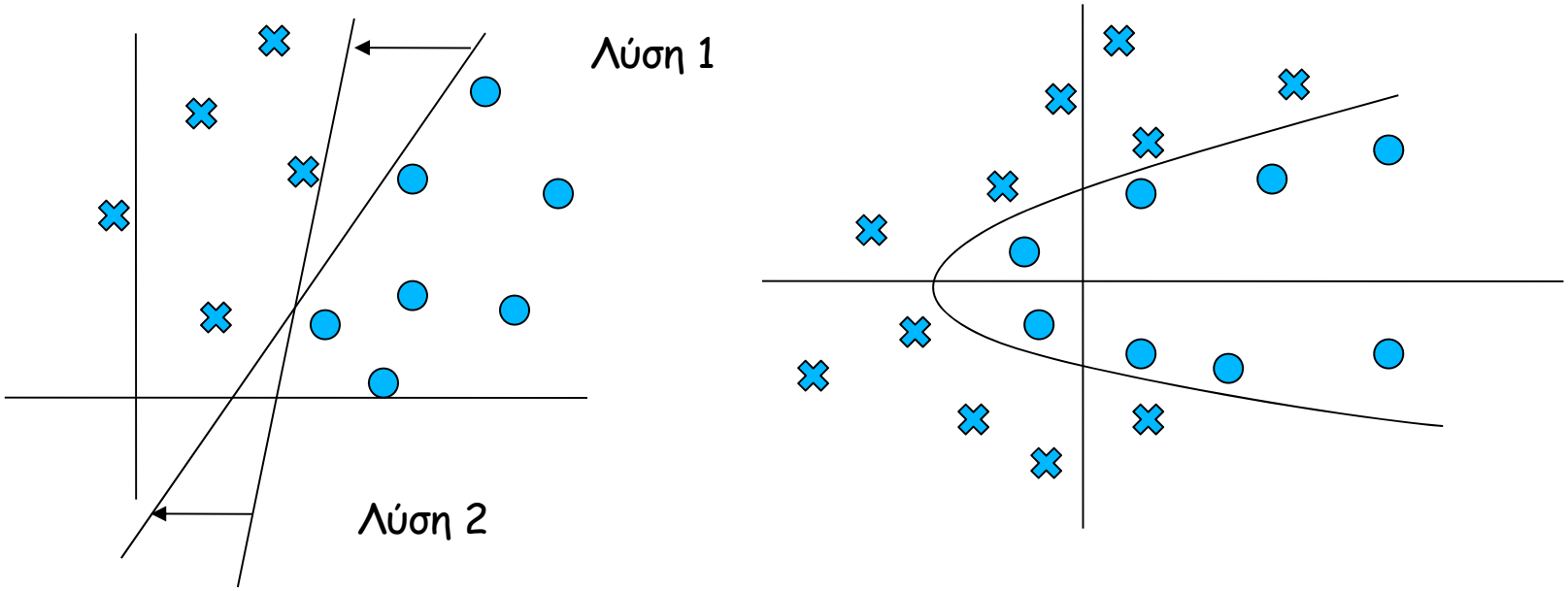
έτσι ώστε να ελαχιστοποιείται το κριτήριο απόδοσης,

$$J(w, y_i)$$

(στην ουσία να ταξινομούνται «σωστά» οι κλάσεις)

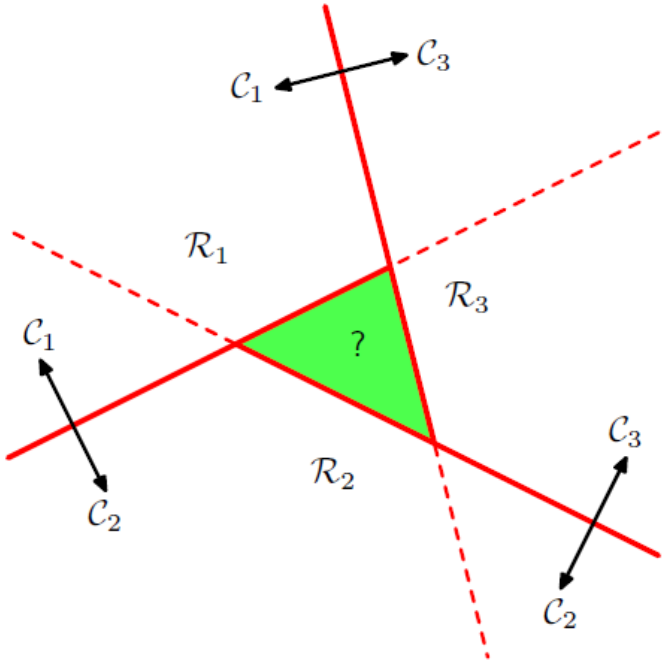
5. Γραμμικοί ταξινομητές

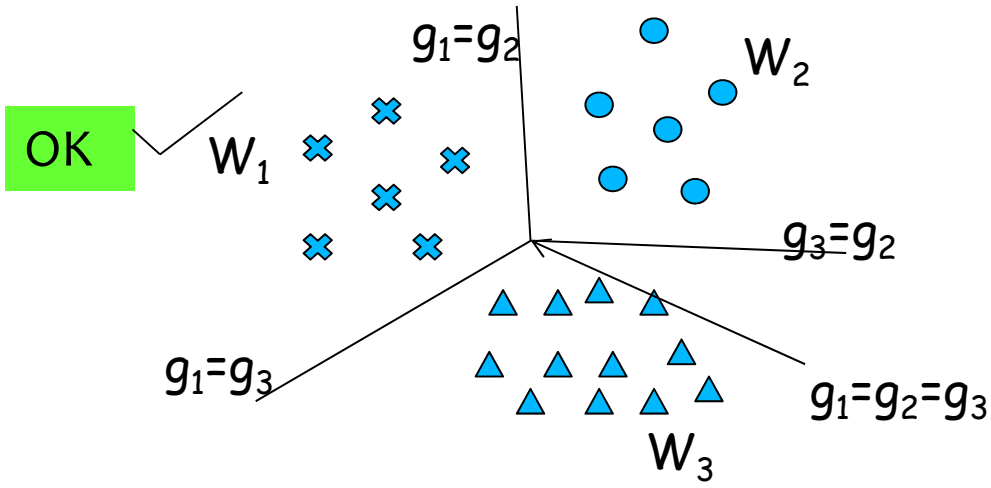
Το πρόβλημα μπορεί να έχει πολλές ή καμία λύση.



5. Γραμμικοί ταξινομητές

Διφορούμενη περιοχή
M(M-1)/2 ταξινομητές





5. Γραμμικοί ταξινομητές

Έστω ένα πρόβλημα δύο κλάσεων με δύο χαρακτηριστικά ($M=2$, $q=2$). Τότε η γραμμική συνάρτηση διάκρισης,

$$g(x) = w^T x + w_0 = w_1 x_1 + w_2 x_2 + w_0$$

ορίζει μία ευθεία στο επίπεδο (η εξίσωση αυτή είναι η γενική μορφή της ευθείας). Αν $x(1)$, $x(2)$ κείνται επί της ευθείας αυτής,

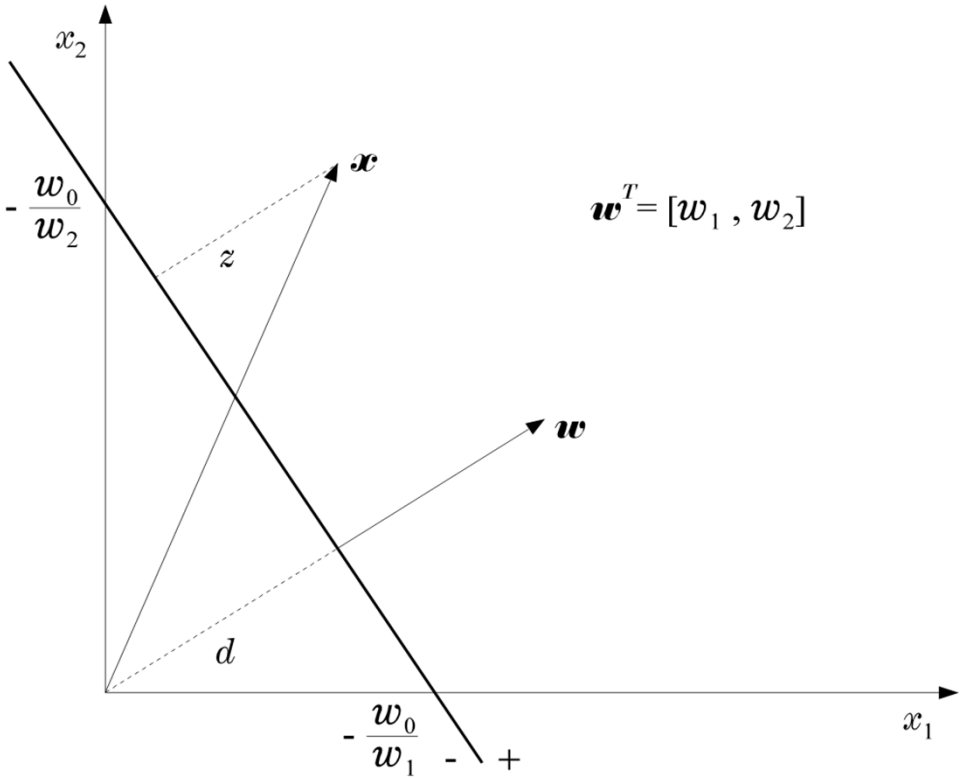
$$w^T x(1) + w_0 = w^T x(2) + w_0 = 0 \Rightarrow$$

$$w^T (x(1) - x(2)) = 0, \quad \forall x(1), x(2)$$

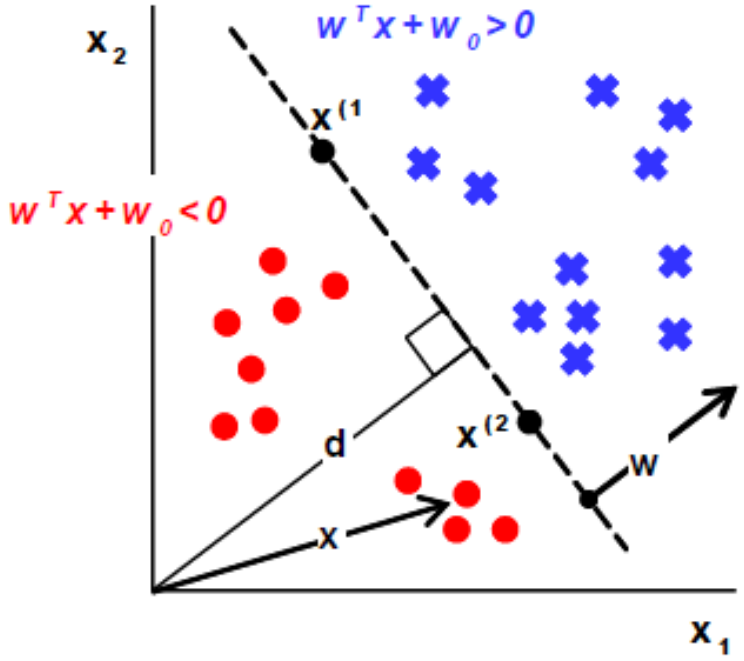
(αλλιώς $w \bullet (x - p) = 0$)

5. Γραμμικοί ταξινομητές

Επομένως το w είναι κάθετο στην ευθεία $g(x) = w^T x + w_o = 0$

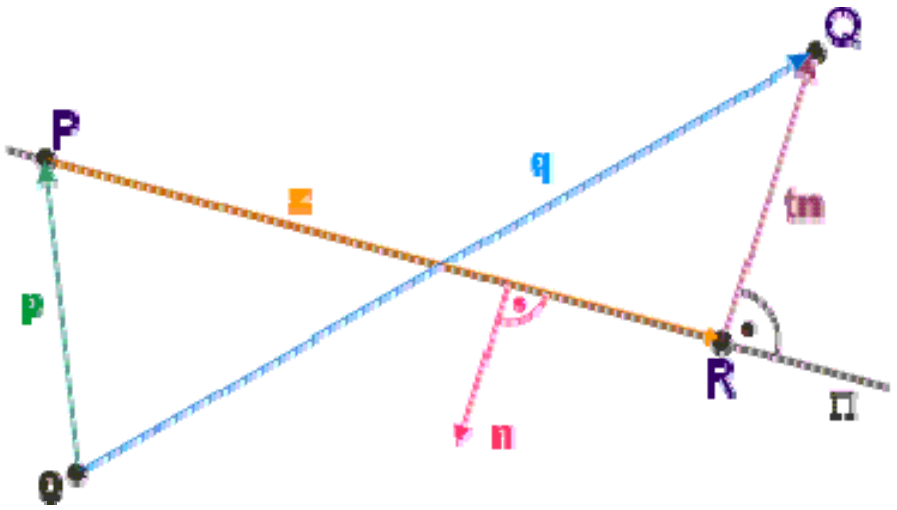


$$\boldsymbol{w}^T = [w_1, w_2]$$



$$z = \frac{|g(x)|}{\sqrt{w_1^2 + w_2^2}}, \quad d = \frac{|g(0)|}{\sqrt{w_1^2 + w_2^2}} = \frac{|w_0|}{\sqrt{w_1^2 + w_2^2}}$$

5. Γραμμικοί ταξινομητές



Έστω το υπερεπίπεδο Π στο R^n που περιγράφεται από την εξίσωση $(x-p).n=0$. Τότε η **απόσταση σημείου Q από το Π** δίνεται από τη σχέση,

$$\text{dist}(Q,\Pi)=\frac{|(q-p)\bullet n|}{|n|}$$

$$q=p+z+tn \Rightarrow tn=q-p-z \Rightarrow$$

$$t|n|^2=(q-p-z)\bullet n=(q-p)\bullet n \Rightarrow$$

$$|t||n|^2=|(q-p)\bullet n| \Rightarrow dist(Q,\Pi)=|tn|=\frac{|(q-p)\bullet n|}{|n|}$$

5. Γραμμικοί ταξινομητές

Έστω $x=[\alpha \ \beta]$,

$$\frac{\left|(q-p)\bullet n\right|}{\left|n\right|}=\frac{\left|\left((\alpha,\beta)-(x_1,x_2)\right)\cdot(w_1,w_2)\right|}{\sqrt{w_1^2+w_2^2}}=$$

$$=\frac{\left|(\alpha-x_1)w_1+(\beta-x_2)w_2\right|}{\sqrt{w_1^2+w_2^2}}=\frac{\left|\alpha w_1+\beta w_2-(x_1w_1+x_2w_2)\right|}{\sqrt{w_1^2+w_2^2}}=$$

$$=\frac{\left|\alpha w_1+\beta w_2-(-w_o)\right|}{\sqrt{w_1^2+w_2^2}}=\frac{\left|g(x)\right|}{\sqrt{w_1^2+w_2^2}}$$

5. Γραμμικοί ταξινομητές

Η επίλυση του συγκεκριμένου προβλήματος γίνεται με πολλούς τρόπους, που διαφοροποιούνται επίσης ανάλογα και με το κριτήριο βελτιστοποίησης. Κάποιοι χρησιμοποιούν δομές **τεχνητών νευρωνικών δικτύων (ANN: artificial neural networks)**, και θα περιγραφούν σε επόμενη ενότητα.

Η συνήθης προσέγγιση είναι να επιλυθεί το πρόβλημα στη περίπτωση δύο κλάσεων, και στη συνέχεια να γενικευθεί. Η γενίκευση οδηγεί στην επίλυση *M* προβλημάτων δύο κλάσεων.

1. Ελάχιστα τετράγωνα σφάλματος ταξινόμησης (MSE)

$$J(w) = \sum_{i=1}^{N} \left(w^T x_i - y_i \right)^2$$

$$\frac{\partial J(w)}{\partial w} = \frac{\partial}{\partial w} \sum_{i=1}^{N} (w^T x_i - y_i)^2 = 0 \Rightarrow$$

$$\left(\sum_{i=1}^{N} x_i x_i^T \right) w = \sum_{i=1}^{N} x_i y_i \Rightarrow$$

$$w = \left(\sum_{i=1}^{N} x_i x_i^T \right)^{-1} \sum_{i=1}^{N} x_i y_i$$

Πιο συνοπτικά,

5. Γραμμικοί ταξινομητές

$$X(N\times(q+1)) = \begin{bmatrix} x_1^T \\ x_2^T \\ \dots \\ x_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1q}\ 1 \\ x_{21} & x_{22} & \cdots & x_{2q}\ 1 \\ \vdots & \vdots & \ddots & \vdots\ \vdots \\ x_{N1} & x_{N1} & \cdots & x_{Nq}\ 1 \end{bmatrix} \quad y(N\times 1) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \; w_i((q+1)\times 1) = \begin{bmatrix} w_{i1} \\ \vdots \\ w_{iq} \\ w_{io} \end{bmatrix}$$

$$y_i \in \{0,1\}$$

$$\hat{w} = (X^T X)^{-1} X^T y$$

$$= X^{\dagger} y$$

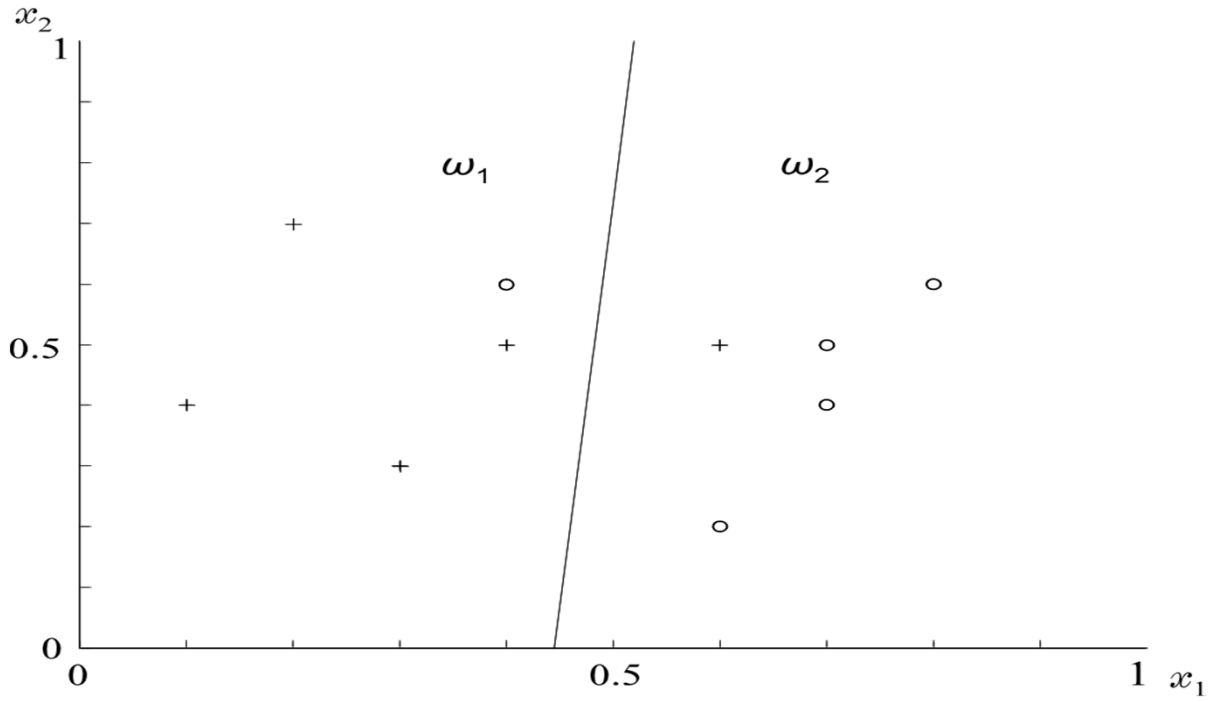
Ο $X^{\dagger}$ καλείται **ψευδοαντίστροφος Moore-Penrose**, και είναι μία γενίκευση του αντιστρόφου για μη τετραγωνικούς πίνακες (προφανώς ισχύει αν ο $(X^T X)^{-1}$ υφίσταται)

5. Γραμμικοί ταξινομητές

Παράδειγμα:
(Θεοδωρίδης)

$$\omega_1 : \begin{bmatrix} 0,4 \\ 0,5 \end{bmatrix}, \begin{bmatrix} 0,6 \\ 0,5 \end{bmatrix}, \begin{bmatrix} 0,1 \\ 0,4 \end{bmatrix}, \begin{bmatrix} 0,2 \\ 0,7 \end{bmatrix}, \begin{bmatrix} 0,3 \\ 0,3 \end{bmatrix}$$

$$\omega_2 : \begin{bmatrix} 0,4 \\ 0,6 \end{bmatrix}, \begin{bmatrix} 0,6 \\ 0,2 \end{bmatrix}, \begin{bmatrix} 0,7 \\ 0,4 \end{bmatrix}, \begin{bmatrix} 0,8 \\ 0,6 \end{bmatrix}, \begin{bmatrix} 0,7 \\ 0,5 \end{bmatrix}$$



$$X = \begin{bmatrix} 0,4 & 0,5 & 1 \\ 0,6 & 0,5 & 1 \\ 0,1 & 0,4 & 1 \\ 0,2 & 0,7 & 1 \\ 0,3 & 0,3 & 1 \\ 0,4 & 0,6 & 1 \\ 0,6 & 0,2 & 1 \\ 0,7 & 0,4 & 1 \\ 0,8 & 0,6 & 1 \\ 0,7 & 0,5 & 1 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

5. Γραμμικοί ταξινομητές

$$X^T X = \begin{bmatrix} 2,8 & 2,24 & 4,8 \\ 2,24 & 2,41 & 4,7 \\ 4,8 & 4,7 & 10 \end{bmatrix}, X^T y = \begin{bmatrix} -1,6 \\ 0,1 \\ 0 \end{bmatrix}$$

$$\hat{w} = (X^T X)^{-1} X^T y = \begin{bmatrix} -3,13 \\ 0,24 \\ 1,34 \end{bmatrix}$$

Οι κλάσεις δεν είναι **γραμμικά** διαχωρίσιμες !

5. Γραμμικοί ταξινομητές

Ο αλγόριθμος MSE είναι απλός, και δίνει πάντα αποτέλεσμα, αλλά έχει ένα μειονέκτημα: δεν εξασφαλίζει τον γραμμικό διαχωρισμό όταν οι κλάσεις είναι γραμμικά διαχωρίσιμες. Το εμπόδιο αυτό υπερπηδάται μέσω του αλγορίθμου **Ho-Kasyap**:

Στο παράδειγμα είδαμε ότι τα y_i είναι {1, -1}, επιλογή εν πολλοίς αυθαίρετη. Η ιδέα του αλγόριθμου Ho-Kasyap είναι ότι αν οι κλάσεις είναι γραμμικά διαχωρίσιμες τότε υπάρχουν w και y τέτοια ώστε,

$$Xw = y > 0$$

(όπου αρνητικά X έχουν αλλάξει πρόσημο).

Επομένως λύνουμε το πρόβλημα ελαχιστοποίησης του κριτηρίου απόδοσης ως προς (w, y).

5. Γραμμικοί ταξινομητές

Άρα πρέπει να λύσουμε το σύστημα:

$$\frac{\partial J(w,y)}{\partial y} = \frac{\partial \|Xw - y\|^2}{\partial y} = 2X^{\mathsf{T}}(Xw - y)$$

$$\frac{\partial J(w,y)}{\partial w} = \frac{\partial \|Xw - y\|^2}{\partial w} = -2(Xw - y)$$

με $\mathbf{y > 0}$.

Αν μπορέσουμε να βρούμε μία εφικτή λύση για τη πρώτη εξίσωση, τότε από τη δεύτερη:

$$\hat{w} = X^{\dagger}\hat{y}$$

5. Γραμμικοί ταξινομητές

Υιοθετώντας προσέγγιση καθόδου προς τη κατεύθυνση της παραγώγου (gradient descent), θα επιλέγαμε:

$$\hat{y}(k+1) = \hat{y}(k) - \eta(k)\frac{\partial J}{\partial y} = \hat{y}(k) + \eta(k)[X\hat{w} - \hat{y}]$$

αλλά επειδή $y > 0$ μηδενίζουμε τις αρνητικές συνιστώσες του αριστερού μέλους:

$$e(k) = X\hat{w}(k) - \hat{y}(k)$$

$$e^{+}(k) = \frac{1}{2}\left(e(k) + |e(k)|\right)$$

$$\hat{y}(k+1) = \hat{y}(k) + \eta(k)e^{+}(k)$$

$$\hat{w}(k+1) = X^{\dagger}\hat{y}(k+1)$$

5. Γραμμικοί ταξινομητές

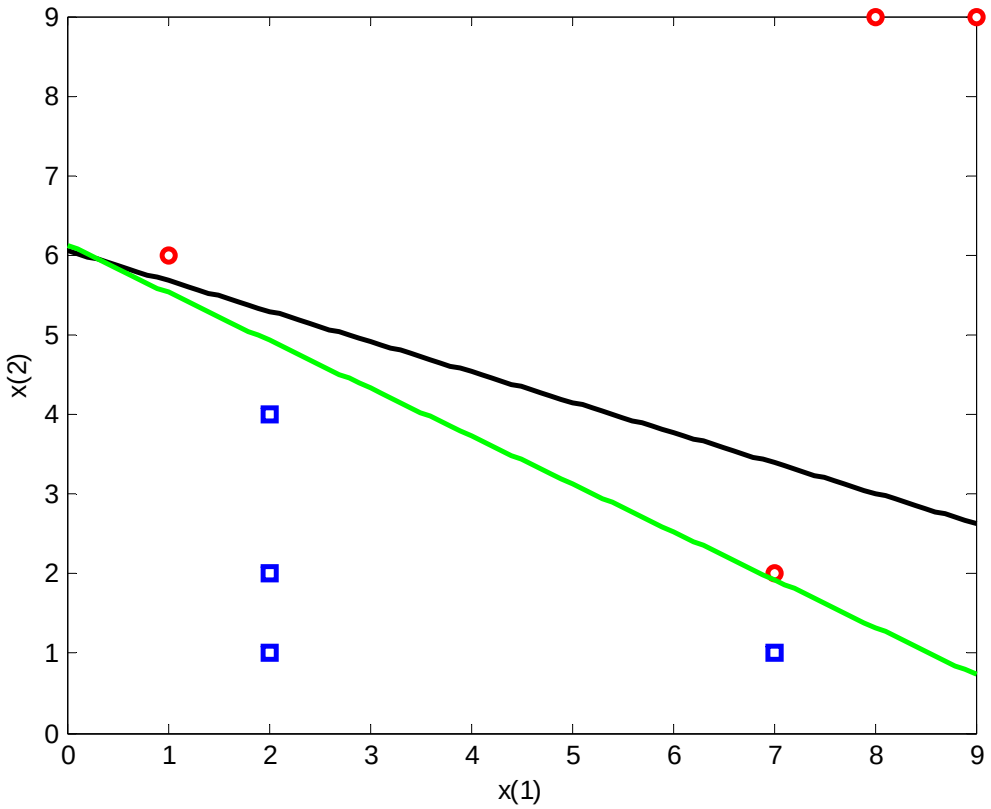
Παρατηρήσεις:

1. Ο αλγόριθμος πάντα συγκλίνει,
   (όταν $\|e(k)\| < \varepsilon$ ή $e(k) > 0$).
2. Ο ψευδοαντίστροφος (που ίσως έχει μεγάλη διάσταση) υπολογίζεται άπαξ.
3. Ο αλγόριθμος επισημαίνει τη μη διαχωρισιμότητα,
   ($e(k) > 0$).
4. Η παράμετρος $\eta(k)$ (ρυθμός εκμάθησης) συνήθως επιλέγεται σταθερή.

5. Γραμμικοί ταξινομητές

$$X1 = [(1,6),(7,2),(8,9),(9,9)]$$
$$X2 = [(2,1),(2,2),(2,4),(7,1)]$$

$$\hat{w}_{MS} = \begin{bmatrix} 0{,}075 & 0{,}2 & -1{,}19 \end{bmatrix}^T$$



Ho-Kasyap

$\eta(k) = 0{,}5$

9829 επαναλήψεις

$\varepsilon = 10^{-7}$

$$\hat{w}_{HK} = \begin{bmatrix} 1{,}29 & 2{,}15 & -13{,}18 \end{bmatrix}^T$$

$$\hat{y}_{HK} = \begin{bmatrix} 1{,}01 & 0{,}15 & 16{,}49 & 17{,}78 & 8{,}45 & 6{,}3 & 2 & 2 \end{bmatrix}^T$$

Πολλές κλάσεις (> 2)

Ένας απλός τρόπος επέκτασης των αλγορίθμων σε προβλήματα πολλών κλάσεων, είναι η κατάτμηση σε M προβλήματα 2 κλάσεων (κάθε κλάση έναντι των υπολοίπων).

Ο καλύτερος τρόπος να προσεγγίσουμε το πρόβλημα είναι να κατασκευάσουμε έναν πλήρη ταξινομητή, ακολουθώντας τα προηγούμενα βήματα, με μόνη διαφορά ότι το διάνυσμα y γίνεται πίνακας Y με στοιχεία,

$$Y(N\times M)=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix},\ y_i=\begin{bmatrix} 0 & \cdots & \overset{x_i\in M\rightarrow y_{i,m}}{1} & \cdots & 0 \end{bmatrix}$$

5. Γραμμικοί ταξινομητές

και το κριτήριο: $$J(w)=\frac{1}{2}\mathrm{tr}\left\{\left(XW-Y\right)^{\mathsf{T}}\left(XW-Y\right)\right\}$$

Ο δε πίνακας των βαρών δίδεται από την ίδια σχέση:

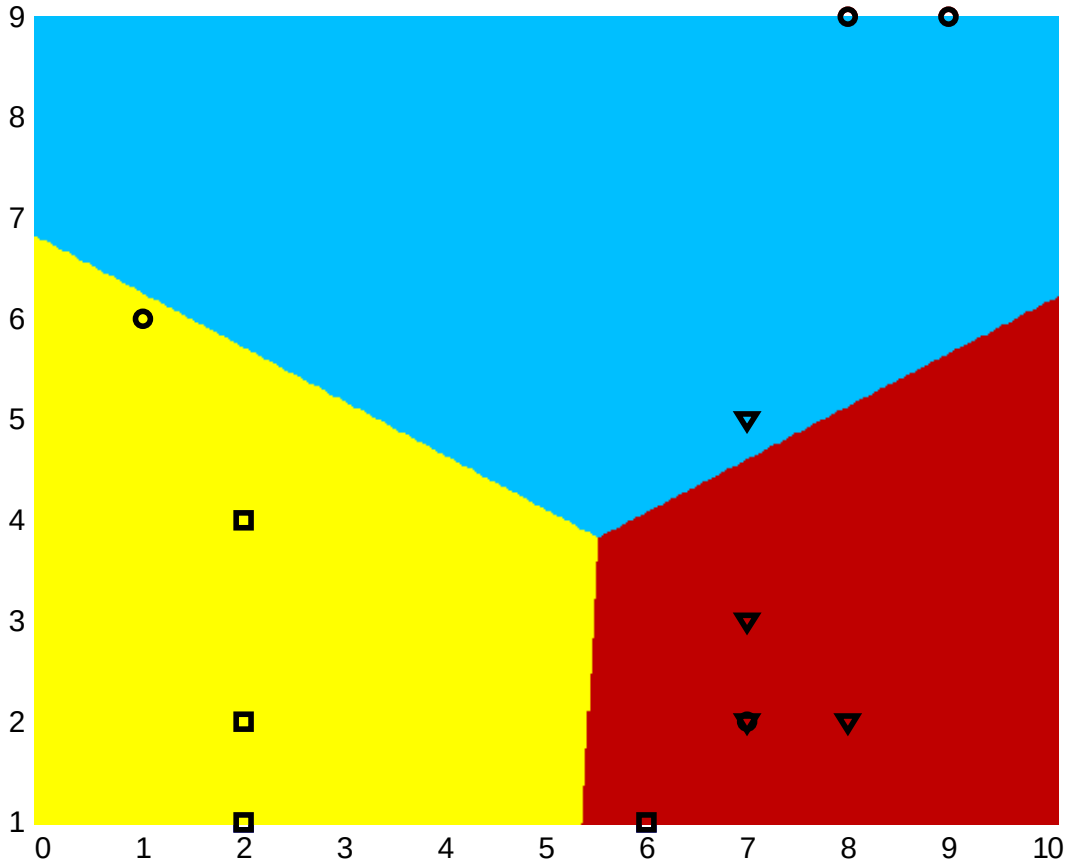
$$\hat{W}=(X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}Y$$
$$=X^{\dagger}Y$$

Τα (υπερ)επίπεδα διαχωρισμού περιγράφονται από την εξίσωση:

$$\left(w_k-w_j\right)^{\mathsf{T}}x+\left(w_{ko}-w_{jo}\right)+0$$

5. Γραμμικοί ταξινομητές

$$W = \begin{matrix} -0.0011 & -0.0943 & 0.0955 \\ 0.1193 & -0.0544 & -0.0649 \\ -0.1178 & 1.0608 & 0.0569 \end{matrix}$$



5. Γραμμικοί ταξινομητές

Το ελάττωμα της μη τέλειας ταξινόμησης, ακόμη και όταν τα δεδομένα είναι γραμμικά διαχωρίσιμα, ισχύει και στη περίπτωση αυτή.

Στη δημοσίευση,

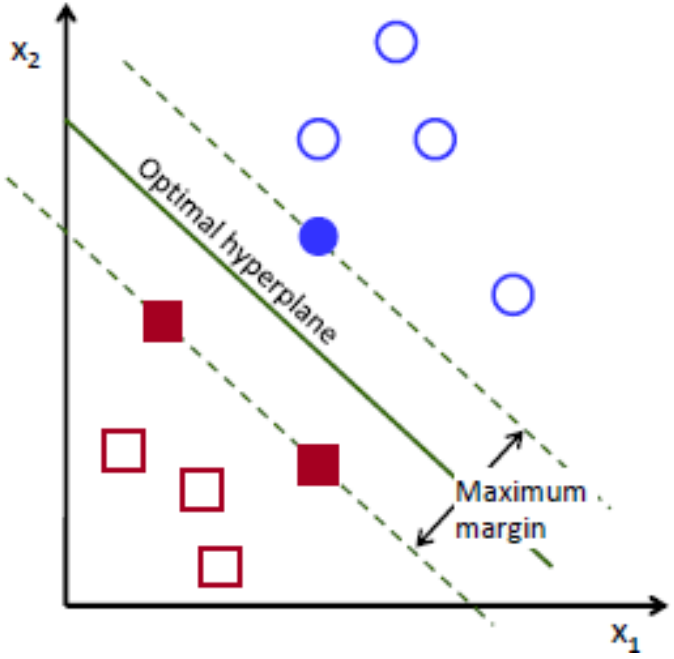
T. L. Teng and C. C. Li, "ON A GENERALIZATION OF THE HO-KASHYAP ALGORITHM TO MULTI-CLASS PATTERN CLASSIFICATION", Proceedings of the 3rd Annual Princeton Conference on Information Sciences and Systems, 1969.

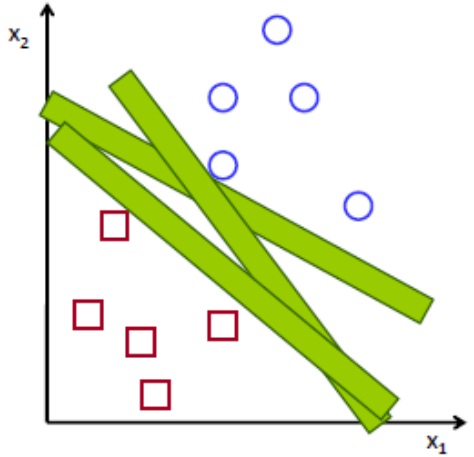
προτείνεται μία γενίκευση του αλγορίθμου Ho-Kasyap στις περιπτώσεις πολλών κλάσεων.

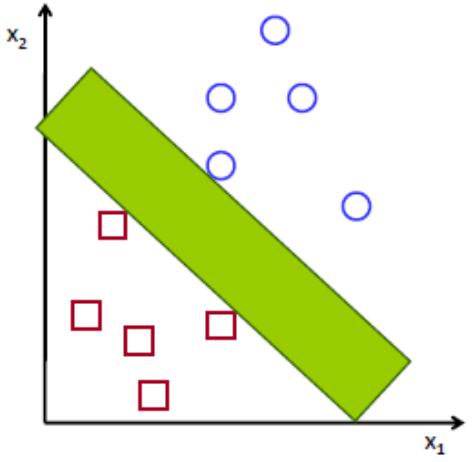
2. Μηχανές διανυσμάτων στήριξης (SVM)

1. Γραμμικά διαχωρίσιμες κλάσεις

Στις μηχανές διανυσματικής στήριξης η ποσότητα που βελτιστοποιείται είναι το «**περιθώριο**», δηλαδή η απόσταση από το εγγύτερο σημείο κάθε κλάσης.





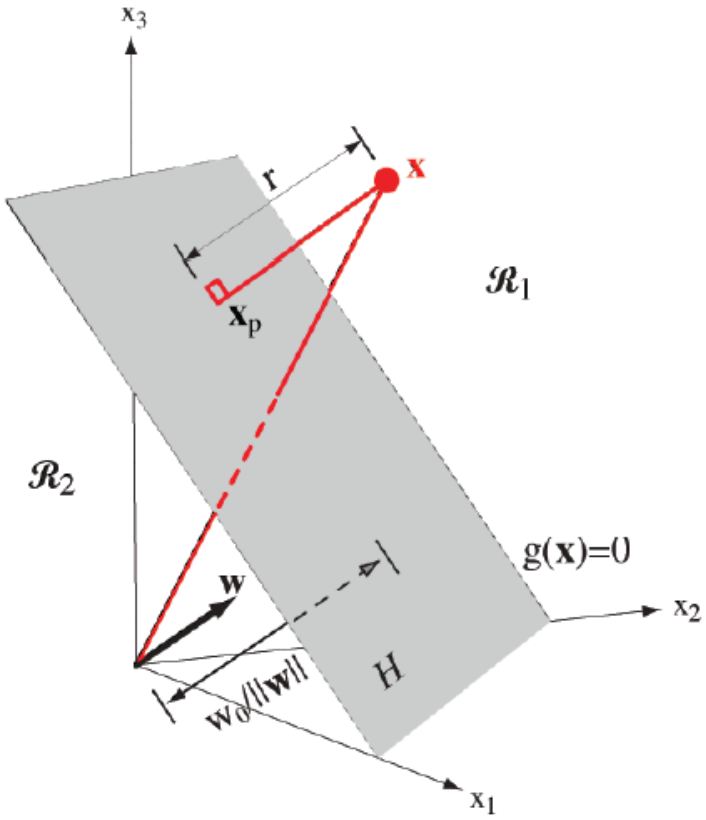


5. Γραμμικοί ταξινομητές

Στην εξίσωση της επιφάνειας διάκρισης,

$$g(x) = w^T x + w_o$$

w είναι η διεύθυνση και w_o η θέση.



5. Γραμμικοί ταξινομητές

- Τα δεδομένα του προβλήματος:

$$g(x)>0 \text{ όταν το } x \in \omega_1, \quad y=1$$

$$g(x)<0 \text{ όταν το } x \in \omega_2, \quad y=-1$$

$$g(x)=0 \text{ ορίζει την επιφάνεια διάκρισης}$$

- Επομένως για κάθε σωστά ταξινομημένο πρότυπο,

$$y_i g(x_i)>0$$

Η απόσταση του x από το επίπεδο είναι,

$$d=\frac{|g(x_i)|}{\|w\|}=\frac{y_i g(x_i)}{\|w\|}=\frac{y_i\left(w^T x_i+w_o\right)}{\|w\|}$$

5. Γραμμικοί ταξινομητές

- Το **περιθώριο** ορίζεται ως η κάθετη απόσταση από το εγγύτερο x_i,

$$m = \frac{\min_i \left\{ y_i \left(w^T x_i + w_0 \right) \right\}}{\|w\|}$$

- Επομένως αναζητούμε το,

$$\arg\max_{w, w_0} \left\{ \|w\|^{-1} \min_i \left\{ y_i \left(w^T x_i + w_0 \right) \right\} \right\}$$

- Πολλαπλασιάζοντας τα w, w_0 έτσι ώστε,

$$y_{i_{\min}} \left(w^T x_{i_{\min}} + w_0 \right) = 1$$

απλοποιεί το πρόβλημα,

5. Γραμμικοί ταξινομητές

$$\arg\max_{w,w_o} \frac{1}{\|w\|} = \arg\min_{w,w_o} \|w\|^2$$

ενώ όλα τα σημεία ικανοποιούν την,

$$y_i \left(w^T x_i + w_o \right) \geq 1$$

- Στη διαδικασία της βελτιστοποίησης θα υπάρχει τουλάχιστον ένα σημείο που να ικανοποιεί την ισότητα (δύο τουλάχιστον στο τέλος). Τα σημεία αυτά καλούνται **ενεργά**, αλλιώς **ανενεργά**.

5. Γραμμικοί ταξινομητές

- Καταλήγουμε δηλαδή στο ακόλουθο μη γραμμικό (τετραγωνικό) πρόβλημα μεγιστοποίησης υπό *N* περιορισμούς:

$$\min_{w,w_o} \|w\|^2$$

$$y_i \left(w^T x_i + w_o \right) \geq 1 \qquad i=1,\ldots, N$$

Το πρόβλημα αυτό έχει τοπικό ελάχιστο που είναι και ολικό, λόγω της κυρτότητας της συνάρτησης υπό μεγιστοποίηση και τους *N* γραμμικούς περιορισμούς.

5. Γραμμικοί ταξινομητές

Βελτιστοποίηση συναρτήσεων υπό περιορισμούς

$$\min \quad f(x)$$
$$h_l(x) = 0 \quad l = 1, \ \ldots, \ r$$
$$g_j(x) \le 0 \quad j = 1, \ \ldots, \ m$$

Langrangian

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{r} \lambda_i h_i(x) + \sum_{i=1}^{m} \mu_i g_i(x)$$

ελαχιστοποίηση

$$\nabla f(x^*) + \sum_{i=1}^{r} \lambda_i^* \cdot \nabla h(x^*) + \sum_{i=1}^{m} \mu_i^* \cdot \nabla g(q^*) = 0$$

Συνθήκες Karush-Kuhn-Tucker (ΚΚΤ)

$$g_j(x^*) \le 0 \quad j = 1, \ \ldots, \ m$$
$$h_l(x^*) = 0 \quad l = 1, \ \ldots, \ r$$

μη αρνητικά

$$\mu_j^* \ge 0 \quad j = 1, \ \ldots, \ m$$

συμπληρωματική χαλαρότητα

$$\mu_j^* \cdot g_j(x^*) = 0 \quad j = 1, \ \ldots, \ m$$

5. Γραμμικοί ταξινομητές

Η δυϊκή διατύπωση

$$\mathcal{D}(\lambda,\mu) = \max_{x} \mathcal{L}(x,\lambda,\mu)$$

$$\max_{\lambda,\mu} \mathcal{D}(\lambda,\mu), \quad \lambda,\mu > 0$$

5. Γραμμικοί ταξινομητές

- **Επομένως:** συνθήκες ακρότατων επί της Λαγκρανζιανής,

$$\mathcal{L}(w, w_o, \lambda) \triangleq \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left[y_i \left(w^T x_i + w_o \right) \right]$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial w_o} = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 = \lambda_i \left[y_i (w^T x_i + w_o) - 1 \right], \; i = 1, \; 2, \; \ldots, \; N$$

και: $$\lambda_i \geq 0, \; i = 1, \; 2, \; \ldots, \; N$$

5. Γραμμικοί ταξινομητές

- Δυϊκή διατύπωση:

$$\max_{\lambda} \mathcal{L}(w^*, w_o^*, \lambda)$$

υπό τους περιορισμούς: $\sum_{i=1}^{N} \lambda_i y_i = 0, \quad \lambda \geq 0$

Αντικαθιστώντας τα w, w_o :

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{ij}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j \right)$$

υπό τους περιορισμούς: $\sum_{i=1}^{N} \lambda_i y_i = 0, \quad \lambda \geq 0$

5. Γραμμικοί ταξινομητές

Το πρόβλημα αυτό είναι επίσης δύσκολο, αλλά υπόκειται σε περιορισμούς ισότητας, γεγονός που καθιστά την επίλυση του ευκολότερη από του αρχικού. Αφού υπολογίσουμε τα λ, τα w, w_o βρίσκονται από τις σχέσεις του αρχικού προβλήματος,

$$\frac{\partial}{\partial w}\mathcal{L}(w, w_o, \lambda) = 0 \Rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

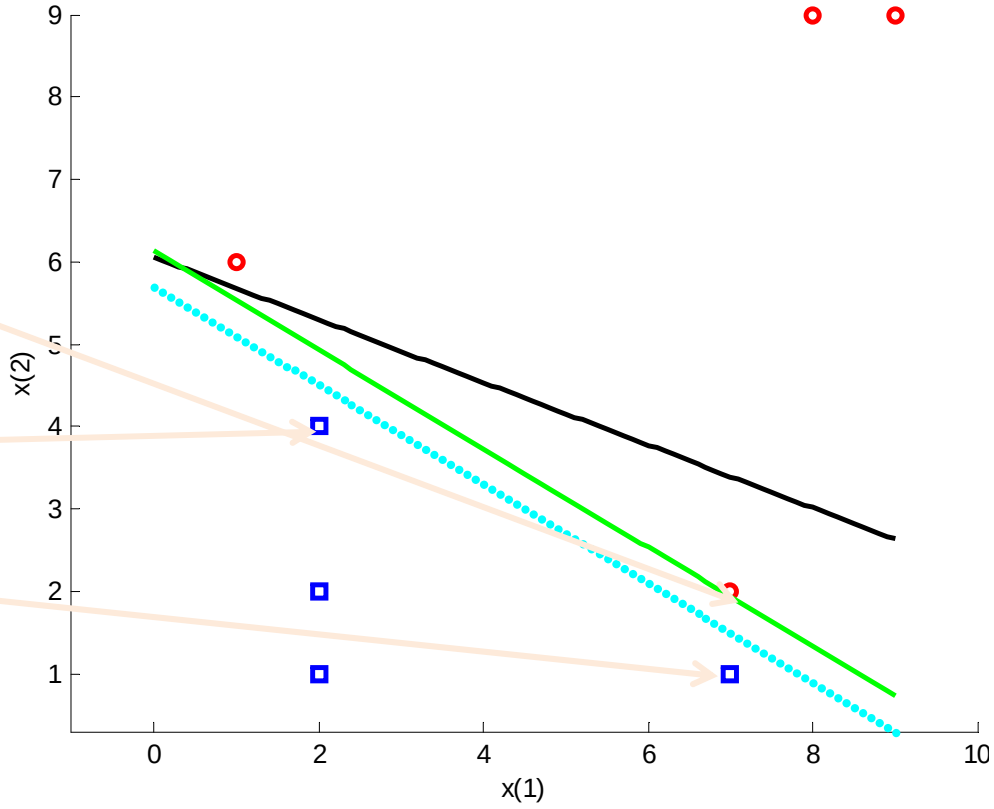
$$\lambda_i \left[y_i (w^T x_i + w_o) - 1 \right] = 0, \; i = 1, \; 2, \; \ldots, \; N \Rightarrow$$

$$w_o = \frac{1}{N_s} \sum \left(y_i - \sum \lambda_s y_s x_s \right)$$

Τα x που προκύπτουν από μη μηδενικά λ καλούνται διανύσματα στήριξης.

5. Γραμμικοί ταξινομητές

$$X=\begin{bmatrix}1 & 6\\ 7 & 2\\ 8 & 9\\ 9 & 9\\ 2 & 1\\ 2 & 2\\ 2 & 4\\ 7 & 1\end{bmatrix},\quad y_{SVM}=\begin{bmatrix}1\\ 1\\ 1\\ 1\\ -1\\ -1\\ 1\\ -1\end{bmatrix},\quad \lambda=\begin{bmatrix}0\\ 2{,}72\\ 0\\ 0\\ 0\\ 0\\ 0{,}24\\ 2{,}48\end{bmatrix}$$



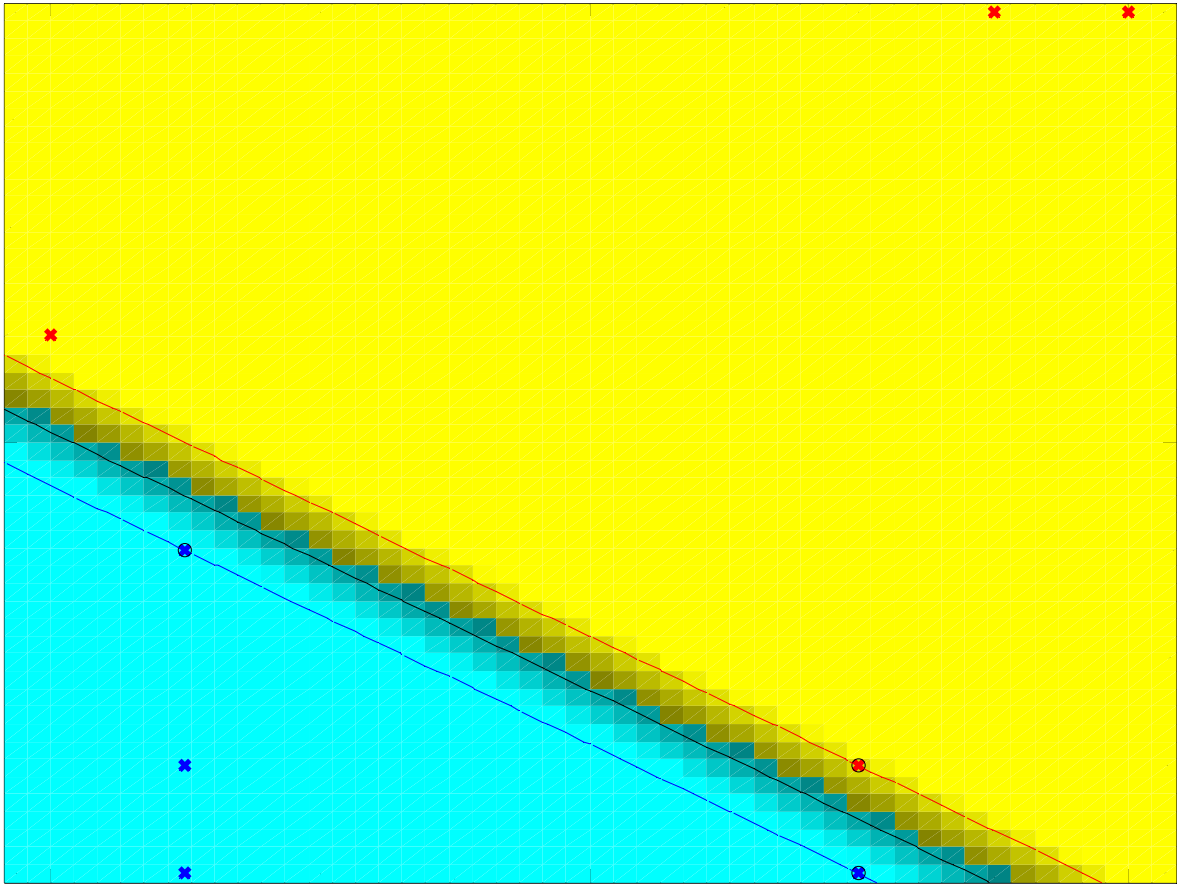
$$w_{SVM}=\begin{bmatrix}1{,}2 & 2\end{bmatrix},\quad w_0=\begin{bmatrix}-11{,}4\end{bmatrix}$$

$$\text{περιθώριο}=\frac{2}{\|w\|}=0{,}86$$

SVM-μέθοδος Platt

$\varepsilon = 10^{-10}$, 24328 επαναλήψεις

5. Γραμμικοί ταξινομητές



Πιο όμορφο σχήμα μέσω της

- **`svcplot_book(X,y,'linear',0,0,lamda,-w0)`**
  (Θεοδωρίδης)

Ταξινόμηση

- Ας μη ξεχνάμε ότι τελικός στόχος είναι η ταξινόμηση νέων δεδομένων. Έτσι για νέο σημείο z, υπολογίζουμε,

$$y_z = w^T z + w_0$$

  - αν $y_z > 0$ ταξινομούμε το z στη $1^η$ κλάση, αλλιώς στη $2^η$.

Σχόλια

- Η επίλυση του προβλήματος της κυρτής βελτιστοποίησης υπό περιορισμούς δεν είναι απλή υπόθεση. Έχουν προταθεί και υλοποιηθεί πολλοί επαναληπτικοί αλγόριθμοι, καθώς δεν υπάρχει αναλυτική λύση. Η διάσταση του προβλήματος είναι ανάλογη της διάστασης N των δεδομένων.

5. Γραμμικοί ταξινομητές

Ιστορικό

Οι μηχανές διανυσματικές στήριξης έχουν τις ρίζες τους στη ***στατιστική θεωρία μάθησης*** των ***Vapnik-Chervonenkis*** (). Η θεωρία αυτή εξετάζει τη πολυπλοκότητα των διαφόρων προτύπων σε σχέση με την ελαχιστοποίηση της συνάρτησης προσδοκώμενου ρίσκου,

$$R(f) = \int C(f(x), y) \mathrm{d}p(x, y)$$

όπου $C(f(x),\ y)$ είναι κάποια κατάλληλη συνάρτηση κόστους, πχ $= (f(x) - y)^2$. Όμως η σππ $p(x, y)$ δεν είναι συνήθως διαθέσιμη, οπότε ελαχιστοποιούμε το εμπειρικό ρίσκο,

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} C\left(f(x_i), y_i\right)$$

5. Γραμμικοί ταξινομητές

Το εμπειρικό ρίσκο πλησιάζει στο αναμενόμενο καθώς το N αυξάνει, αλλά για λίγα δείγματα έχουμε πρόβλημα: η ικανότητα γενίκευσης είναι πτωχή.

Πως μπορούμε να βελτιστοποιήσουμε αυτή την ιδιότητα; Διαισθητικά, επιλέγοντας το απλούστερο υπόδειγμα που εξηγεί τα δεδομένα. Εδώ υπεισέρχεται η ***διάσταση Vapnik-Chervonenkis (VC) :***

- Είναι μέτρο πολυπλοκότητας: μετρά το μέγιστο αριθμό παραδειγμάτων που μπορεί να εξηγηθεί από μία οικογένεια συναρτήσεων $f(\alpha)$.

5. Γραμμικοί ταξινομητές

Πως υπεισέρχεται; Αποδεικνύεται ότι η ακόλουθη ανισότητα ισχύει με πιθανότητα 1-η:

$$R(f) \le R_{emp}(f) + \underbrace{\sqrt{\frac{h\left(\ln\left(\frac{2N}{h}\right)+1\right)-\ln\left(\frac{\eta}{4}\right)}{N}}}_{\text{εμπιστοσύνη VC}}$$

όπου h η διάσταση VC της $f(\alpha)$, $N > h$.

(καθώς ο λόγος N/h μεγαλώνει, το διάστημα εμπιστοσύνης VC μικραίνει)

5. Γραμμικοί ταξινομητές

Δυστυχώς οι υπολογισμοί αυτοί είναι δύσκολοι έως αδύνατοι σε γενικά μη γραμμικά προβλήματα. Σε γραμμικά διαχωρίσιμες κλάσεις όμως, και χρησιμοποιώντας λογική SVM, αποδεικνύεται ότι (Vapnik, 1998):

$$h \le \min\left(\frac{R^2}{m^2}, q\right) + 1$$

όπου R η ακτίνα της μικρότερης σφαίρας που περιέχει όλα τα δεδομένα. Άρα,

- μεγιστοποιώντας το περιθώριο m, ελαχιστοποιούμε τη διάσταση VC, h.
- ελαχιστοποιώντας το h, ελαχιστοποιούμε το άνω φράγμα του προσδοκώμενου ρίσκου (γιατί το εμπειρικό ρίσκο είναι μηδέν).

2. Μηχανές διανυσμάτων στήριξης (SVM)

2. Μη γραμμικά διαχωρίσιμες κλάσεις

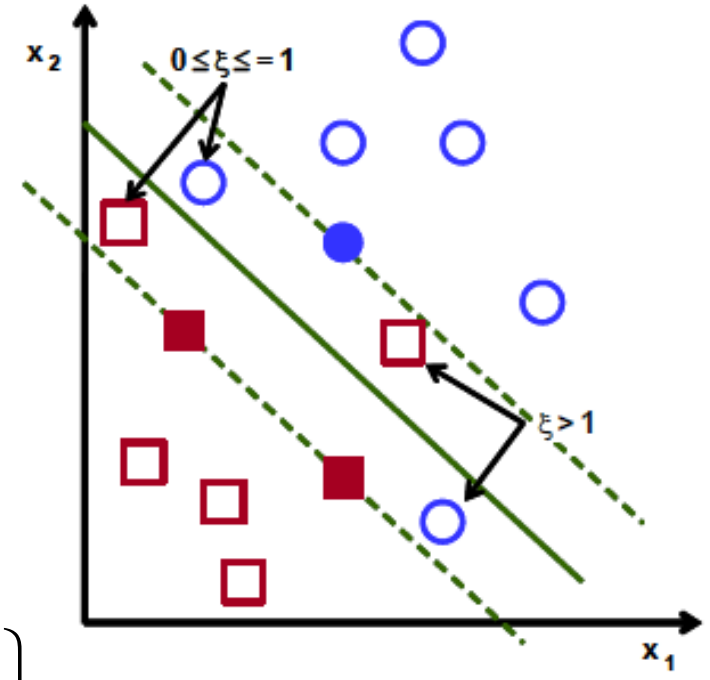
Σε προβλήματα που οι κλάσεις δεν είναι γραμμικά διαχωρίσιμες, η θεωρία των SVM μπορεί να επεκταθεί ώστε να δώσει λύση. Στη περίπτωση αυτή τα σημεία x_i είναι τριών ειδών:

1. Σημεία εκτός περιθωρίου σωστά ταξινομημένα: $y_i(w^T x_i + w_0) > 1$
2. Σημεία εντός περιθωρίου σωστά ταξινομημένα: $0 \le y_i(w^T x_i + w_0) < 1$
3. Σημεία λάθος ταξινομημένα: $y_i(w^T x_i + w_0) < 0$

5. Γραμμικοί ταξινομητές

Και οι τρεις κατηγορίες περιγράφονται ομοίως από την :

$$y_i(w^T x_i + w_o) \geq 1 - \xi_i$$



όπου τα $\xi_i > 0$ είναι στην ουσία ***μεταβλητές χαλαρότητας*** (slack).
Στόχος τώρα είναι η ελαχιστοποίηση της συνάρτησης,

$$\Theta(\xi) = \sum_{i=1}^{N} I(\xi_i - 1) \quad I(\xi_i) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi_i \leq 0 \end{cases}$$

που μετρά τις εσφαλμένες ταξινομήσεις.

5. Γραμμικοί ταξινομητές

Με τη συνάρτηση αυτή (που δεν είναι διαφορίσιμη) το πρόβλημα αυτό δεν λύνεται για πολλά σημεία. Αντ' αυτού θα λύσουμε το προσεγγιστικό,

$$J(w, w_o, \xi) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i$$

υπό τους περιορισμούς,

$$y_i(w^T x_i + w_o) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

όπου η σταθερά C παίζει το ρόλο αντιστάθμισης ανάμεσα στη πολυπλοκότητα (μικρά C) και την απόδοση (μεγάλα C)

5. Γραμμικοί ταξινομητές

Η ανάλογη Λαγκρανζιανή είναι,

$$\mathcal{L}(w,w_o,\xi)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\lambda_i\left(y_i x_i+w_o-1+\xi_i\right)-\sum_{i=1}^{N}\mu_i\xi_i$$

ενώ οι συνθήκες ΚΚΤ,

$$y_i(w^T x_i+w_o)-1+\xi_i\geq 0$$
$$\lambda_i[y_i(w^T x_i+w_o)-1+\xi_i]=0$$
$$\mu_i\xi_i=0$$
$$\mu_i,\lambda_i,\xi_i\geq 0$$

5. Γραμμικοί ταξινομητές

Δυϊκό:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \lambda_i = C - \mu_i$$

και αντικαθιστώντας:

$$D(l) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j$$

Η μόνη διαφορά:

$$0 \le \lambda_i \le C, \quad i = 1, 2, \ldots, N$$

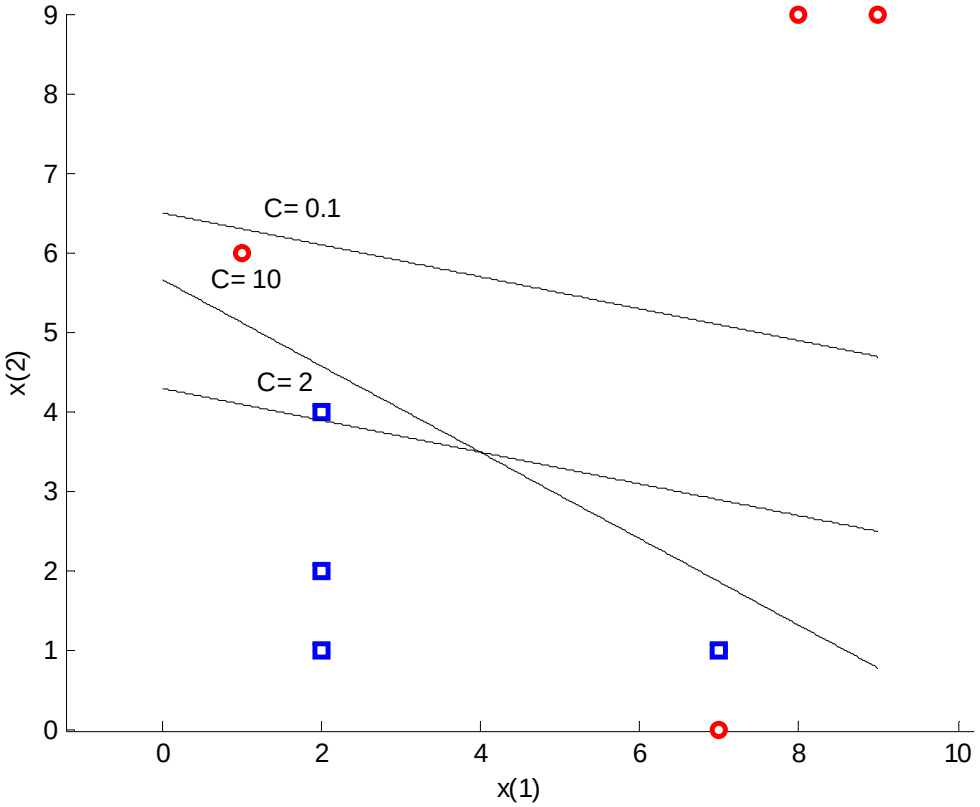
$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

5. Γραμμικοί ταξινομητές

Παράδειγμα

$$X = \begin{bmatrix} 1 & 6 \\ 7 & 0 \\ 8 & 9 \\ 9 & 9 \\ 2 & 1 \\ 2 & 2 \\ 2 & 4 \\ 7 & 1 \end{bmatrix}, \quad \lambda =$$

C=0.1	C=2	C=10
0.10	**1.72**	**8.23**
0.10	**2.00**	**10.00**
0.02	**0**	**0**
0	**0**	**0**
0	**0**	**0**
0.02	**0.09**	**0**
0.10	**2.00**	**10.00**
0.09	**1.63**	**8.23**



$\lambda_i < C$

Σημεία για τα οποία $\lambda_i > 0$ είναι διανύσματα στήριξης

C ↑ λιγότερα σφάλματα ταξινόμησης

Επέκταση σε πολλές κλάσεις

Ισχύει κι εδώ ότι και στη περίπτωση της μεθόδου των ελαχίστων τετραγώνων.

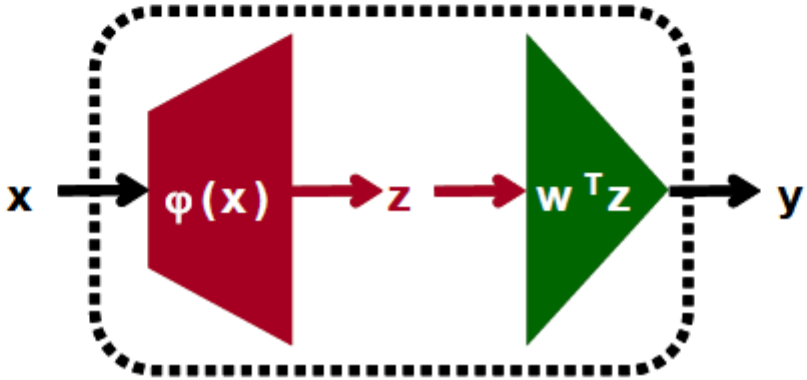
Μία γενίκευση της μεθόδου για πολλές κλάσεις παρουσιάζεται στη δημοσίευση:

❖Koby Crammer, Yoram Singer,"On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines", *Journal of Machine Learning Research*, **2**: 265-292 (2001).

Non-linear SVMs

Cover's theorem on the separability of patterns

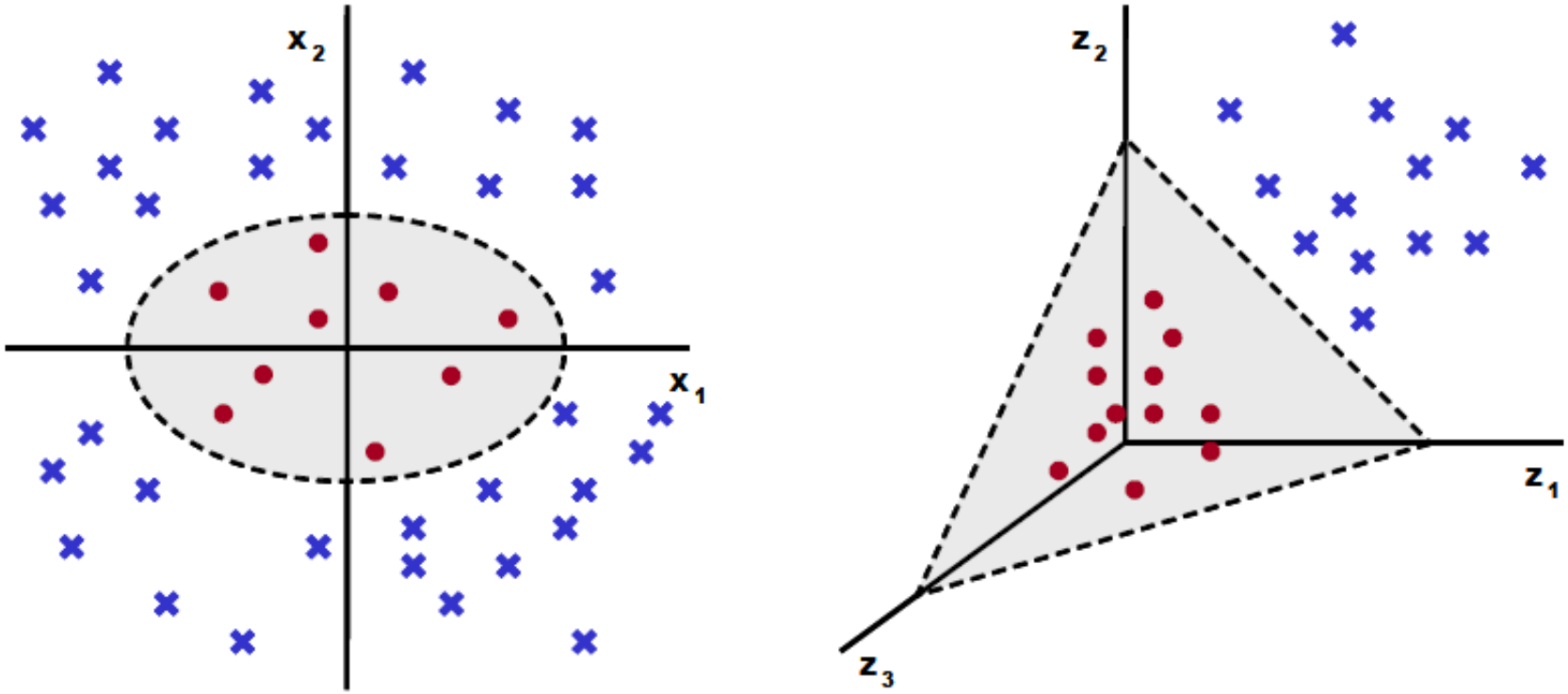
- *"A complex pattern-classification problem cast in a high-dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space"*
- The power of SVMs resides in the fact that they represent a robust and efficient implementation of Cover's theorem
- SVMs operate in two stages
  - Perform a non-linear mapping of the feature vector x onto a high-dimensional space that is hidden from the inputs or the outputs
  - Construct an optimal separating hyperplane in the high-dim space



5. Γραμμικοί ταξινομητές

$$\varphi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$



[Schölkopf, 2002 @; http://kernel-machines.org/]

CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU 7

5. Γραμμικοί ταξινομητές

Naïve application of this concept by simply projecting to a high-dimensional non-linear manifold has two major problems

- **Statistical**: operation on high-dimensional spaces is ill-conditioned due to the "curse of dimensionality" and the subsequent risk of overfitting
- **Computational**: working in high-dim requires higher computational power, which poses limits on the size of the problems that can be tackled

SVMs bypass these two problems in a robust and efficient manner

- First, generalization capabilities in the high-dimensional manifold are ensured by enforcing a **largest margin** classifier
  - Recall that generalization in SVMs is strictly a function of the margin (or the VC dimension), regardless of the dimensionality of the feature space
- Second, projection onto a high-dimensional manifold is only **implicit**
  - Recall that the SVM solution depends only on the dot product $\langle x_i, x_j \rangle$ between training examples
  - Therefore, operations in high-dim space $\varphi(x)$ do not have to be performed explicitly if we find a function $K(x_i, x_j)$ such that $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$
  - $K(x_i, x_j)$ is called a **kernel** function in SVM terminology

E 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU 8

Consider a pattern recognition problem in R^2

- Assume we choose a kernel function $K(x_i, x_j) = (x_i^T x_j)^2$
- Our goal is to find a non-linear projection $\varphi(x)$ such that $(x_i^T x_j)^2 = \varphi^T(x_i)\varphi(x_j)$
- Performing the expansion of $K(x_i, x_j)$

$$K(x_i, x_j) = (x_i^T x_j)^2 = \left((x_{1,1}, x_{1,2})^T (x_{2,1}, x_{2,2})\right)^2 = (x_{1,1}x_{2,1} + x_{1,2}x_{2,2})^2$$
$$= x_{1,1}^2 x_{2,1}^2 + 2x_{1,1}x_{2,1}x_{1,2}x_{2,2} + x_{1,2}^2 x_{2,2}^2$$
$$= (x_{1,1}^2, \sqrt{2}x_{1,1}x_{1,2}, x_{1,2}^2)^T (x_{2,1}^2, \sqrt{2}x_{2,1}x_{2,2}, x_{2,2}^2)$$

  - where $x_{i,k}$ denotes the k^{th} coordinate of example x_i

- So in using the kernel $K(x_i, x_j) = (x_i^T x_j)^2$, we are implicitly operating on a higher-dimensional non-linear manifold defined by

$$\varphi(x_i) = [x_{i,1}^2, \sqrt{2}x_{i,1}x_{i,2}, x_{i,2}^2]^T$$

- Notice that the inner product $\varphi^T(x_i)\varphi(x_j)$ can be computed in R^2 by means of the kernel $(x_i^T x_j)^2$ without ever having to project onto R^3!

666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

[Burges, 1998]

Kernel methods

Let's now see how to put together all these concepts

- Assume that our original feature vector x lives in a space R^D
- We are interested in non-linearly projecting x onto a higher dimensional implicit space $\varphi(x) \in R^{D1}$ $(D1 > D)$ where classes have a better chance of being linearly separable
  - Notice that we are not guaranteeing linear separability, we are only saying that we have a better chance because of Cover's theorem
- The separating hyperplane in R^{D1} will be defined by

$$\sum_{j=1}^{D1} w_j \varphi_j(x) + b = 0$$

- To eliminate the bias term b, let's augment the feature vector in the implicit space with a constant dimension $\varphi_0(x) = 1$
- Using vector notation, the resulting hyperplane becomes

$$w^T \varphi(x) = 0$$

- From our previous results, the optimal (maximum margin) hyperplane in the implicit space is given by

$$w = \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)$$

5. Γραμμικοί ταξινομητές

– Merging this optimal weight vector with the hyperplane equation

$$w^T \varphi(x) = 0$$

$$\Rightarrow \left(\sum_{i=1}^{N} \alpha_i y_i \varphi(x_i) \right)^T \varphi(x) = 0$$

$$\Rightarrow \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)^T \varphi(x) = 0$$

– and, since $\varphi^T(x_i)\varphi(x_j) = K(x_i, x_j)$, the optimal hyperplane becomes

$$\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) = 0$$

– Therefore, classification of an unknown example x is performed by computing the weighted sum of the kernel function with respect to the support vectors x_i (remember that only the support vectors have non-zero dual variables α_i)

How do we compute dual variables α_i in the implicit space?

- Very simple: we use the same optimization problem as before, and replace the dot product $\varphi^T(x_i)\varphi(x_j)$ with the kernel $K(x_i, x_j)$
- The Lagrangian dual problem for the non-linear SVM is simply

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i^T, x_j)$$

- subject to the constraints

$$\begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad i = 1 \ldots N \end{cases}$$

5. Γραμμικοί ταξινομητές

How do we select the implicit mapping $\varphi(x)$?

- As we saw in the example a few slides back, we will normally select a kernel function first, and then determine the implicit mapping $\varphi(x)$ that it corresponds to

Then, how do we select the kernel function $K(x_i, x_j)$?

- We must select a kernel for which an implicit mapping exists, this is, a kernel that can be expressed as the dot-product of two vectors

For which kernels $K(x_i, x_j)$ does there exist an implicit mapping $\varphi(x)$?

- The answer is given by Mercer's Condition

Mercer's Condition

Let $K(x, x')$ be a continuous symmetric kernel that is defined in the closed interval $a \leq x \leq b$

- The kernel can be expanded in the series:

$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x) \varphi_i(x')$$

  - Strictly speaking, the space where $\varphi(x)$ resides is a Hilbert space, a "generalization" of an Euclidean space where the inner product can be any inner product, not just the scalar dot product [Burges, 1998]
  - With positive coefficients $\lambda_i > 0 \ \forall i$
- For this expansion to be valid and for it to converge absolutely and uniformly, it is necessary and sufficient that the condition

$$\int_a^b \int_a^b K(x, x') \psi(x) \psi(x') dx dx' \geq 0$$

- holds for all $\psi(\cdot)$ for which $\int_a^b \psi^2(x) dx \leq \infty$
  - The functions $\varphi_i(x)$ are called eigenfunctions of the expansion, and the numbers λ_i are the eigenvalues. The fact that all of the eigenvalues are positive means that the kernel is positive definite
- Notice that the dimensionality of the implicit space can be infinitely large
- Mercer's Condition only tells us whether a kernel is actually an inner-product kernel, but it does not tell us how to construct the functions $\varphi_i(x)$ for the expansion

[Kaykin, 1999]

: 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU 14

Which kernels meet Mercer's condition?

- Polynomial kernels

$$K(x, x') = (x^T x' + 1)^p$$

  - The degree of the polynomial is a user-specified parameter

- Radial basis function kernels

$$K(x, x') = \exp\left(-\frac{1}{2\sigma^2}\|x - x'\|^2\right)$$

  - The width σ is a user-specified parameter, but the number of radial basis functions and their centers are determined automatically by the number of support vectors and their values

- Two-layer perceptron

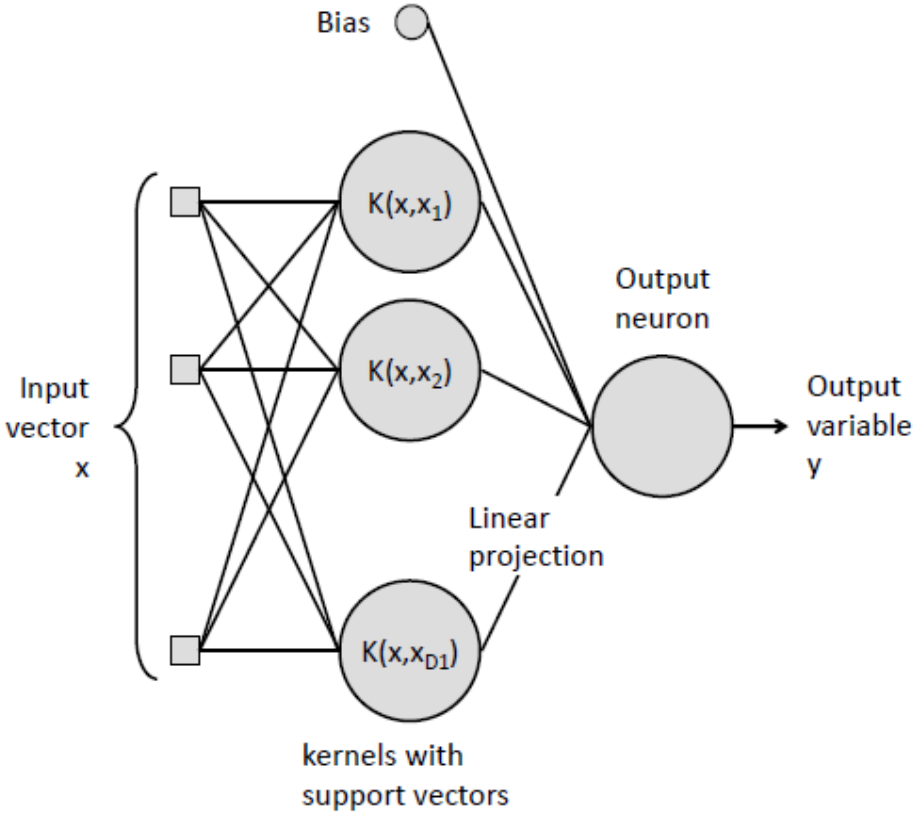
$$K(x, x') = \tanh(\beta_0 x^T x' + \beta_1)$$

  - The number of hidden neurons and their weight vectors are determined automatically by the number of support vectors and their values, respectively. The H-O weights are the Lagrange multipliers α_i
  - However, this kernel will only meet Mercer's condition for certain values of β_0 and β_1

[Burges, 1998; Kaykin, 1999]

: 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU 15

Architecture of an SVM



[Kaykin, 1999]

CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

Numerical example

To illustrate the operation of a non-linear SVM we will solve the classical XOR problem

- Dataset
  - Class 1: $x_1 = (-1,-1), \quad x_4 = (+1,+1)$
  - Class 2: $x_2 = (-1,+1), \quad x_3 = (+1,-1)$
- Kernel function
  - Polynomial of order 2: $K(x,x') = (x^T x' + 1)^2$

Solution

- The implicit mapping can be shown to be 5-dimensional

$$\varphi(x) = \begin{bmatrix} 1 & \sqrt{2}x_{i,1} & \sqrt{2}x_{i,2} & \sqrt{2}x_{i,1}x_{i,2} & x_{i,1}^2 & x_{i,2}^2 \end{bmatrix}^T$$

- To achieve linear separability, we will use $C = \infty$
- The objective function for the dual problem becomes

$$L_D(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4}\alpha_i\alpha_j y_i y_j k_{ij}$$

  - subject to the constraints $\begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \quad i = 1 \ldots N \end{cases}$

[Cherkassky and Mulier, 1998; Haykin, 1999]

666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

5. Γραμμικοί ταξινομητές

- where the inner product is represented as a 4×4 K matrix

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

- Optimizing with respect to the Lagrange multipliers leads to the following system of equations

$$\begin{aligned} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 &= 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 &= 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 &= 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 &= 1 \end{aligned}$$

- whose solution is $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.125$

- Thus, all data points are support vectors in this case

5. Γραμμικοί ταξινομητές

- For this simple problem, it is worthwhile to write the decision surface in terms of the polynomial expansion

$$w = \sum_{i=1}^{4} \alpha_i y_i \varphi(x_i) = \begin{bmatrix} 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 \end{bmatrix}^T$$

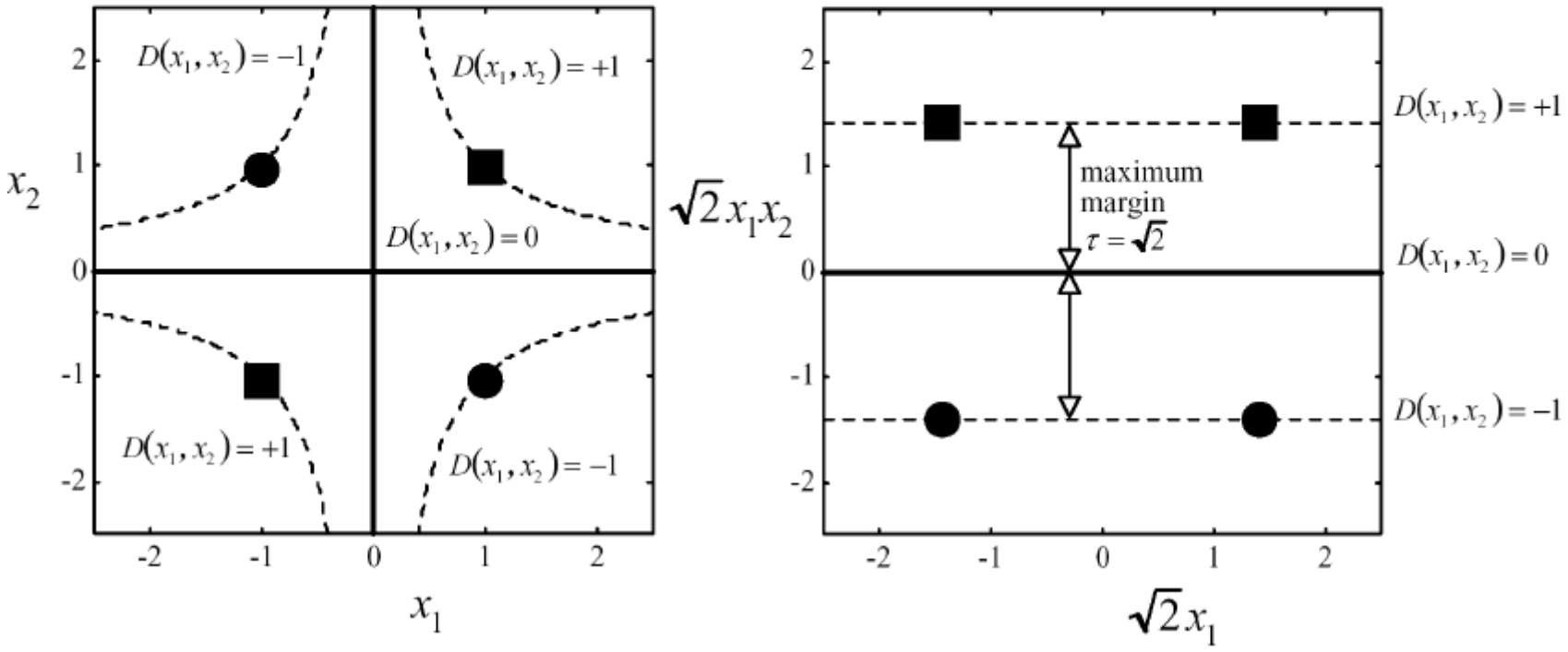
- resulting in the intuitive non-linear discriminant function

$$g(x) = \sum_{i=1}^{6} w_i \varphi_i(x) = x_1 x_2$$

- which has zero empirical error on the XOR training set

Decision function defined by the SVM

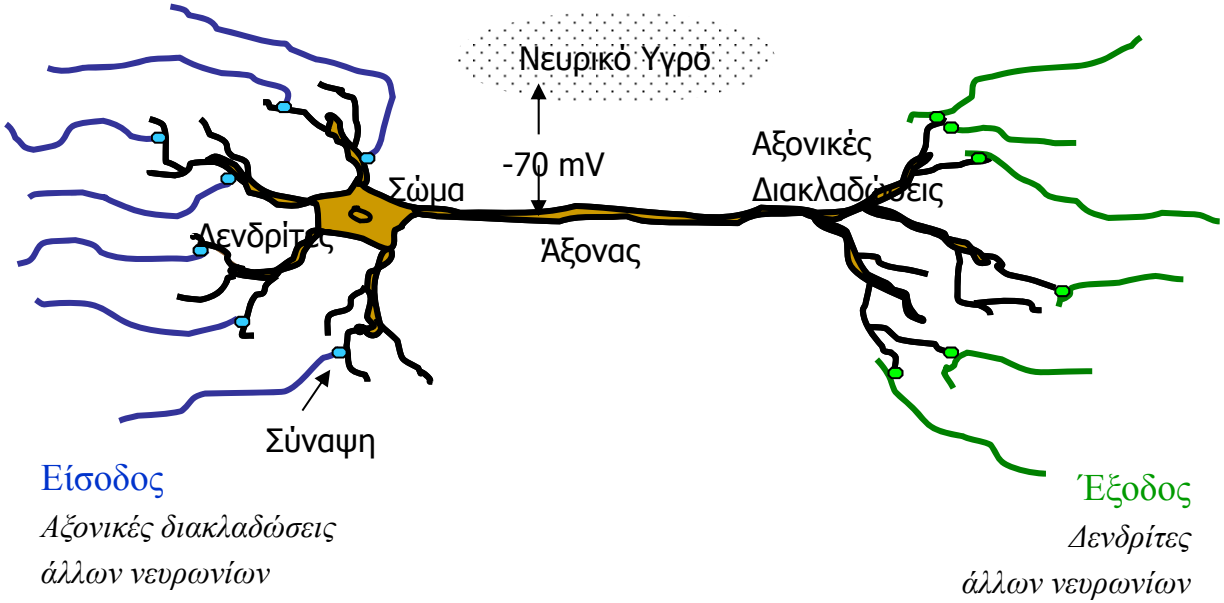
– Notice that the decision boundaries are non-linear in the original space R^2, but linear in the implicit space R^6



5. Γραμμικοί ταξινομητές

7. Τεχνητά νευρωνικά δίκτυα

(Φυσικός) νευρώνας

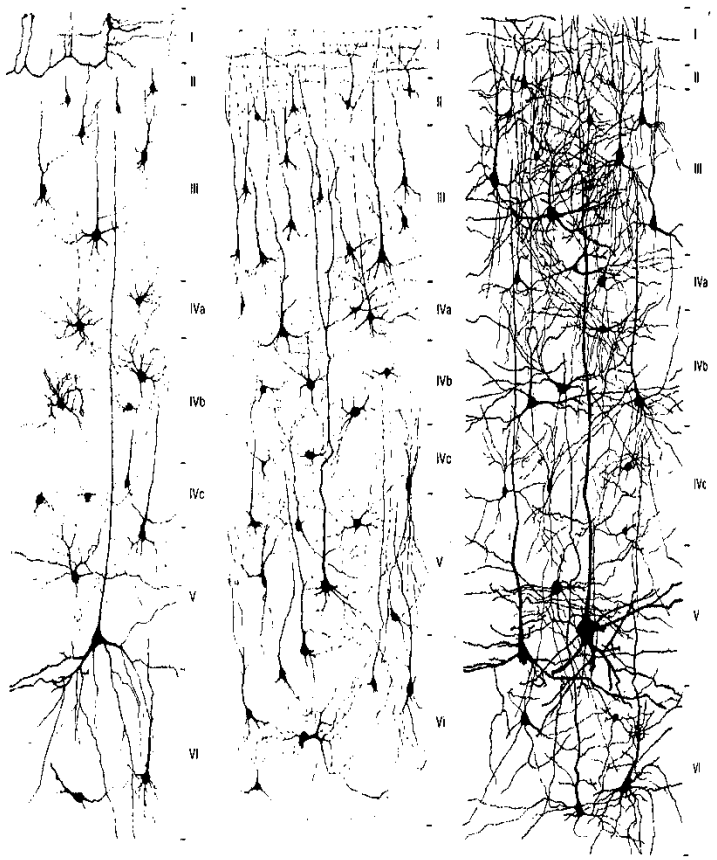


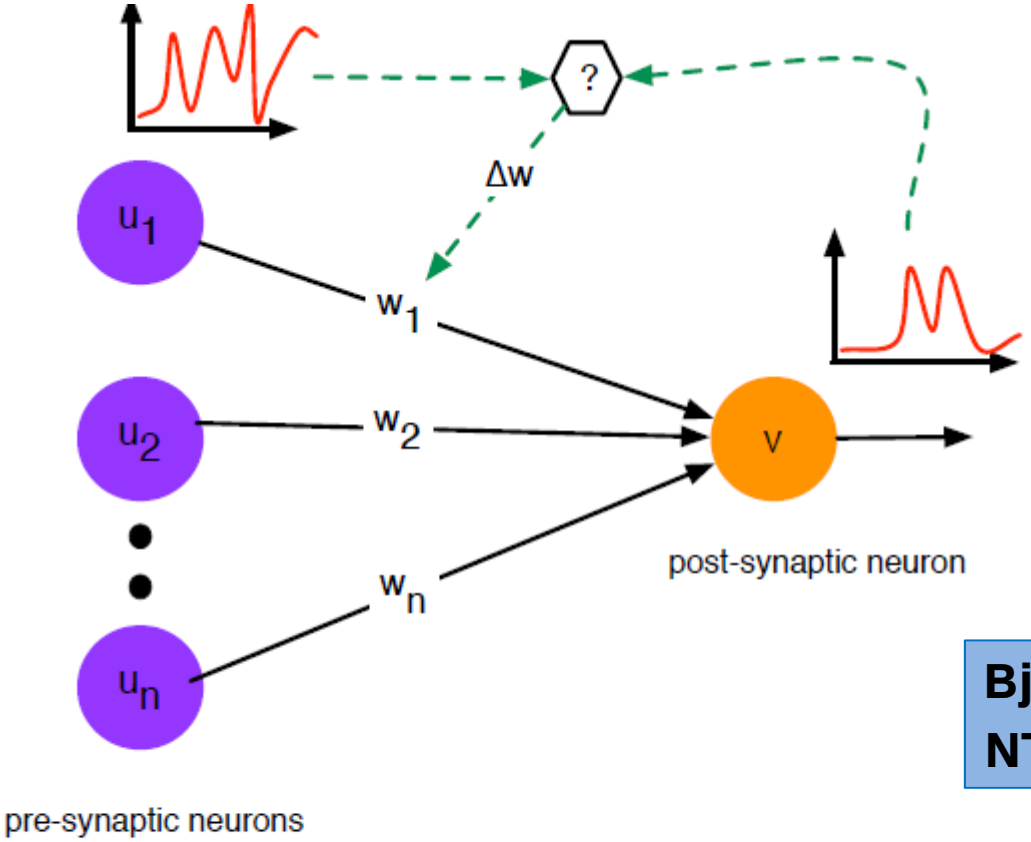
7. Τεχνητά νευρωνικά δίκτυα

Υπάρχουν περίπου $3 \cdot 10^{10}$ νευρώνες στον ανθρώπινο εγκέφαλο.

Κάθε νευρώνας διαθέτει κατά μέσο όρο 10000 συνάψεις (εισόδους) και 500 συναπτικές απολήξεις (εξόδους).

Το ανθρώπινο Κεντρικό Νευρικό Συστημα αποτελείται από περίπου 10^{15} συναπτικές συνδέσεις.





Björn Gambäck, Prof.,
NTNU, Trondheim

Figure 7: The essence of learning in neural networks: the comparison of pre-synaptic and post-synaptic firing histories determines changes to the connection weight between the two neurons. Here, weights have a single subscript, denoting the pre-synaptic neuron. Graphs are of the neuron's membrane potential as a function of time.

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells, such that A's efficiency as one of the cells firing B, is increased.

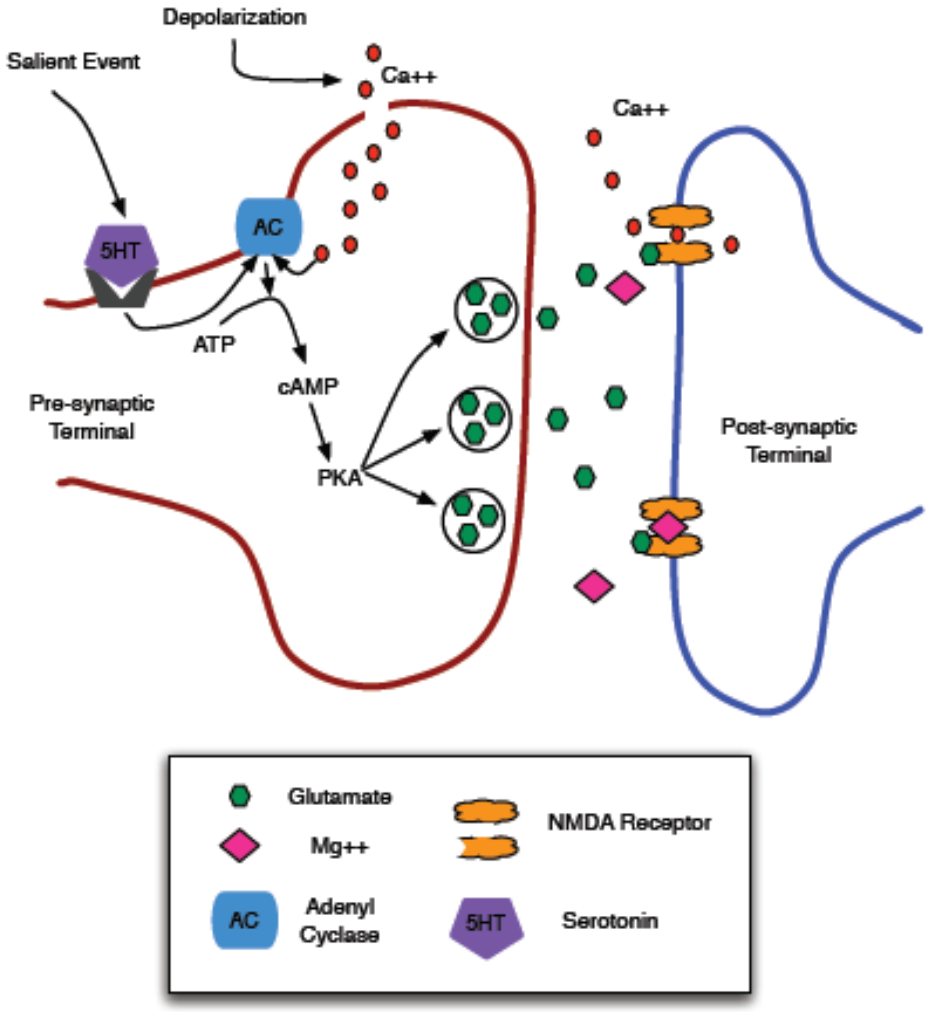
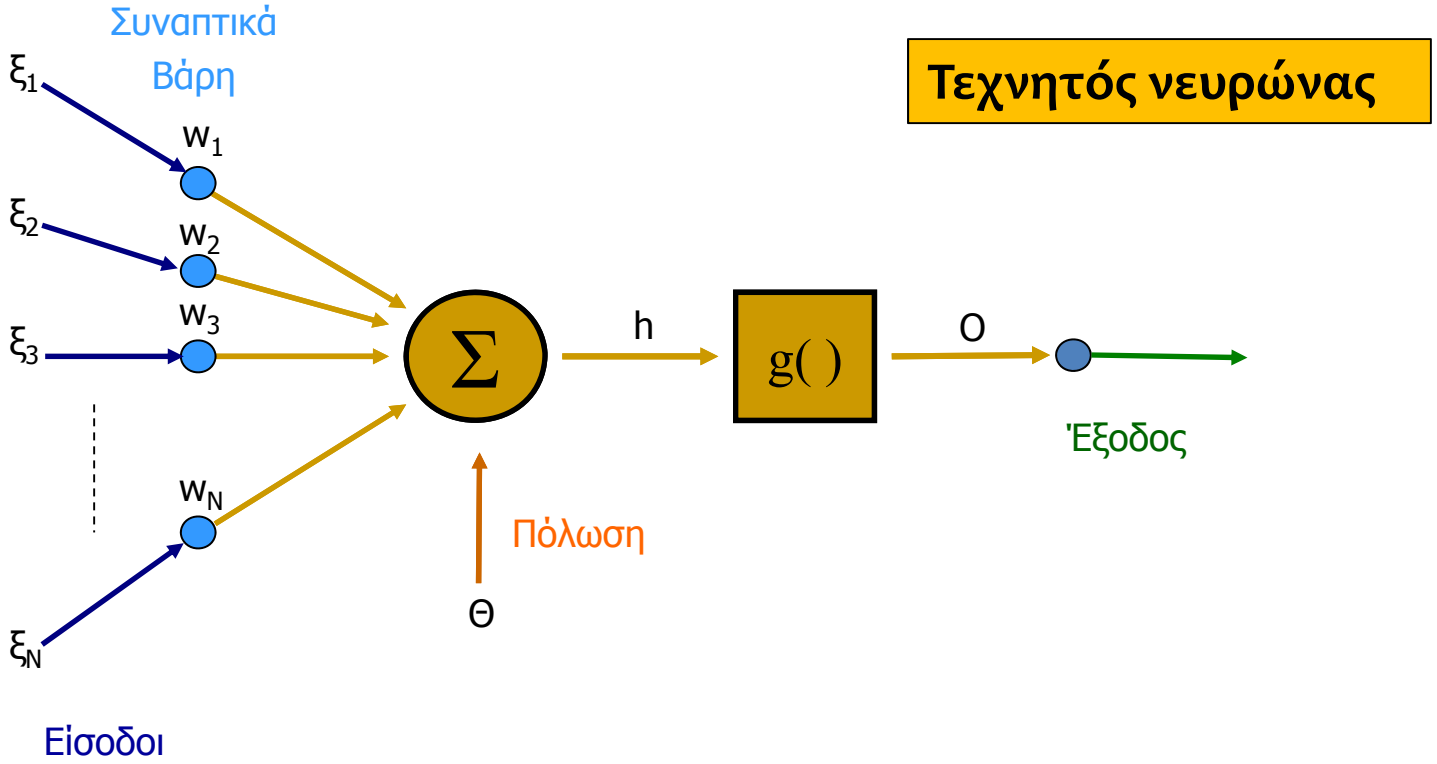


Figure 8: Overview of the key processes involved in coincidence detection by Adenyl Cyclase (AC) and the ensuing increase in propensity for neurotransmitter release in the pre-synaptic terminal. Note that the biochemical coincidences (serotonin and Ca++) are results of two other correlated activities: the detection of a salient event and the depolarization of the pre-synaptic terminal.

7. Τεχνητά νευρωνικά δίκτυα



Ενεργοποίηση $h = \sum_{k=1}^{N} w_k \xi_k + \theta$

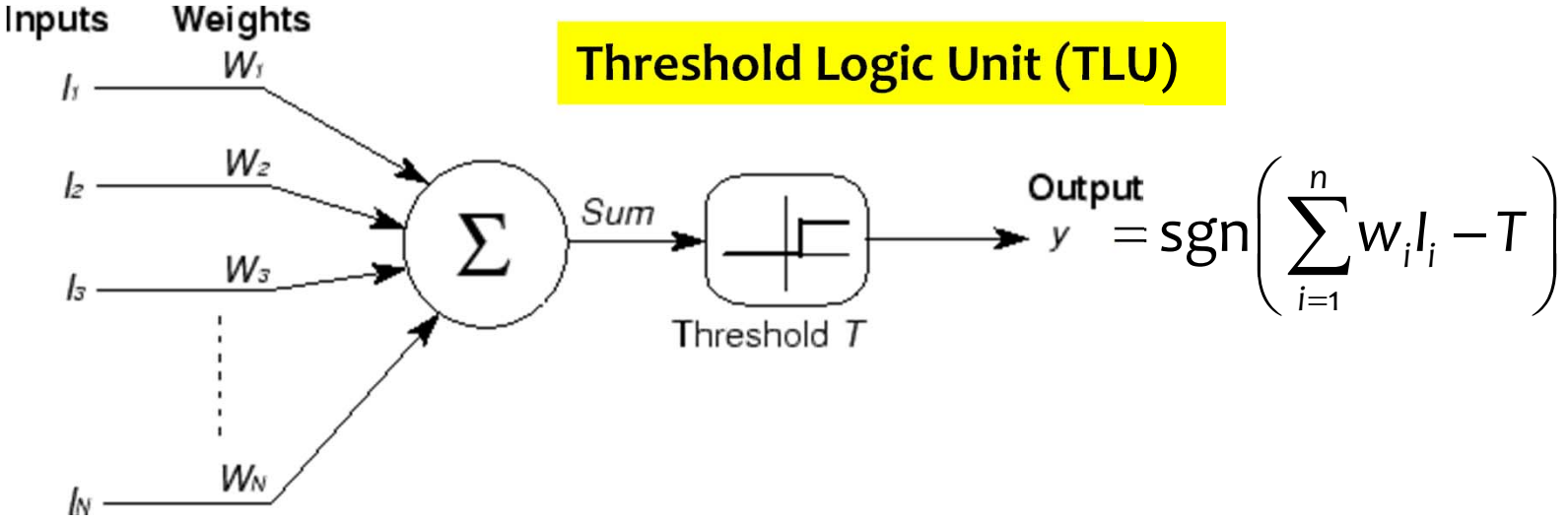
Συνάρτηση ενεργοποίησης: $g(\ )$

Έξοδος: $O = g(h) = g\left(\sum_{k=1}^{N} w_k \xi_k + \theta \right)$

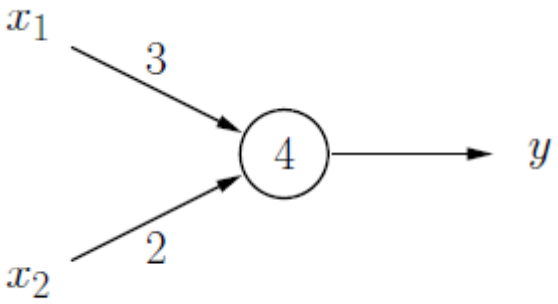
McCulloch and Pitts, 1943

- The modern era of ANNs starts in the 1940's, when Warren McCulloch (a psychiatrist and neuroanatomist) and Walter Pitts (a mathematician) explored the computational capabilities of networks made of very simple neurons
- A McCulloch-Pitts network fires if the sum of the excitatory inputs exceeds the threshold, as long as it does not receive an inhibitory input
- Using a network of such neurons, they showed that it was possible to construct any logical function

Χωρίς εκμάθηση



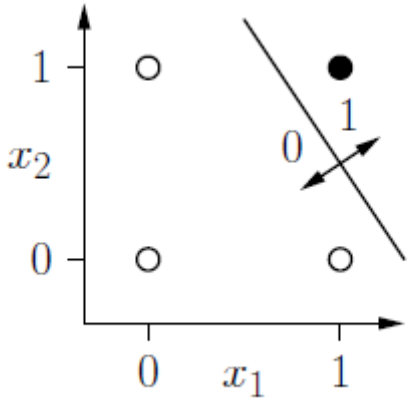
Threshold Logic Unit (TLU)



x_1	x_2	$3x_1 + 2x_2$	y
0	0	0	0
1	0	3	0
0	1	2	0
1	1	5	1

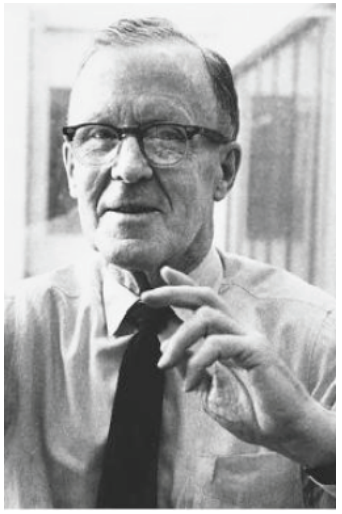
Τελεστής AND

Διαχωριστής κλάσης

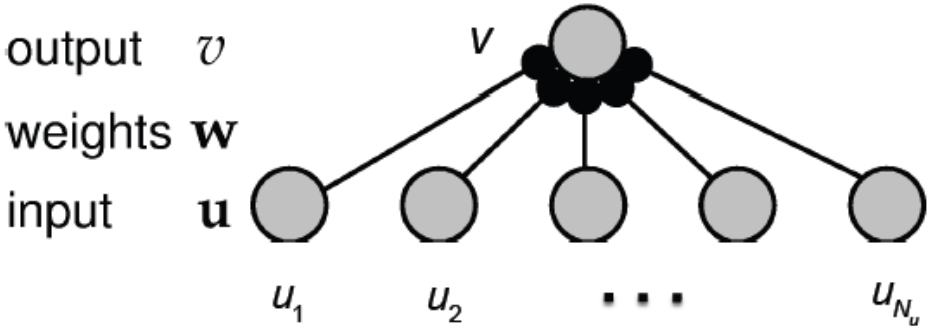


Hebb, 1949

- In his book "The organization of Behavior", Donald Hebb introduced his postulate of learning (a.k.a. Hebbian learning), which states that the effectiveness of a variable synapse between two neurons is increased by the repeated activation of one neuron by the other across that synapse
- The Hebbian rule has a strong similarity to the biological process in which a neural pathway is strengthened each time it is used

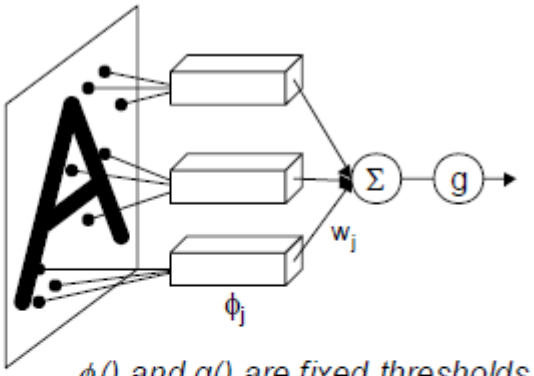


$$v = \mathbf{w} \bullet \mathbf{u} = \mathbf{w}^t \mathbf{u} \qquad \tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}$$



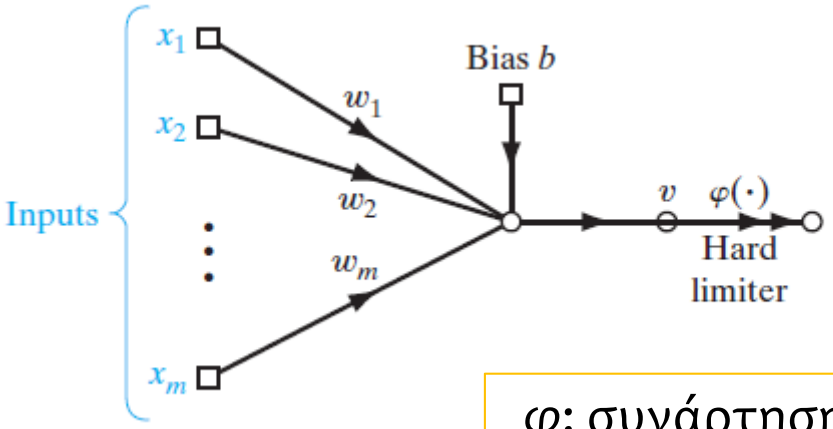
Rosenblatt, 1958

- Frank Rosenblatt introduced the perceptron, the simplest form of neural network
- The perceptron is a single neuron with adjustable synaptic weights and a threshold activation function
- Rosenblatt's original perceptron consisted of three layers (sensory, association and response)
- Only one layer had variable weights, so his original perceptron is actually similar to a single neuron
- Rosenblatt also developed an error-correction rule to adapt these weights (the perceptron learning rule)
- He also proved that if the (two) classes were linearly separable, the algorithm would converge to a solution (the perceptron convergence theorem)



$\phi_j()$ and $g()$ are fixed thresholds
w_j are variable weights

- **Με εκμάθηση.**
- **Ψηφιακές είσοδοι**
- **Δεν συγκλίνει στη μη διαχωρισιμότητα**



$$y = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i x_i + b\right)$$

φ: συνάρτηση κατωφλίου

κανών εκμάθησης: $\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$

Widrow and Hoff, 1960

- Bernard Widrow and Ted Hoff introduced the LMS algorithm and used it to train the Adaline (ADAptive Linear Neuron)
- The Adaline was similar to the perceptron, except that it used a linear activation function instead of a threshold
- The LMS algorithm is still heavily used in adaptive signal processing

Ισχύει και για αναλογικές εισόδους

$$\varepsilon_k \triangleq d_k - \mathbf{W}_k^T \mathbf{X}_k$$

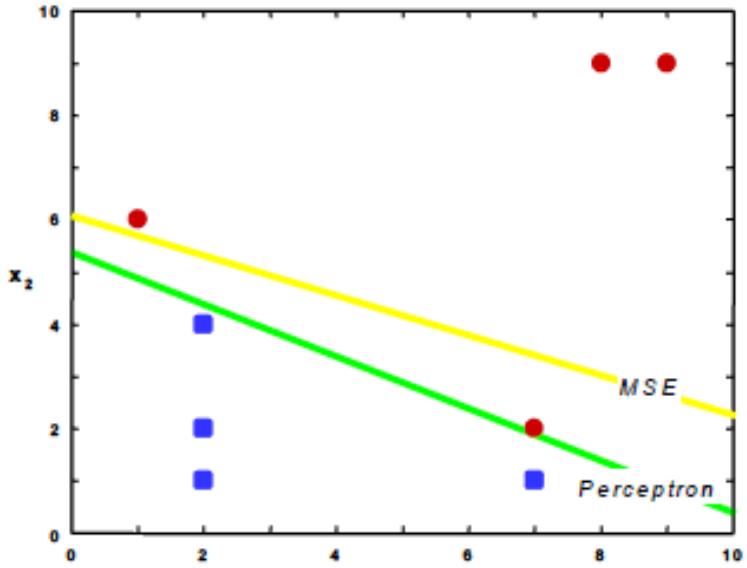
$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu\varepsilon_k \mathbf{X}_k$$

- $X1 = [(1,6),(7,2),(8,9),(9,9)]$
- $X2 = [(2,1),(2,2),(2,4),(7,1)]$

Perceptron leaning

- Assume $\eta = 0.1$ and an online update rule
- Assume $a(0) = [0.1, 0.1, 0.1]$
- SOLUTION
  - Normalize the dataset

$$Y = \begin{bmatrix} 1 & 1 & 6 \\ 1 & 7 & 2 \\ 1 & 8 & 9 \\ 1 & 9 & 9 \\ -1 & -2 & -1 \\ -1 & -2 & -2 \\ -1 & -2 & -4 \\ -1 & -7 & -1 \end{bmatrix}$$



  - Iterate through all the examples and update $a(k)$ on the ones that are misclassified
    - Y(1) ⇒ [1 1 6]*[0.1 0.1 0.1]^T>0 ⇒ no update
    - Y(2) ⇒ [1 7 2]*[0.1 0.1 0.1]^T>0 ⇒ no update

      ...
    - Y(5) ⇒ [-1 -2 -1]*[0.1 0.1 0.1]^T<0 ⇒ update a(1) = [0.1 0.1 0.1] + η[-1 -2 -1] = [0 -0.1 0]
    - Y(6) ⇒ [-1 -2 -2]*[0 -0.1 0]^T>0 ⇒ no update

      ....
    - Y(1) ⇒ [1 1 6]*[0 -0.1 0]^T<0 ⇒ update a(2) = [0 -0.1 0] + η[1 1 6] = [0.1 0 0.6]
    - Y(2) ⇒ [1 7 2]*[0.1 0 0.6]^T>0 ⇒ no update

      ...
  - In this example, the perceptron rule converges after 175 iterations to $a = [-3.5\ 0.3\ 0.7]$
  - To convince yourself this is a solution, compute Ya (you will find out that all terms are non-negative)

MSE

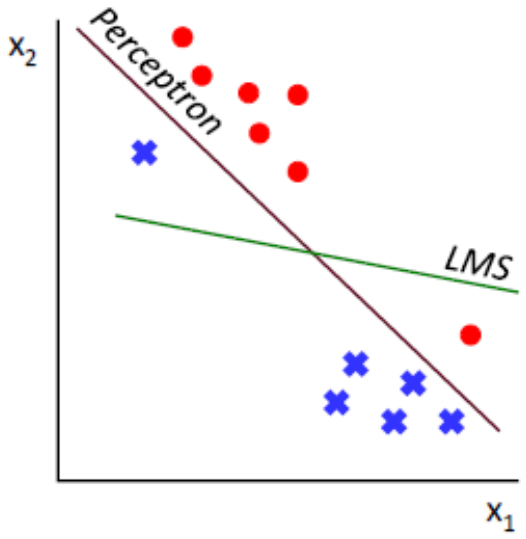
- The MSE solution is found in one shot as $a = (Y^TY)^{-1}Y^Tb = [-1.1870\ \ 0.0746\ 0.1959]$
  - For the choice of targets b = [1 1 1 1 1 1 1 1]^T
  - As you can see in the figure, the MSE solution misclassifies one of the samples

Perceptron rule

- The perceptron rule always finds a solution if the classes are linearly separable, but does not converge if the classes are non-separable

MSE criterion

- The MSE solution has guaranteed convergence, but it may not find a separating hyperplane if classes are linearly separable
  - Notice that MSE tries to minimize the sum of the squares of the distances of the training data to the separating hyperplane, as opposed to finding this hyperplane



Minsky and Papert, 1969

- In their monograph "Perceptrons", Marvin Minsky and Seymour Papert (1969) mathematically proved the limitations of Rosenblatt's perceptron and conjectured that multi-layered perceptrons would suffer from the same limitations

> The perceptron has shown itself worthy of study despite (and even because of!) its severe limitations. It has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgement that the extension to multilayer systems is sterile.

- As a result of this book, research in ANNs was almost abandoned in the 1970s
  - Only a handful of researchers continued working on ANNs, mostly outside the US
- The 1970s saw the emergence of SOMs [van der Malsburg, 1973], [Amari, 1977], [Grossberg, 1976] and associative memories: [Kohonen, 1972], [Anderson, 1972]

SOM: self organizing map

The resurgence of the early 1980s

- 1980: Steven **Grossberg**, one of the few researchers in the US that persevered despite the lack of support, establishes a new principle of self-organization called Adaptive Resonance Theory (ART) in collaboration with Gail Carpenter
- 1982: John **Hopfield** explains the operation of a certain class of recurrent ANNs (Hopfield networks) as an associative memory using statistical mechanics. Along with backprop, Hopfield's work was most responsible for the re-birth of ANNs
- 1982: Teuvo **Kohonen** presents SOMs using one- and two-dimensional lattices. Kohonen SOMs have received far more attention than van der Malsburg's work and have become the benchmark for innovations in self-organization
- 1983: **Kirkpatrick, Gelatt and Vecchi** introduced Simulated Annealing for solving combinatorial optimization problems. The concept of Simulated Annealing was later used by Ackley, Hinton and Sejnowsky (1985) to develop the Boltzmann machine (the first successful realization of multi-layered ANNs)
- 1983: **Barto, Sutton and Anderson** popularized reinforcement learning (Interestingly, it had been considered by Minsky in his 1954 PhD dissertation)
- 1984: Valentino **Braitenberg** publishes his book "Vehicles" in which he advocates for a bottom-up approach to understand complex systems: start with very elementary mechanism and build up

Rumelhart, Hinton and Williams, 1986

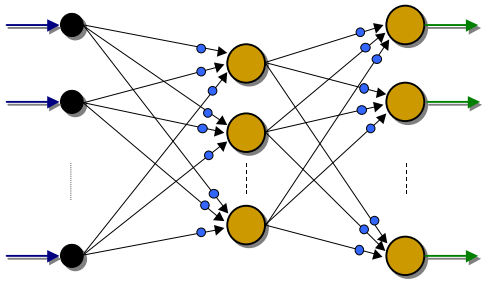
- In 1986, Rumelhart, Hinton and Williams announced the discovery of a method that allowed a network to learn to discriminate between not linearly separable classes. They called the method "backward propagation of errors", a generalization of the LMS rule
- Backprop provided the solution to the problem that had puzzled Connectionists for two decades
  - Backprop was in reality a multiple invention: David Parker (1982, 1985) and Yann LeCun (1986) published similar discoveries
  - However, the honor of discovering backprop goes to **Paul Werbos** who presented these techniques in his 1974 Ph.D. dissertation at Harvard University

Multilayer perceptrons

MLPs are feed-forward networks of simple processing units with at least ONE "hidden" layer

- Each processing unit is similar to a perceptron, except for the threshold function is replaced by a differentiable non-linearity
  - A differentiable non-linearity is required to ensure that the gradient can be computed
- The critical feature in MLPs is the non-linearity at the hidden layer
  - Note that if all neurons in an MLP had a linear activation function, the MLP could be replaced by a single layer of perceptrons, which can only solve linearly separable problems

k-διάστατος υπόχωρος εισόδου



i-διάστατος υπόχωρος εξόδου

Notation

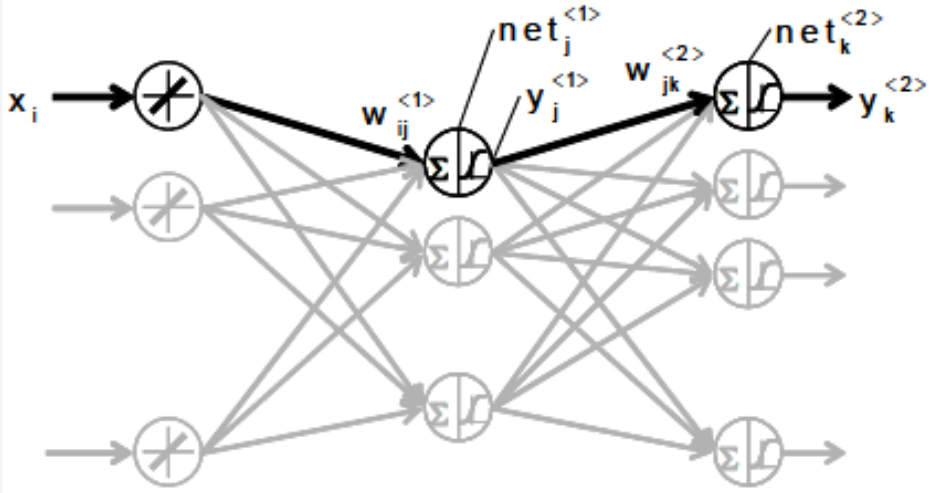
- x_i is the i^{th} input to the network
- $w_{ij}^{\langle 1 \rangle}$ is the weight connecting the i^{th} input to the j^{th} hidden neuron
- $net_j^{\langle 1 \rangle}$ is the dot product at the j^{th} hidden neuron
- $y_j^{\langle 1 \rangle}$ is the output of the j^{th} hidden neuron
- $w_{jk}^{\langle 2 \rangle}$ is the weight connecting the j^{th} hidden neuron to the k^{th} output
- $net_k^{\langle 2 \rangle}$ is the dot product at the k^{th} output neuron
- $y_k^{\langle 2 \rangle}$ is the output of the k^{th} output neuron
- t_k is the target (desired) output at the k^{th} output neuron
- For convenience, we will treat biases as regular weights with an input of 1

$$net_j^{\langle 1 \rangle} = \sum_{i=1}^{N_I} x_i w_{ij}^{\langle 1 \rangle}$$

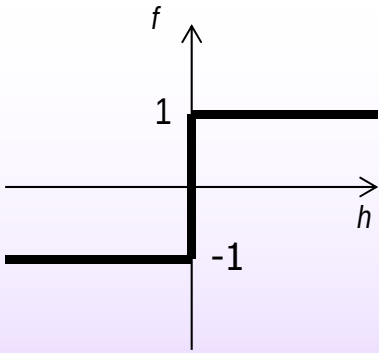
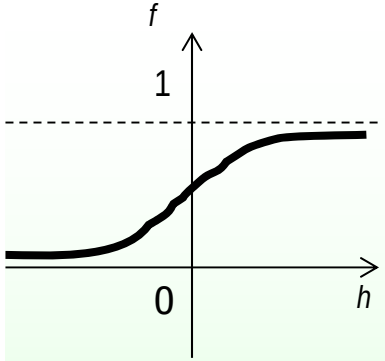
$$y_j^{\langle 1 \rangle} = f\left(net_j^{\langle 1 \rangle}\right) = \frac{1}{1 + \exp\left(-net_j^{\langle 1 \rangle}\right)}$$

$$net_k^{\langle 2 \rangle} = \sum_{j=1}^{N_H} y_j^{\langle 1 \rangle} w_{jk}^{\langle 2 \rangle}$$

$$y_k^{\langle 2 \rangle} = f\left(net_k^{\langle 2 \rangle}\right) = \frac{1}{1 + \exp\left(-net_k^{\langle 2 \rangle}\right)}$$



Συναρτήσεις ενεργοποίησης

Βηματική	Γραμμική	Μη γραμμική	Στοχαστική
 $f(h)=\begin{cases} 1, & h\geq 0 \\ -1, & h<0 \end{cases}$ ή οποιαδήποτε άλλη βηματική συνάρτηση	Έξοδος χωρίς συνάρτηση ενεργοποίησης $f(h)=h$	 $f(h)=\frac{1}{1+e^{-h}}$ ή οποιαδήποτε άλλη διαφορίσιμη συνάρτηση	Κατανομή Πιθανοτήτων $f(h)=P(h)$

The back propagation algorithm

The MLP learning problem is that of finding the weights W that capture the I/O mapping implicit in a dataset of examples

- We will use SSE at the outputs as our objective function

$$J(W) = \sum_{n=1}^{N_{EX}} \sum_{k=1}^{N_O} \frac{1}{2} \left(t_k^{(n} - y_k^{\langle 2 \rangle (n} \right)^2$$

  - where $t_k^{(n}$ is the desired target of the k^{th} output neuron for the n^{th} example
  - and $y_k^{\langle 2 \rangle (n}$ is the output of the k^{th} output neuron for the n^{th} example
- Back-prop learns the weights through gradient descent

$$w = w + \Delta w = w - \eta \frac{\partial J(W)}{\partial w}$$

- The remaining part of this lecture will be concerned with finding an expression for $\partial J(W)/\partial w$ (for each weight) in terms of what we know: the inputs x_i, the MLP outputs $y_k^{\langle 2 \rangle}$ and the desired targets t_k
  - W (upper case) denotes the set of all weights in the MLP, whereas w (lower case) denotes a generic individual weight
- For simplicity we will perform the derivation for one example ($N_{EX} = 1$), allowing us to drop the outer summation

$$J(W) = \sum_{k=1}^{N_O} \frac{1}{2} \left(t_k - y_k^{\langle 2 \rangle} \right)^2$$

Calculation of $\partial J/\partial w$ for H-O weights

- Using the **chain rule**, the derivative of $J(W)$ w.r.t. an H-O weight is

$$\frac{\partial J(W)}{\partial w_{jk}^{\langle 2\rangle}} = \frac{\partial J(W)}{\partial y_k^{\langle 2\rangle}} \frac{\partial y_k^{\langle 2\rangle}}{\partial net_k^{\langle 2\rangle}} \frac{\partial net_k^{\langle 2\rangle}}{\partial w_{jk}^{\langle 2\rangle}}$$

H-O: hidden-output

- We calculate each of these terms separately

$$\frac{\partial J(W)}{\partial y_k^{\langle 2\rangle}} = \frac{\partial}{\partial y_k^{\langle 2\rangle}} \left[\sum_{n=1}^{N_O} \frac{1}{2} \left(y_n^{\langle 2\rangle} - t_n \right)^2 \right] = \left(y_k^{\langle 2\rangle} - t_k \right)$$

$$\frac{\partial y_k^{\langle 2\rangle}}{\partial net_k^{\langle 2\rangle}} = \frac{\partial}{\partial net_k^{\langle 2\rangle}} \left[\frac{1}{1+\exp\left(-net_k^{\langle 2\rangle}\right)} \right] = \frac{\exp\left(-net_k^{\langle 2\rangle}\right)}{\left(1+\exp\left(-net_k^{\langle 2\rangle}\right)\right)^2} = \left(1 - y_k^{\langle 2\rangle}\right) y_k^{\langle 2\rangle}$$

$$\frac{\partial net_k^{\langle 2\rangle}}{\partial w_{jk}^{\langle 2\rangle}} = \frac{\partial}{\partial w_{jk}^{\langle 2\rangle}} \left[\sum_{n=1}^{N_H} w_{nk}^{\langle 2\rangle} y_n^{\langle 1\rangle} \right] = y_j^{\langle 1\rangle}$$

- Merging all these derivatives yields

$$\frac{\partial J(W)}{\partial w_{jk}^{\langle 2\rangle}} = \left(y_k^{\langle 2\rangle} - t_k \right) \left(1 - y_k^{\langle 2\rangle} \right) y_k^{\langle 2\rangle} y_j^{\langle 1\rangle}$$

  - For the bias weights, use $y_j^{\langle 1\rangle} = 1$ in the expression above

Calculation of $\partial J/\partial w$ for I-H weights

I-H: input-hidden

– Using the chain rule, the derivative of $J(W)$ w.r.t. an I-H weight is

$$\frac{\partial J(W)}{\partial w_{ij}^{\langle 1\rangle}} = \frac{\partial J(W)}{\partial y_j^{\langle 1\rangle}} \frac{\partial y_j^{\langle 1\rangle}}{\partial net_j^{\langle 1\rangle}} \frac{\partial net_j^{\langle 1\rangle}}{\partial w_{ij}^{\langle 1\rangle}}$$

– The 2nd and 3rd terms are easy to calculate from our previous result

$$\frac{\partial y_j^{\langle 1\rangle}}{\partial net_j^{\langle 1\rangle}} = \left(1 - y_j^{\langle 1\rangle}\right) y_j^{\langle 1\rangle}$$

$$\frac{\partial net_j^{\langle 1\rangle}}{\partial w_{ij}^{\langle 1\rangle}} = x_i$$

– The first term, however, is not straightforward since we do not know what the outputs of the hidden neurons ought to be

– This is known as the **credit assignment problem** [Minski,1961], which puzzled connectionists for two decades

The solution lies in realizing that H neurons do not make errors, they only contribute to errors at the output nodes

- The derivative of the error with respect to a hidden node's output is therefore the sum of that hidden node's contribution to the errors of all the output neurons

$$\frac{\partial J(W)}{\partial y_j^{\langle 1\rangle}} = \sum_{n=1}^{N_O} \frac{\partial J(W)}{\partial y_n^{\langle 2\rangle}} \frac{\partial y_n^{\langle 2\rangle}}{\partial net_n^{\langle 2\rangle}} \frac{\partial net_n^{\langle 2\rangle}}{\partial y_j^{\langle 1\rangle}}$$

- The first two terms in the summation are known from our earlier derivation

$$\frac{\partial J(W)}{\partial y_n^{\langle 2\rangle}} \frac{\partial y_n^{\langle 2\rangle}}{\partial net_n^{\langle 2\rangle}} = \left(y_n^{\langle 2\rangle} - t_n\right)\left(1 - y_n^{\langle 2\rangle}\right) y_n^{\langle 2\rangle} = p_n$$

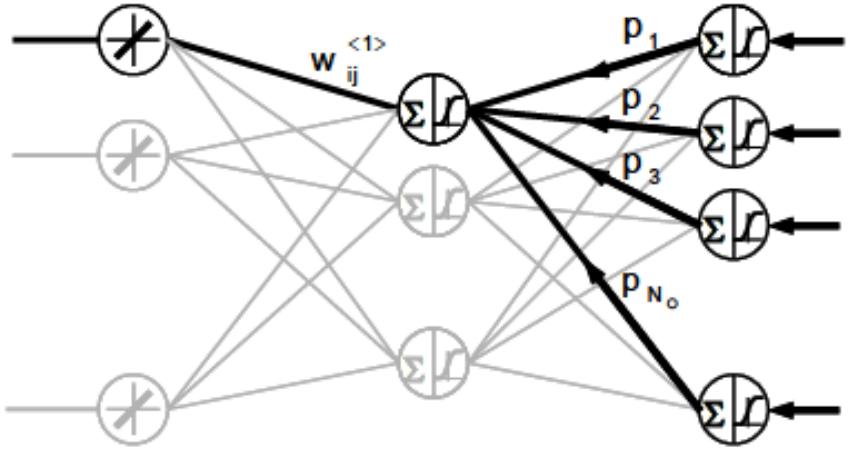
- The last term in the summation is

$$\frac{\partial net_n^{\langle 2\rangle}}{\partial y_j^{\langle 1\rangle}} = w_{jn}^{\langle 2\rangle}$$

– Merging these derivatives yields

$$\frac{\partial J(W)}{\partial y_j^{\langle 1\rangle}} = \sum_{n=1}^{N_O} \frac{\partial J(W)}{\partial y_n^{\langle 2\rangle}} \frac{\partial y_n^{\langle 2\rangle}}{\partial net_n^{\langle 2\rangle}} \frac{\partial net_n^{\langle 2\rangle}}{\partial y_j^{\langle 1\rangle}} = \sum_{n=1}^{N_O} \underbrace{\left(y_n^{\langle 2\rangle} - t_n\right)\left(1 - y_n^{\langle 2\rangle}\right) y_n^{\langle 2\rangle}}_{p_n} w_{jn}^{\langle 2\rangle}$$

– Notice how this can be viewed as propagating the error term p_n backwards trough the H-O weighs (hence the term backprop)



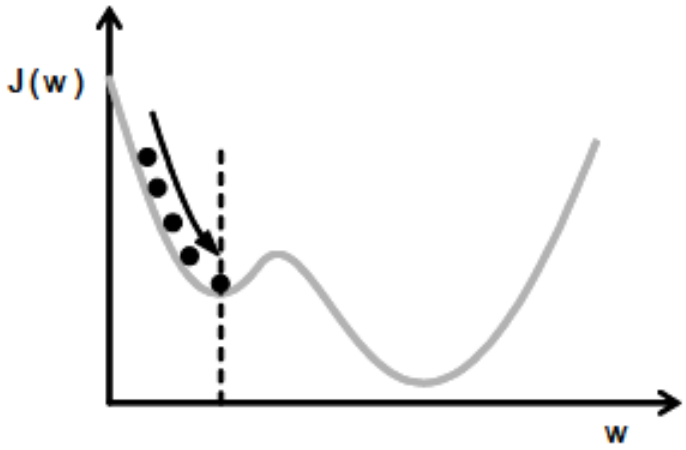
– And the final expression of $\partial J(W)/\partial w$ for I-H weights is

$$\frac{\partial J(W)}{\partial w_{ij}^{\langle 1\rangle}} = \left[\sum_{n=1}^{N_O} \left(y_n^{\langle 2\rangle} - t_n\right)\left(1 - y_n^{\langle 2\rangle}\right) y_n^{\langle 2\rangle} w_{jn}^{\langle 2\rangle}\right]\left(1 - y_j^{\langle 1\rangle}\right) y_j^{\langle 1\rangle} x_i$$

- For the bias weights, use $x_i = 1$ in the expression above

One of the main limitations of back-prop is local minima

- When the gradient descent algorithm reaches a local minimum, the gradient becomes zero and the weights converge to a sub-optimal solution
- A very popular method to avoid local minima is to compute a temporal average direction in which the weights have been moving recently



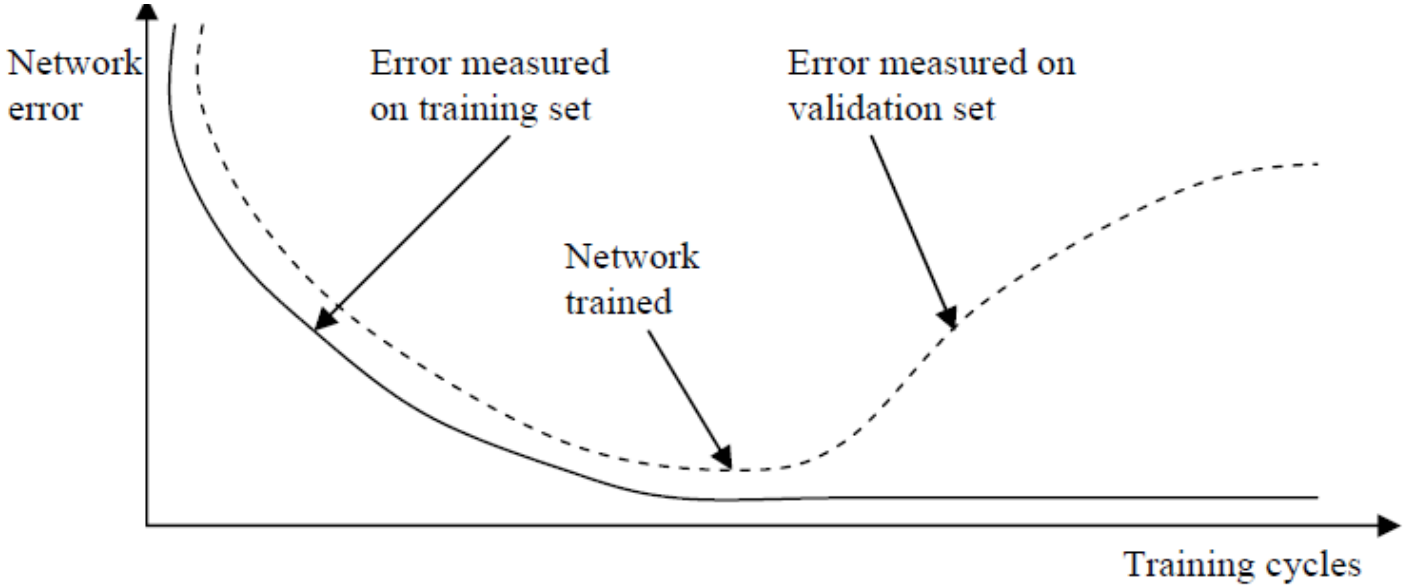
- An easy way to implement this is by using an exponential average

$$\Delta w(n) = \mu[\Delta w(n-1)] + (1-\mu)\left[\eta \frac{\partial J(w)}{\partial w}\right]$$

- The term μ is called the momentum
  - The momentum has a value between 0 and 1 (typically 0.9)
  - The closer to 1, the stronger the influence of the instantaneous steepest descent direction

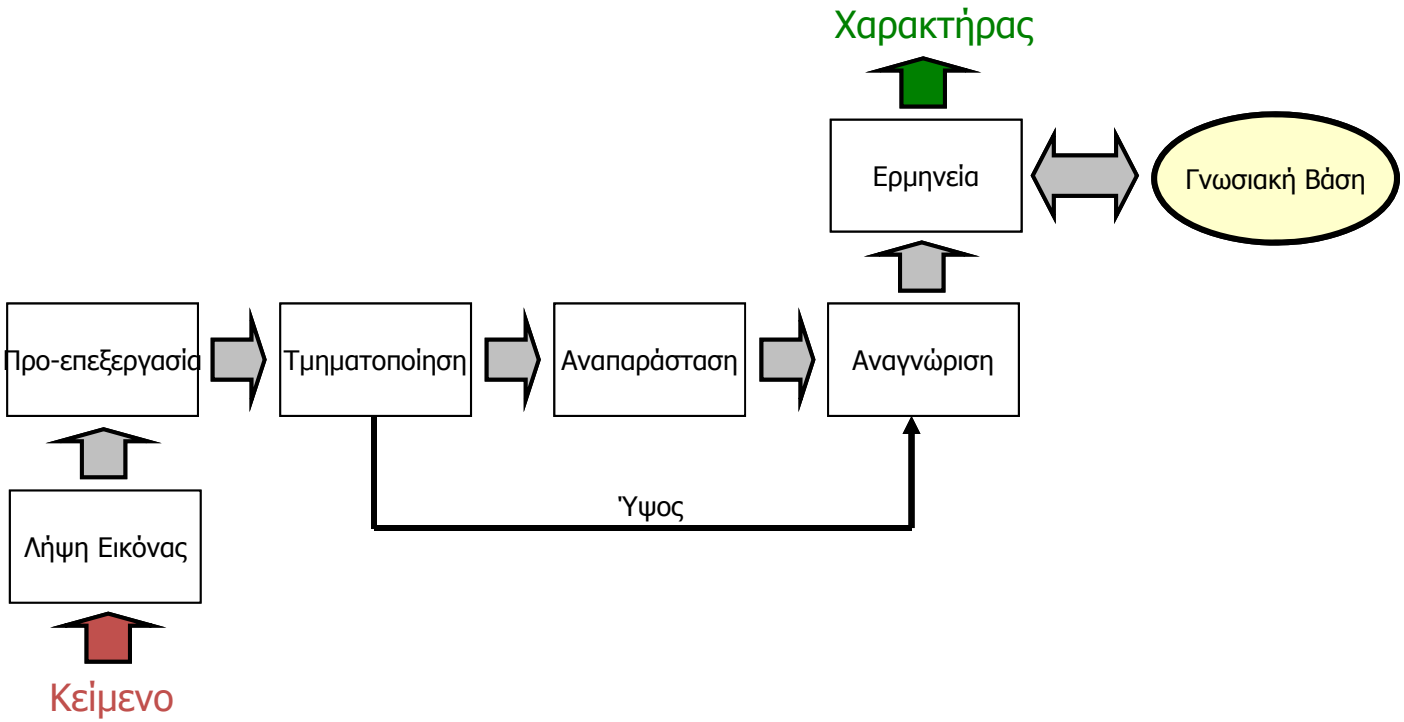
Σταμάτημα

There is a better way of working out when to stop network training - which is to use a Validation Set. This stops the network overtraining (becoming too accurate, which can lessen its performance). It does this by having a second set of patterns which are noisy versions of the training set (but aren't used for training themselves). Each time after the network has trained; this set (called the Validation Set) is used to calculate an error. When the error becomes low the network stops. Figure 3.8 shows the idea.



When the network has fully trained, the Validation Set error reaches a minimum. When the network is overtraining (becoming too accurate) the validation set error starts rising. If the network overtrains, it won't be able to handle noisy data so well.

Αναγνώριση χαρακτήρων



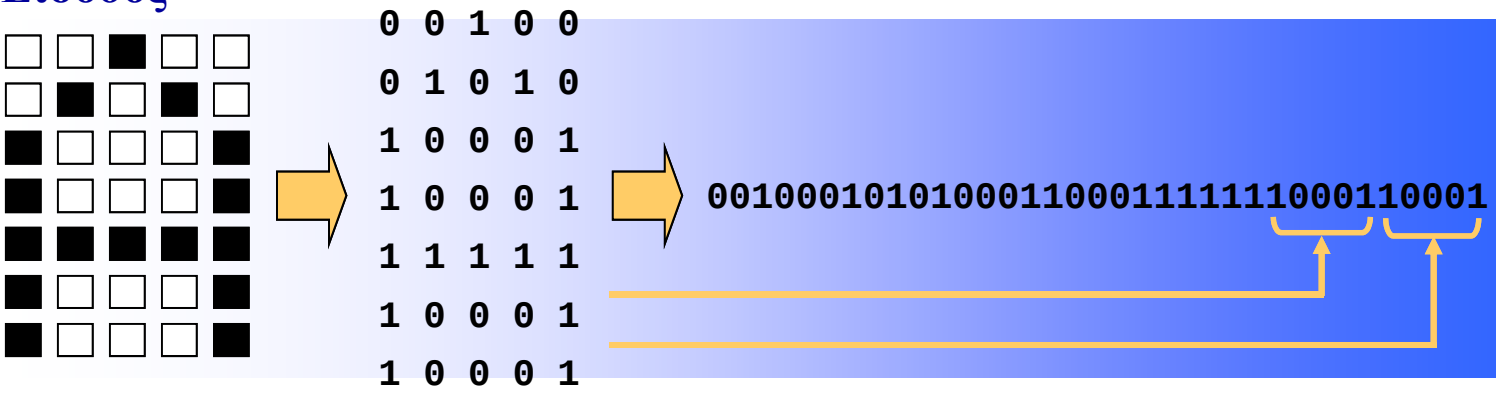
Σπύρου Σπυρίδων - Δρ Σ. Κ. Λεβέντης
Τ.Ε.Ι Πειραιά - PeLAB

7. Τεχνητά νευρωνικά δίκτυα

Σκοπός

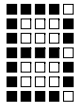
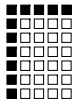
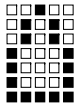
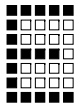
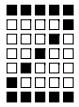
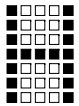
Αναγνώριση των 12 πρώτων κεφαλαίων χαρακτήρων του Ελληνικού αλφάβητου

Είσοδος

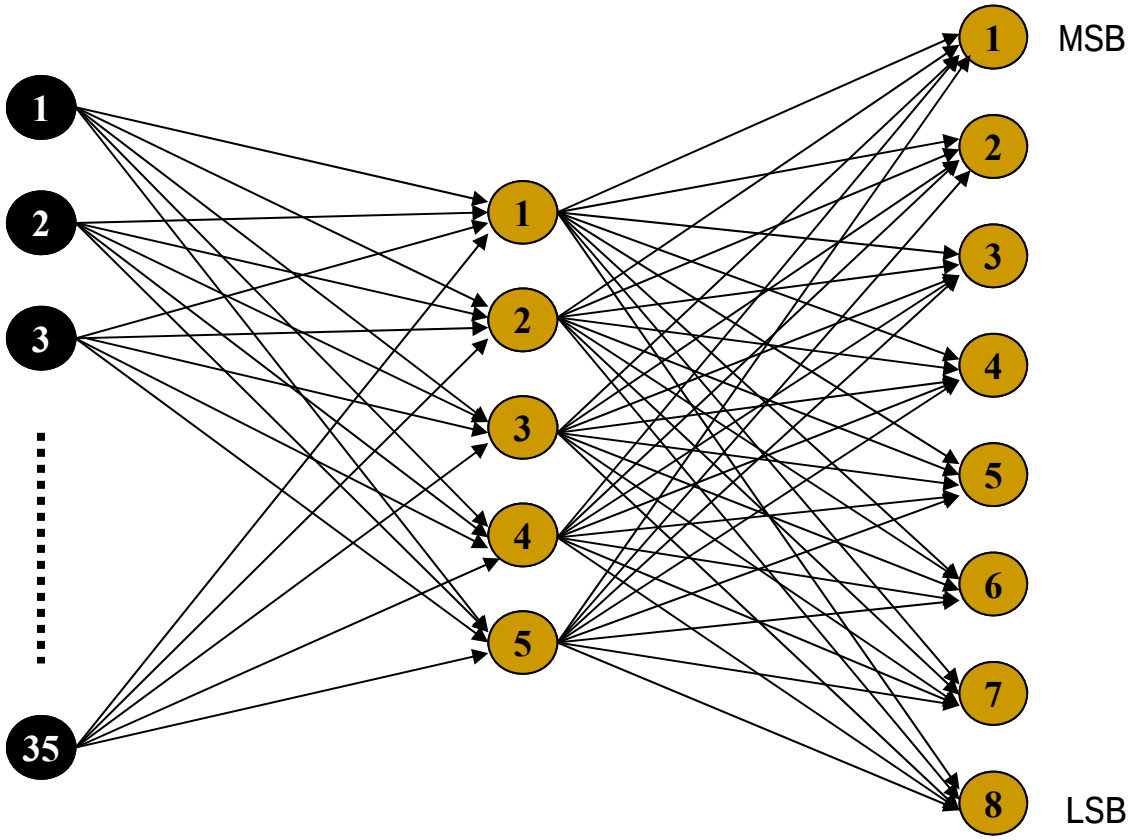


Έξοδος

Οκταψήφια δυαδική αναπαράσταση σύμφωνα με το πρότυπο ΕΛΟΤ 928

	Ψηφιοποιημένος χαρακτήρας	Δείγμα εισόδου	Κωδικοποίηση ΕΛΟΤ 928 (επιθυμητό δείγμα εξόδου)
Α		00100010101000110001111111000110001	11000001
Β		11110100011000111110100011000111110	11000010
Γ		11111100001000010000100001000010000	11000011
Δ		00100010100101010001100011000111111	11000100
Ε		11111100001000011110100001000011111	11000101
Ζ		11111000010001000100010001000011111	11000110
Η		10001100011000111111100011000110001	11000111

Σχεδίαση


1
2
3
35
1
2
3
4
5
1
2
3
4
5
6
7
8
MSB
LSB
Στοιχεία Εισόδου
Νευρώνια Εξόδου

Εκπαίδευση

Σύνολο Εκπαίδευσης

```
00100 01010 10001 10001 11111 10001 10001  11000001
11110 10001 10001 11110 10001 10001 11110  11000010
11111 10000 10000 10000 10000 10000 10000  11000011
00100 01010 01010 10001 10001 10001 11111  11000100
11111 10000 10000 11110 10000 10000 11111  11000101
11111 00001 00010 00100 01000 10000 11111  11000110
10001 10001 10001 11111 10001 10001 10001  11000111
01110 10001 10001 11111 10001 10001 01110  11001000
01110 00100 00100 00100 00100 00100 01110  11001001
10001 10010 10100 11000 10100 10010 10001  11001010
00100 01010 01010 10001 10001 10001 10001  11001011
10001 11011 10101 10001 10001 10001 10001  11001100
```

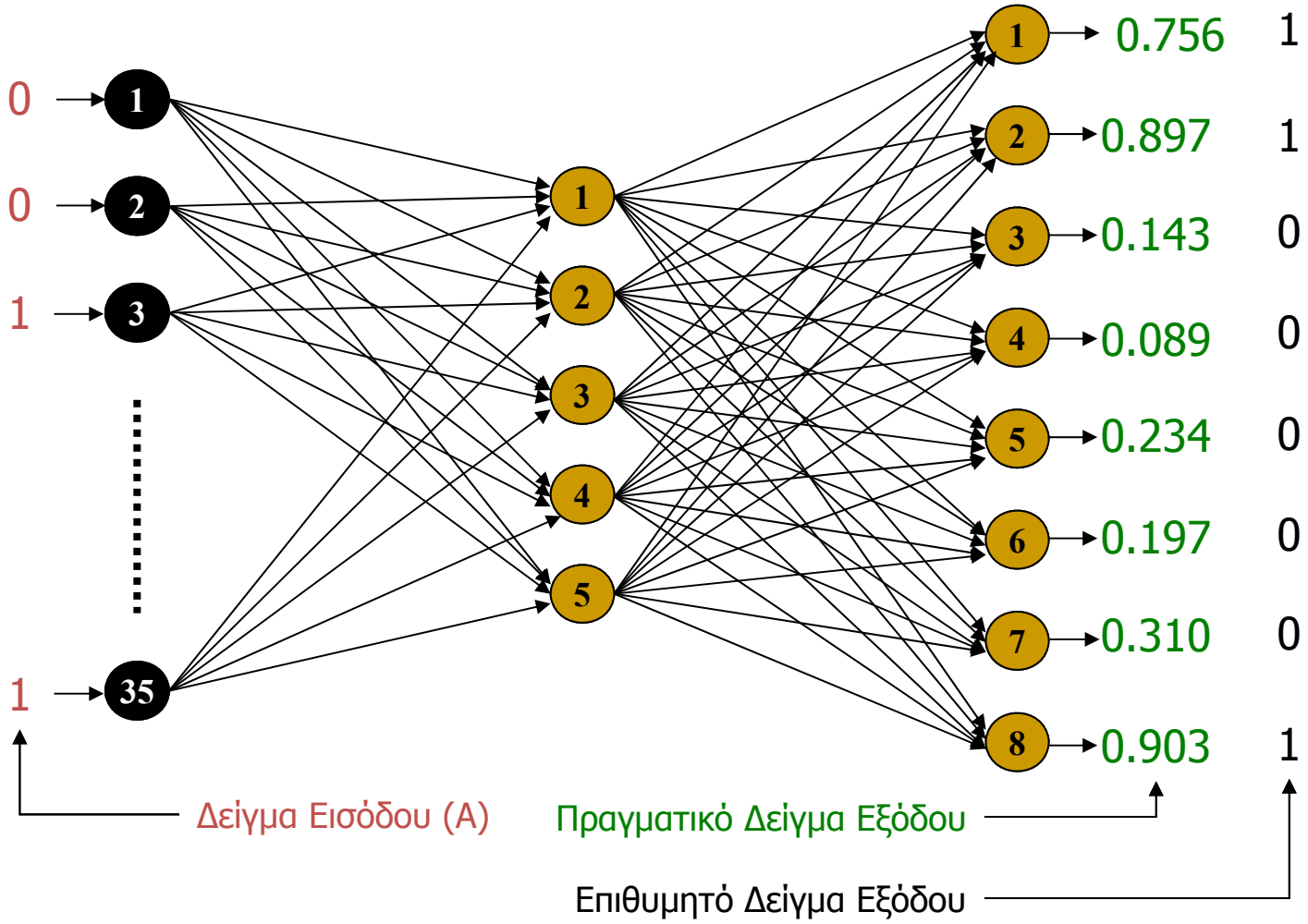
Παράμετροι Εξομοίωσης

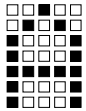
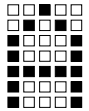
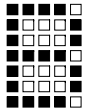
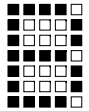
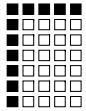
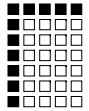
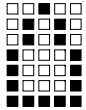
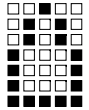
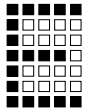
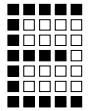
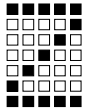
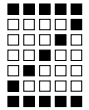
1	*Λειτουργία εκπαίδευσης*
0.1	*Ανέχεια σφάλματος*
0.02	*Ρυθμός μάθησης*
0.07	*Παράμετρος ορμής*
0.5	*Παράγοντας θορύβου*
10000	*Μέγιστος αριθμός εποχών*
0	*Παραγωγή ψευδοτυχαίων βαρών εκκίνησης*
3	*Πλήθος επιπέδων*
35	*Πλήθος στοιχείων εισόδου*
5	*Πλήθος κρυμμένων νευρωνίων*
8	*Πλήθος νευρωνίων εξόδου*

Αποτελέσματα Εξομοίωσης

Μέσο σφάλμα ανά εποχή:	1.15241
Σφάλμα τελευταίας εποχής:	0.337198
Μέσο σφάλμα ανά δείγμα τελευταίας εποχής:	0.0973406
Σύνολο εποχών:	794
Σύνολο δειγμάτων:	9528

Στιγμιότυπο εκπαίδευσης



Χαρακτήρας εισόδου	Έξοδος δικτύου	Έξοδος συγκριτών	Χαρακτήρας εξόδου
	0.999 0.999 0.000 0.000 0.007 0.022 0.000 0.999	11000001	
	0.998 0.998 0.000 0.001 0.204 0.006 0.984 0.041	11000010	
	0.999 0.999 0.000 0.000 0.000 0.051 0.992 0.994	11000011	
	0.999 0.999 0.000 0.000 0.034 0.999 0.060 0.030	11000100	
	0.998 0.998 0.001 0.001 0.000 0.994 0.041 0.922	11000101	
	0.999 0.999 0.001 0.000 0.035 0.999 0.786 0.124	11000110	

The pattern recognition methods covered in class up to this point have focused on the issue of classification

- A pattern consisted of a pair of variables $\{x, \omega\}$ where
  - x was a collection of observations or features (feature vector)
  - ω was the concept behind the observation (label)
- Such pattern recognition problems are called supervised (training with a teacher) since the system is given BOTH the feature vector and the correct answer

In the next three lectures we investigate a number of methods that operate on unlabeled data

- Given a collection of feature vectors $X = \{x^{(1}, x^{(2} \ldots, x^{(N}\}$ without class labels ω_i, these methods attempt to build a model that captures the structure of the data
- These methods are called unsupervised (training without a teacher) since they are not provided the correct answer

Approaches to unsupervised learning

Parametric (mixture models)

- These methods model the underlying class-conditional densities with a mixture of parametric densities, and the objective is to find the model parameters

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

- These methods are closely related to parameter estimation (L6)
- Mixture models are the subject of this lecture

Non-parametric unsupervised learning

- No density functions are considered in these methods
- Instead, we are concerned with finding natural groupings (clusters) in a dataset

Non-parametric clustering involves three steps

- Defining a measure of (dis)similarity between examples
- Defining a criterion function for clustering
- Defining an algorithm to minimize (or maximize) the criterion function

Proximity measures

Definition of metric

- A measuring rule $d(x, y)$ for the distance between two vectors x and y is considered a metric if it satisfies the following properties

$$d(x, y) \geq d_0$$
$$d(x, y) = d_0 \; iff \; x = y$$
$$d(x, y) = d(y, x)$$
$$d(x, y) \leq d(x, z) + d(z, y)$$

  - If the metric has the property $d(ax, ay) = |a| d(x, y)$ then it is called a norm and denoted $d(x, y) = \|x - y\|$

The most general form of distance metric is the power norm

$$\|x - y\|_{p/r} = \left(\sum_{i=1}^{D} |x_i - y_i|^p \right)^{1/r}$$

- p controls the weight placed on any dimension dissimilarity, whereas r controls the distance growth of patterns that are further apart
- Notice that the definition of norm must be relaxed, allowing a power factor for $|a|$

Most commonly used metrics are derived from the power norm

- Minkowski metric (L_k norm)

$$\|x - y\|_k = \left(\sum_{i=1}^{D} |x_i - y_i|^k\right)^{1/k}$$

  - The choice of an appropriate value of k depends on the amount of emphasis that you would like to give to the larger differences between dimensions

- Manhattan or city-block distance (L_1 norm)

$$\|x - y\|_{c-b} = \sum_{i=1}^{D} |x_i - y_i|$$

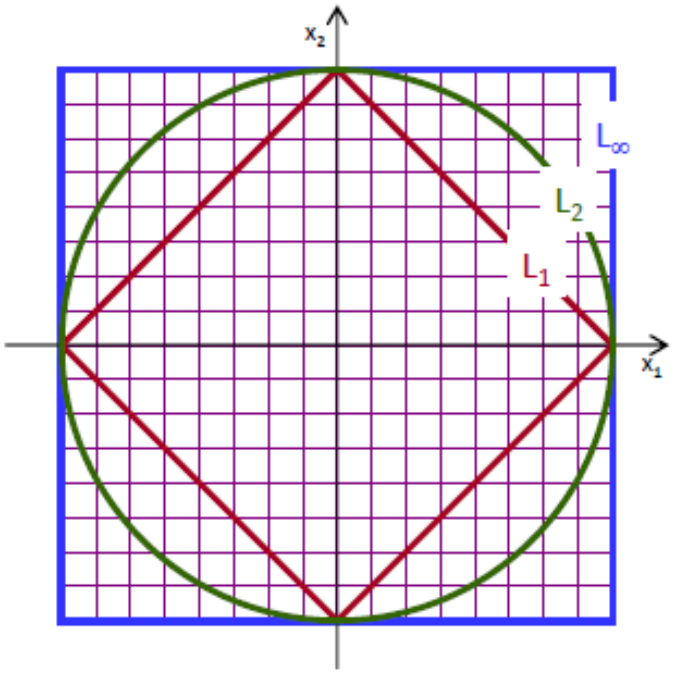
  - When used with binary vectors, the L1 norm is known as the Hamming distance

- Euclidean norm (L_2 norm)

$$\|x - y\|_e = \left(\sum_{i=1}^{D} |x_i - y_i|^2\right)^{1/2}$$

- Chebyshev distance (L_∞ norm)

$$\|x - y\|_c = \max_{1 \le i \le D} |x_i - y_i|$$



Contours of equal distance

Other metrics are also popular

- Quadratic distance

$$d(x,y) = \sqrt{(x-y)^T B (x-y)}$$

  - The Mahalanobis distance is a particular case of this distance

- Canberra metric (for non-negative features)

$$d_{ca}(x,y) = \sum_{i=1}^{D} \frac{|x_i - y_i|}{x_i + y_i}$$

- Non-linear distance

$$d_N(x,y) = \begin{cases} 0 & if\ d_e(x,y) < T \\ H & o.w. \end{cases}$$

  - where T is a threshold and H is a distance
  - An appropriate choice for H and T for feature selection is that they should satisfy

$$H = \frac{\Gamma\left(\frac{p}{2}\right)}{T^P 2\sqrt{\pi^P}}$$

  - and that T satisfies the unbiasedness and consistency conditions of the Parzen estimator: $T^P N \to \infty, T \to 0\ as\ N \to \infty$

The above distance metrics are measures of dissimilarity

Some measures of similarity also exist

- Inner product

$$s_{INNER}(x, y) = x^T y$$

  - The inner product is used when the vectors x and y are normalized, so that they have the same length

- Correlation coefficient

$$s_{CORR}(x, y) = \frac{\sum_{i=1}^{D}(x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^{D}(x_i - \bar{x})^2 \sum_{i=1}^{D}(y_i - \bar{y})^2\right]^{1/2}}$$

- Tanimoto measure (for binary-valued vectors)

$$s_T(x, y) = \frac{x^T y}{|x|^2 + |y|^2 - x^T y}$$

Criterion function for clustering

Once a (dis)similarity measure has been determined, we need to define a criterion function to be optimized

- The most widely used clustering criterion is the sum-of-square-error

$$J_{MSE} = \sum_{i=1}^{C} \sum_{x \in \omega_i} |x - \mu_i|^2 \quad \text{where} \quad \mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

  - This criterion measures how well the data set $X = \{x^{(1} \dots x^{(N}\}$ is represented by the cluster centers $\mu = \{\mu^{(1} \dots \mu^{(C}\}$ $(C < N)$
  - Clustering methods that use this criterion are called minimum variance
- Other criterion functions exist, based on the scatter matrices used in Linear Discriminant Analysis
  - For details, refer to [Duda, Hart and Stork, 2001]

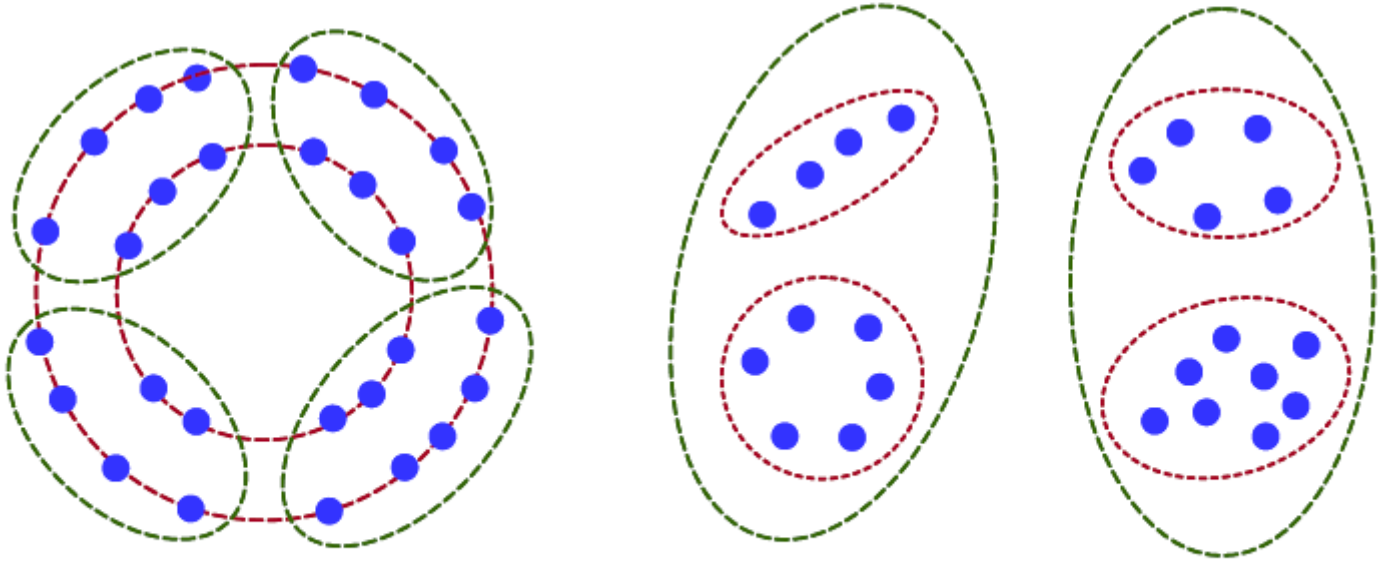
Cluster validity

The validity of the final cluster solution is highly subjective

- This is in contrast with supervised training, where a clear objective function is known: Bayes risk
- Note that the choice of (dis)similarity measure and criterion function will have a major impact on the final clustering produced by the algorithms

Example

- Which are the meaningful clusters in these cases?
- How many clusters should be considered?



- A number of quantitative methods for cluster validity are proposed in [Theodoridis and Koutrombas, 1999]

Iterative optimization

Once a criterion function has been defined, we must find a partition of the data set that minimizes the criterion

- Exhaustive enumeration of all partitions, which guarantees the optimal solution, is unfeasible
  - For example, a problem with 5 clusters and 100 examples yields 10^{67} partitionings

The common approach is to proceed in an iterative fashion

1) Find some reasonable initial partition and then
2) Move samples from one cluster to another in order to reduce the criterion function

These iterative methods produce sub-optimal solutions but are computationally tractable

We will consider two groups of iterative methods

- Flat clustering algorithms
  - These algorithms produce a set of disjoint clusters
  - Two algorithms are widely used: k-means and ISODATA
- Hierarchical clustering algorithms:
  - The result is a hierarchy of nested clusters
  - These algorithms can be broadly divided into agglomerative and divisive approaches

The k-means algorithm

Method

– k-means is a simple clustering procedure that attempts to minimize the criterion function J_{MSE} in an iterative fashion

$$J_{MSE} = \sum_{i=1}^{C} \sum_{x \in \omega_i} |x - \mu_i|^2 \quad \text{where} \quad \mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

1. Define the number of clusters
2. Initialize clusters by
   - an arbitrary assignment of examples to clusters or
   - an arbitrary set of cluster centers (some examples used as centers)
3. Compute the sample mean of each cluster
4. Reassign each example to the cluster with the nearest mean
5. If the classification of all samples has not changed, stop, else go to step 3

– It can be shown (L14) that k-means is a particular case of the EM algorithm for mixture models

8. Ομαδοποίηση

ISODATA

Iterative Self-Organizing Data Analysis (ISODATA)

- An extension to the k-means algorithm with some heuristics to automatically select the number of clusters

ISODATA requires the user to select a number of parameters

- N_{MIN_EX} minimum number of examples per cluster
- N_D desired (approximate) number of clusters
- σ_S^2 maximum spread parameter for splitting
- D_{MERGE} maximum distance separation for merging
- N_{MERGE} maximum number of clusters that can be merged

The algorithm works in an iterative fashion

1) Perform k-means clustering
2) Split any clusters whose samples are sufficiently dissimilar
3) Merge any two clusters sufficiently close
4) Go to 1)

8. Ομαδοποίηση

1. Select an initial number of clusters N_C and use the first N_C examples as cluster centers $\mu_k, k = 1..NC$
2. Assign each example to the closest cluster
   a. Exit the algorithm if the classification of examples has not changed
3. Eliminate clusters that contain less than N_{MIN_EX} examples and
   a. Assign their examples to the remaining clusters based on minimum distance
   b. Decrease N_C accordingly
4. For each cluster k,
   a. Compute the center μ_k as the sample mean of all the examples assigned to that cluster
   b. Compute the average distance between examples and cluster centers

$$d_{avg} = \frac{1}{N}\sum_{k=1}^{NC} N_k d_k \text{ and } d_k = \frac{1}{N_k}\sum_{x\in\omega_k}|x-\mu_k|$$

   c. Compute the variance of each axis and find the axis n^* with maximum variance $\sigma_k^2(n^*)$
6. For each cluster k with $\sigma_k^2(n^*) > \sigma_S^2$, if $\{d_k > d_{AVG}$ *and* $N_k > 2NMIN_{EX} + 1\}$ or $\{NC < ND/2\}$
   a. Split that cluster into two clusters where the two centers μ_{k1} and μ_{k2} differ only in the coordinate n^*
      i. $\mu_{k1}(n*) = \mu_k(n*) + \sigma_k(n*)$ (all other coordinates remain the same, $0 < \varepsilon < 1$)
      ii. $\mu_{k2}(n*) = \mu_k(n*) - \sigma_k(n*)$ (all other coordinates remain the same, $0 < \varepsilon < 1$)
   b. Increment N_C accordingly
   c. Reassign the cluster's examples to one of the two new clusters based on minimum distance to cluster centers
7. If $N_C > 2ND$ then
   a. Compute all distances $D_{ij} = d(\mu_i, \mu_j)$
   b. Sort D_{ij} in decreasing order
   b. For each pair of clusters sorted by D_{ij}, if (1) neither cluster has been already merged, (2) $D_{ij} < DMER_{GE}$ and (3) not more than N_{MERGE} pairs of clusters have been merged in this loop, then
      i. Merge i^{th} and j^{th} clusters
      ii. Compute the cluster center $\mu' = \frac{N_i\mu_i + N_j\mu_j}{N_i + N_j}$
      iii. Decrement N_C accordingly
8. Go to step 1

[Therrien, 1989]

8. Ομαδοποίηση

ISODATA has been shown to be an extremely powerful heuristic

Some of its advantages are

- Self-organizing capabilities
- Flexibility in eliminating clusters that have very few examples
- Ability to divide clusters that are too dissimilar
- Ability to merge clusters that are sufficiently similar

However, it suffers from the following limitations

- Data must be linearly separable (long narrow or curved clusters are not handled properly)
- It is difficult to know a priori the "optimal" parameters
- Performance is highly dependent on these parameters
- For large datasets and large number of clusters, ISODATA is less efficient than other linear methods
- Convergence is unknown, although it appears to work well for non-overlapping clusters

In practice, ISODATA is run multiple times with different values of the parameters and the clustering with minimum SSE is selected

8. Ομαδοποίηση

Hierarchical clustering

k-means and ISODATA create disjoint clusters, resulting in a flat data representation

- However, sometimes it is desirable to obtain a hierarchical representation of data, with clusters and sub-clusters arranged in a tree-structured fashion
- Hierarchical representations are commonly used in the sciences (e.g., biological taxonomy)

Hierarchical clustering methods can be grouped in two general classes

- Agglomerative
  - Also known as bottom-up or merging
  - Starting with N singleton clusters, successively merge clusters until one cluster is left
- Divisive
  - Also known as top-down or splitting
  - Starting with a unique cluster, successively split the clusters until N singleton examples are left

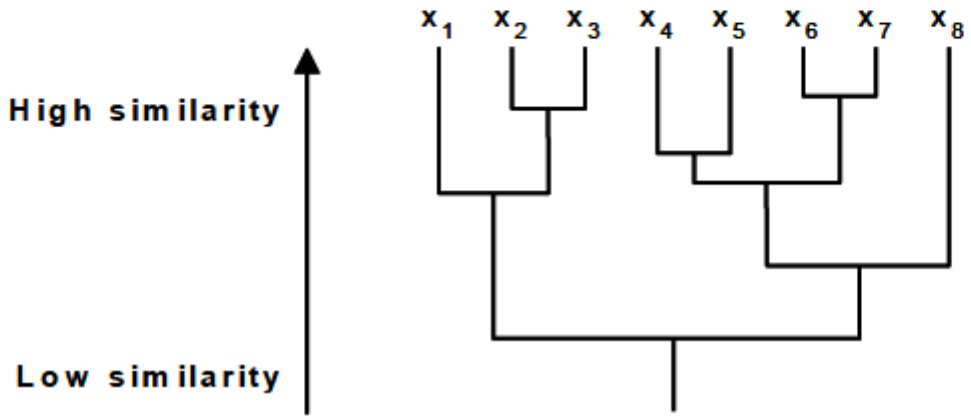
Dendrograms

A binary tree that shows the structure of the clusters

- Dendrograms are the preferred representation for hierarchical clusters
  - In addition to the binary tree, the dendrogram provides the similarity measure between clusters (the vertical axis)
- An alternative representation is based on sets

$$\{\{x_1, \{x_2, x_3\}\}, \{\{\{x_4, x_5\}, \{x_6, x_7\}\}, x_8\}\}$$

  - However, unlike the dendrogram, sets cannot express quantitative information



Divisive clustering

Define

- N_C Number of clusters
- N_{EX} Number of examples

1. Start with one large cluster
2. Find "worst" cluster
3. Split it
4. If $N_C < NEX$ go to 2

How to choose the "worst" cluster

- Largest number of examples
- Largest variance
- Largest sum-squared-error…

How to split clusters

- Mean-median in one feature direction
- Perpendicular to the direction of largest variance…

The computations required by divisive clustering are more intensive than for agglomerative clustering methods

- For this reason, agglomerative approaches are more popular

Agglomerative clustering

Define

- N_C Number of clusters
- N_{EX} Number of examples

1. Start with N_{EX} singleton clusters
2. Find nearest clusters
3. Merge them
4. If $N_C > 1$ go to 2

How to find the "nearest" pair of clusters

- Minimum distance $$d_{\min}(\omega_i, \omega_j) = \min_{\substack{x \in \omega_i \\ y \in \omega_j}} \|x - y\|$$
- Maximum distance $$d_{\max}(\omega_i, \omega_j) = \max_{\substack{x \in \omega_i \\ y \in \omega_j}} \|x - y\|$$
- Average distance $$d_{\text{avg}}(\omega_i, \omega_j) = \frac{1}{N_i N_j} \sum_{x \in \omega_i} \sum_{y \in \omega_j} \|x - y\|$$
- Mean distance $$d_{\text{mean}}(\omega_i, \omega_j) = \|\mu_i - \mu_j\|$$

Minimum distance

- When d_{min} is used to measure distance between clusters, the algorithm is called the nearest-neighbor or single-linkage clustering algorithm
- If the algorithm is allowed to run until only one cluster remains, the result is a minimum spanning tree (MST)
- This algorithm favors elongated classes

Maximum distance

- When d_{max} is used to measure distance between clusters, the algorithm is called the farthest-neighbor or complete-linkage clustering algorithm
- From a graph-theoretic point of view, each cluster constitutes a complete sub-graph
- This algorithm favors compact classes

Average and mean distance

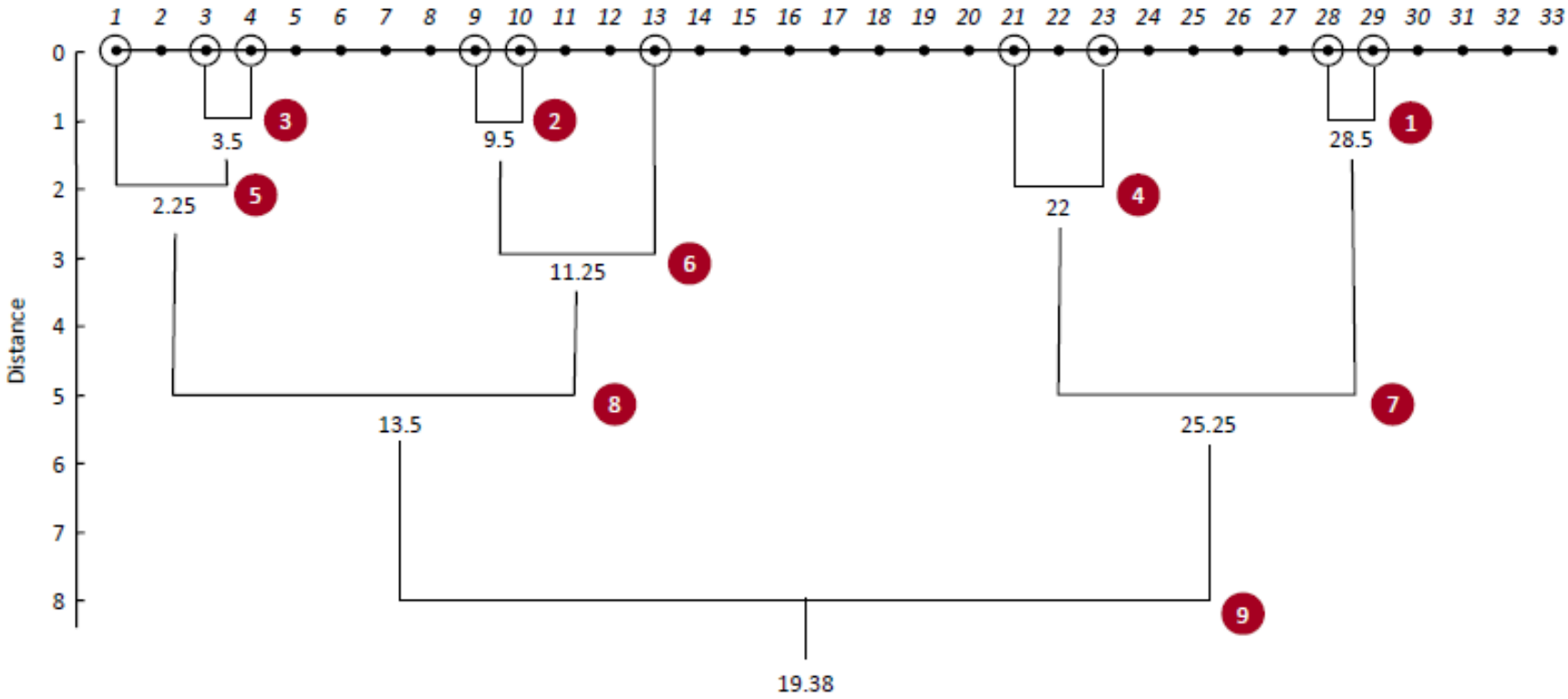
- d_{min} and d_{max} are extremely sensitive to outliers since their measurement of between-cluster distance involves minima or maxima
- d_{ave} and d_{mean} are more robust to outliers
- Of the two, d_{mean} is more attractive computationally
  - Notice that d_{ave} involves the computation of $N_i N_j$ pairwise distances

Example

– Perform agglomerative clustering on X using the single-linkage metric

$$X = \{1, 3, 4, 9, 10, 13, 21, 23, 28, 29\}$$

- In case of ties, always merge the pair of clusters with the largest mean
- Indicate the order in which the merging operations occur



1. MEDEA

Quality control for household appliances by on-line evaluation of mechanical defects.

Standards, Measurements and Testing (1/1/96-31-12-98).

AEA (Applicazioni Elettroniche Avanzante Srl.), Italy

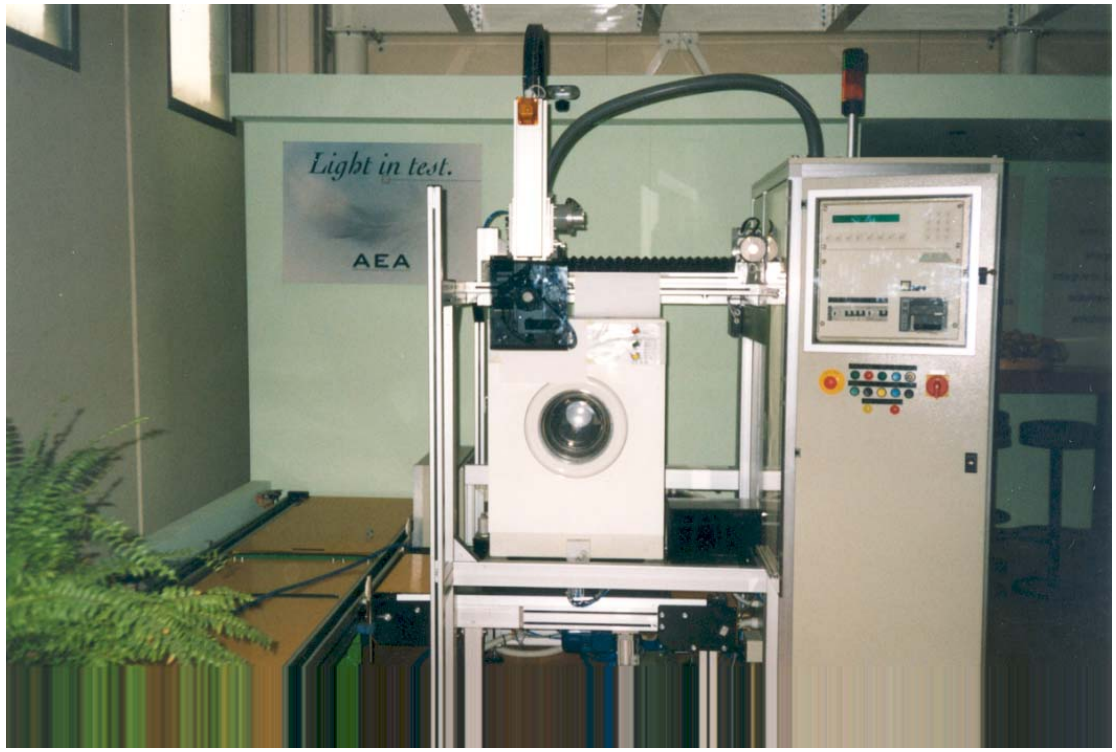
MIT (Management Intelligenter Technologien), Germany

CEA-LETI-DMITEC-SIA (Commissariat à l'Energie Atomique), France

CSO-MESURE (Capteurs et systèmes optiques de mesure), France

Universita degli Studi Ancona, Department of Mechanics, Italy

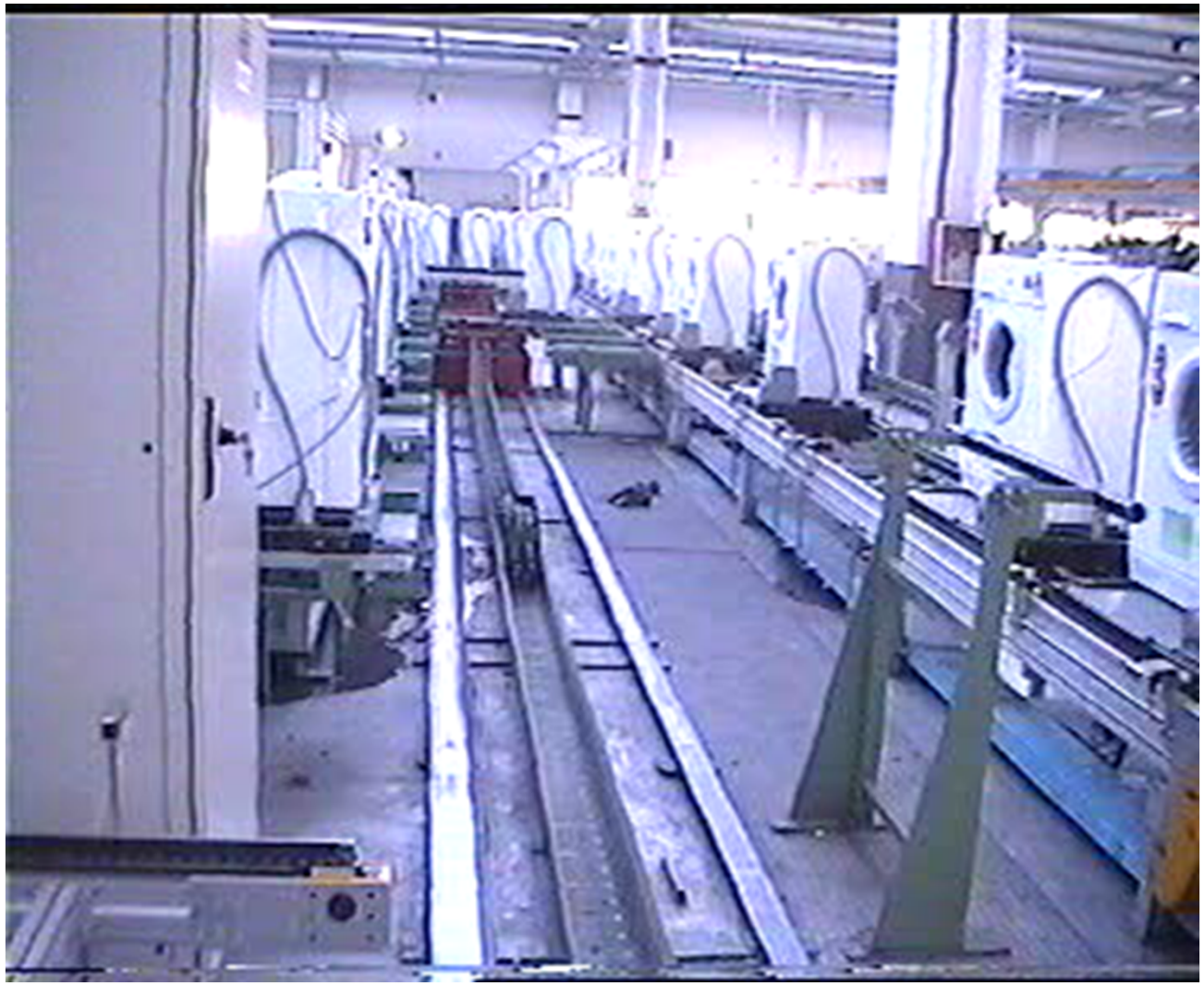
Technical University of Crete, Greece



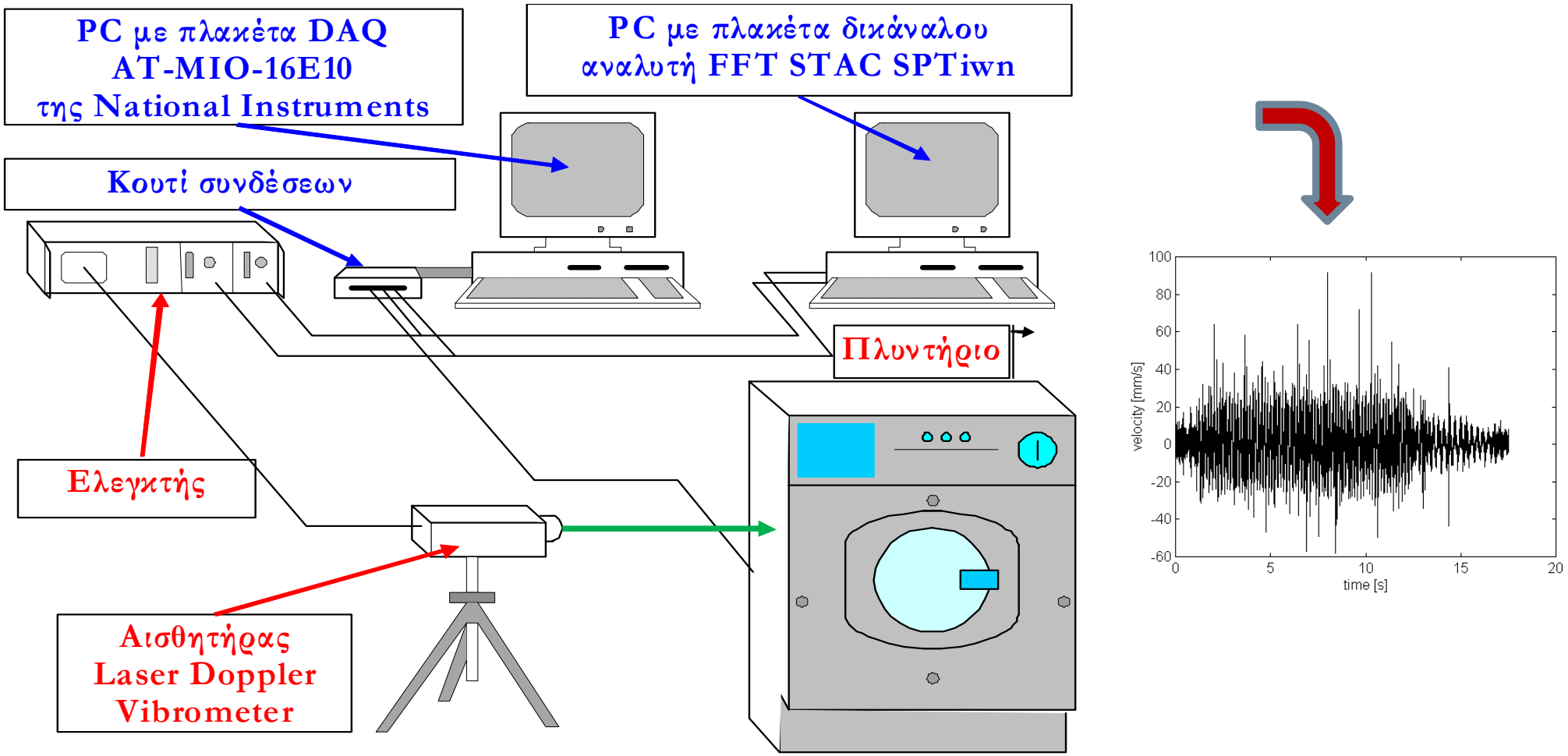
Ariston Dialogic (1000 rpm)

Παραδείγματα συνόλων δεδομένων

Στόχος: διάγνωση ελαττωματικών πλυντηρίων με τη χρήση δεδομένων ταλάντωσης σε διάφορα σημεία.

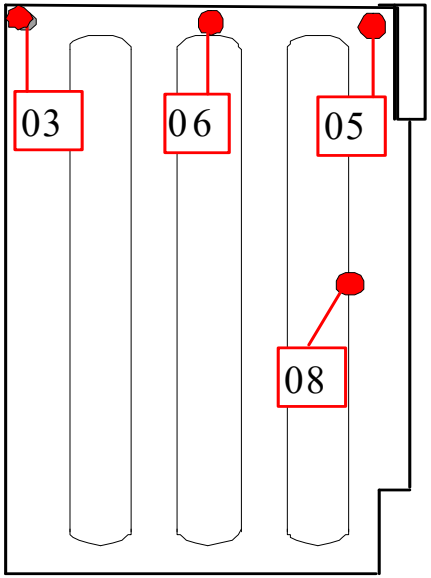


Μετρητική διάταξη

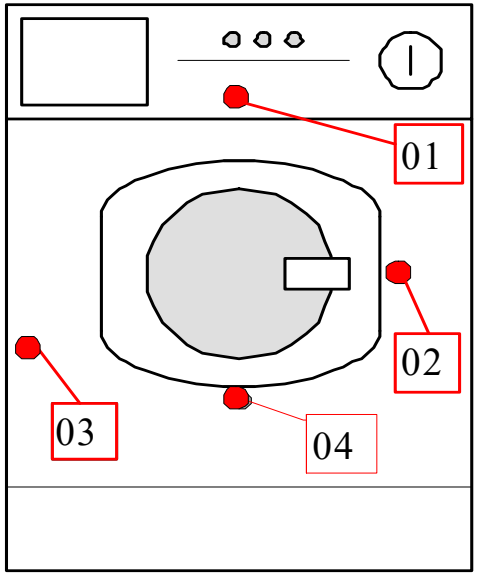


Σημεία μέτρησης (11)

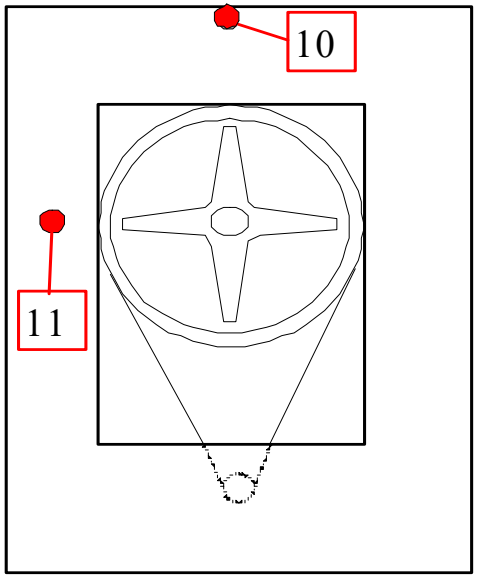
Αριστερά



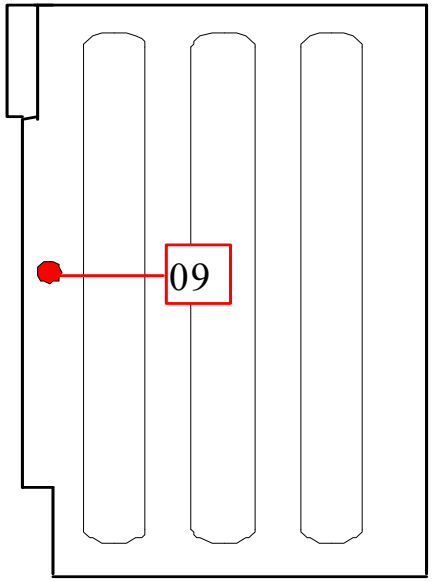
Μπροστά



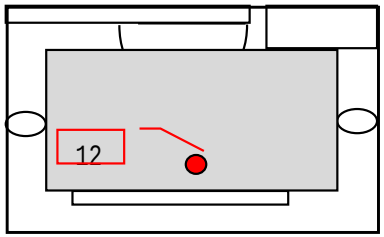
Πίσω



Δεξιά



πάνω



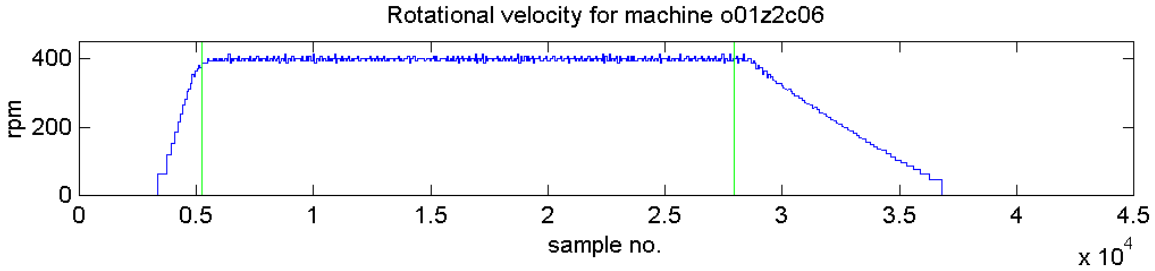
Κλάσεις (3)

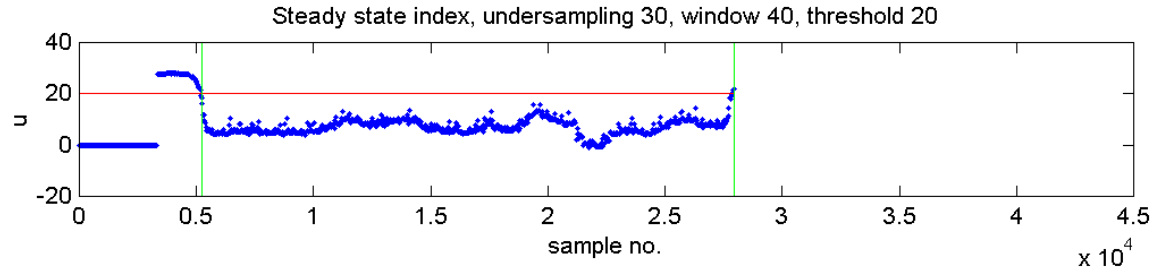
Κλάση	κωδικός	πλήθος
Χωρίς βλάβη	**z**	106
Βλάβη στο κινητήρα (ψήκτρα)	**b**	39
Βλάβη σε ρουλεμάν	**d**	70
Βλάβη στη τροχαλία	**g**	
Βλάβη στον αποσβεστήρα	**h**	
Σφάλμα στα ελατήρια	**m**	

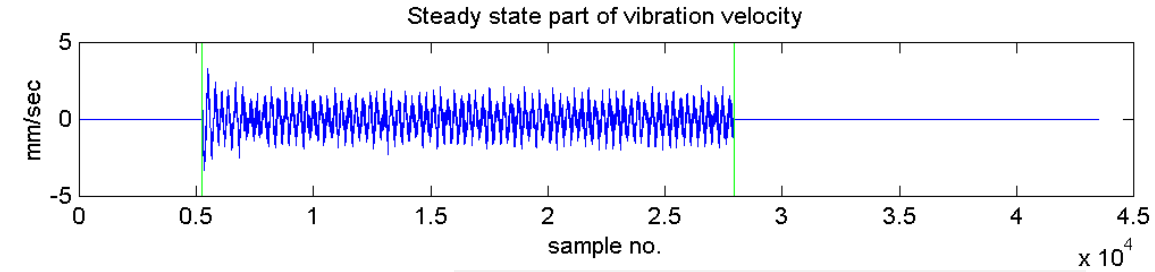
Παραδείγματα συνόλων δεδομένων

Φάση	Επιταχυνσιόμετρο	Χαρακτηριστικό	α/α
Μεταβατική	γραμμικό	Cepstrum Maximum	1
"	"	Envelope Maximum	2
"	"	Wavelet	3
"	γωνιακό	Envelope Maximum	4
"	"	Short Time Frequency Analysis - Window Energy value	5
"	**"**	**Cepstrum maximum**	**6**
Σταθερή	γωνιακό	RMS	7
"	Γραμμικό	RMS	8
"	**Longitudinal**	**Cepstrum maximum**	**9**

Παραδείγματα συνόλων δεδομένων

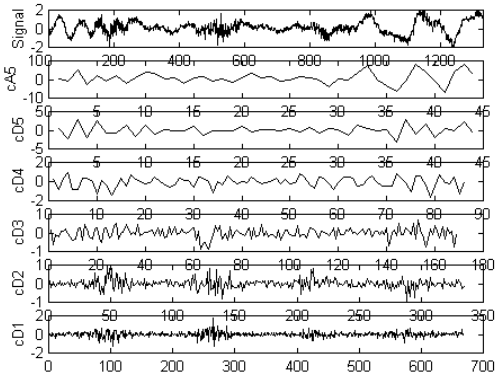




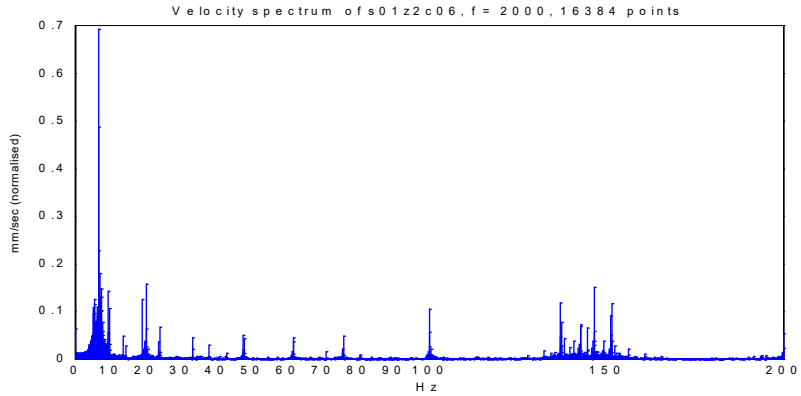


Χρήση περιστροφικής ταχύτητας για το καθορισμό περιοχής σταθερής κατάστασης

Wavelet του μεταβατικού τμήματος

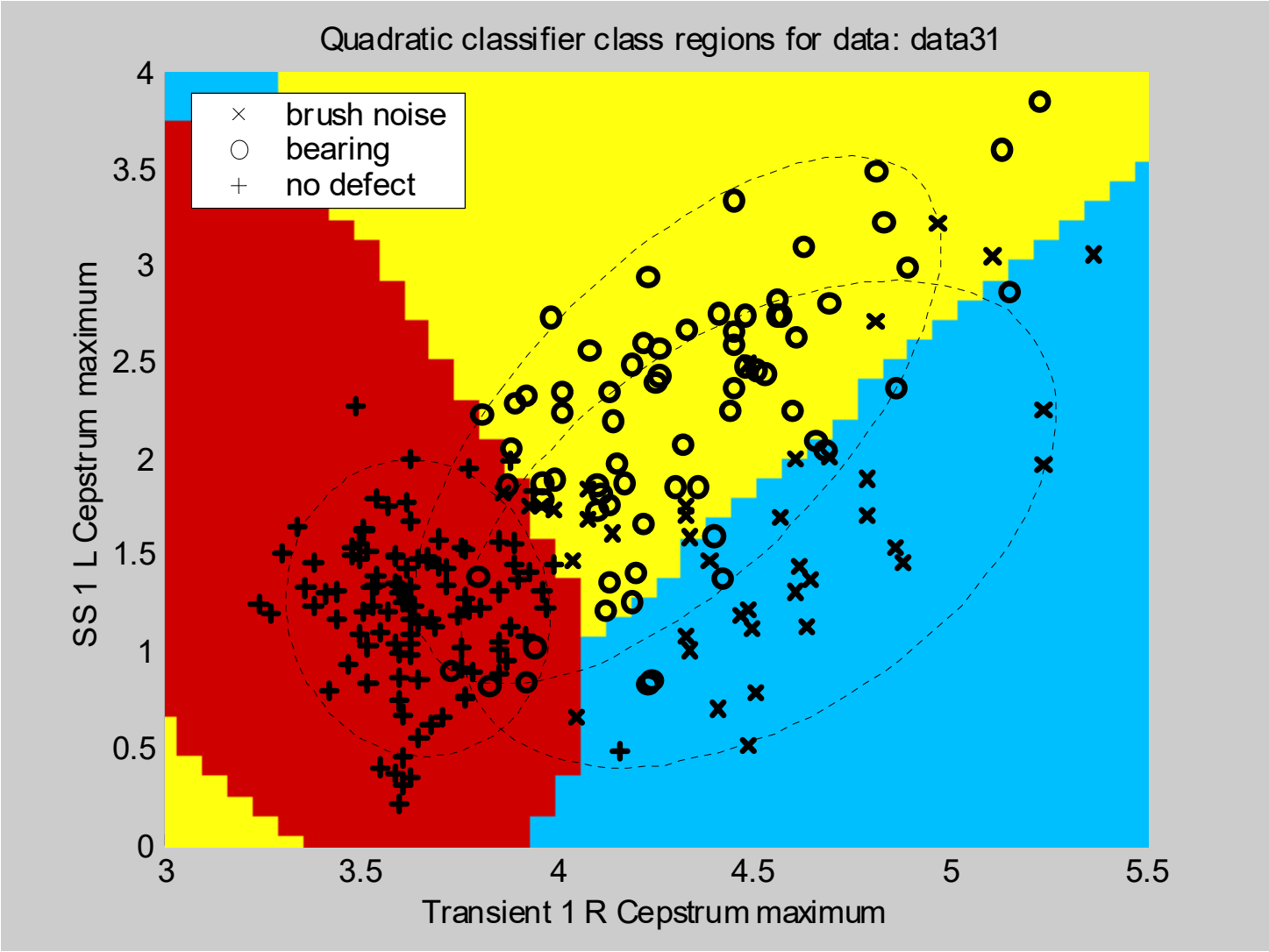


FFT του τμήματος σταθερής κατάστασης



Δημιουργία χαρακτηριστικών

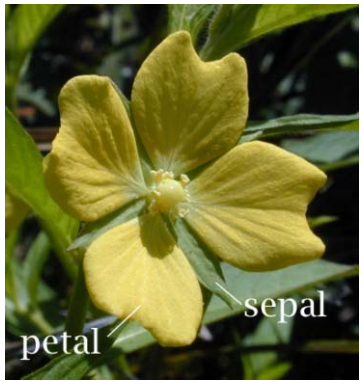
Παραδείγματα συνόλων δεδομένων



Παραδείγματα συνόλων δεδομένων

IRIS

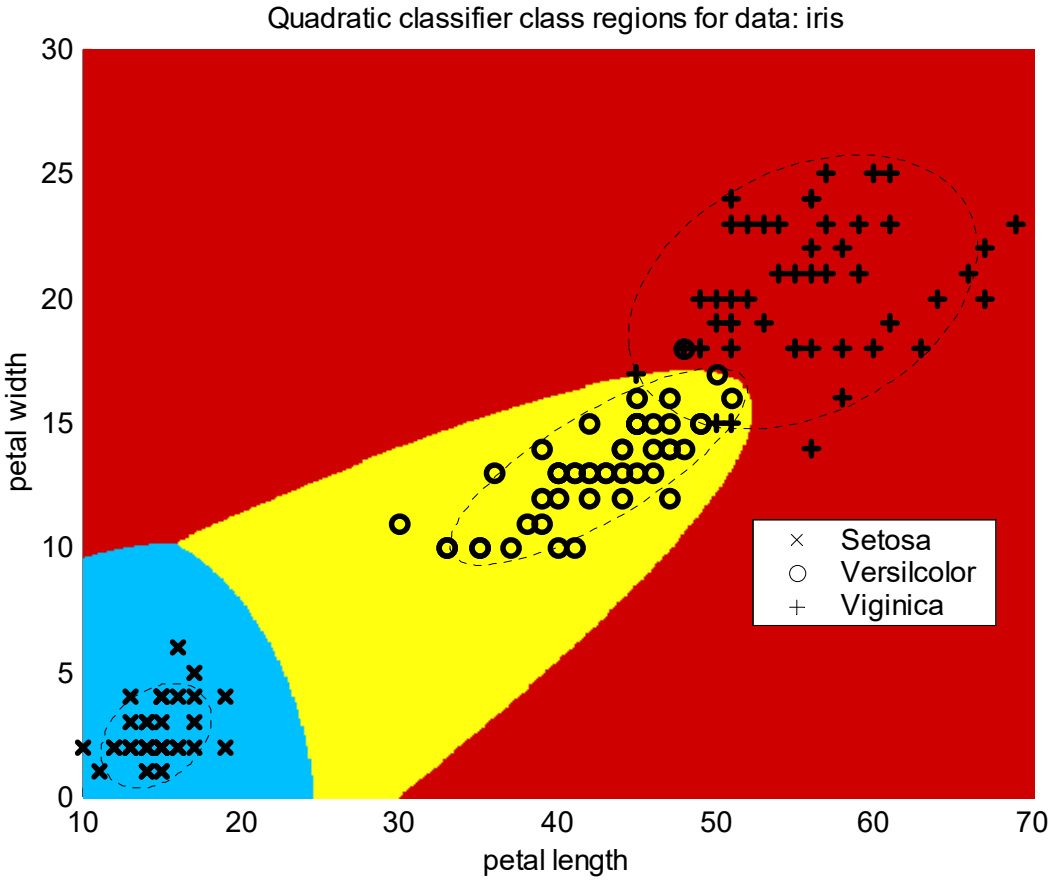
150 δεδομένα από τρία διαφορετικά είδη (κλάσεις): **Setosa**, **Versicolor** και **Virginica**.
Για κάθε είδος 4 χαρακτηριστικά: μήκος σεπάλων, πλάτος σεπάλων, πλάτος πετάλων, μήκος πετάλων



versicolor

setosa

virginica



http://en.wikipedia.org/wiki/Iris_flower_data_set