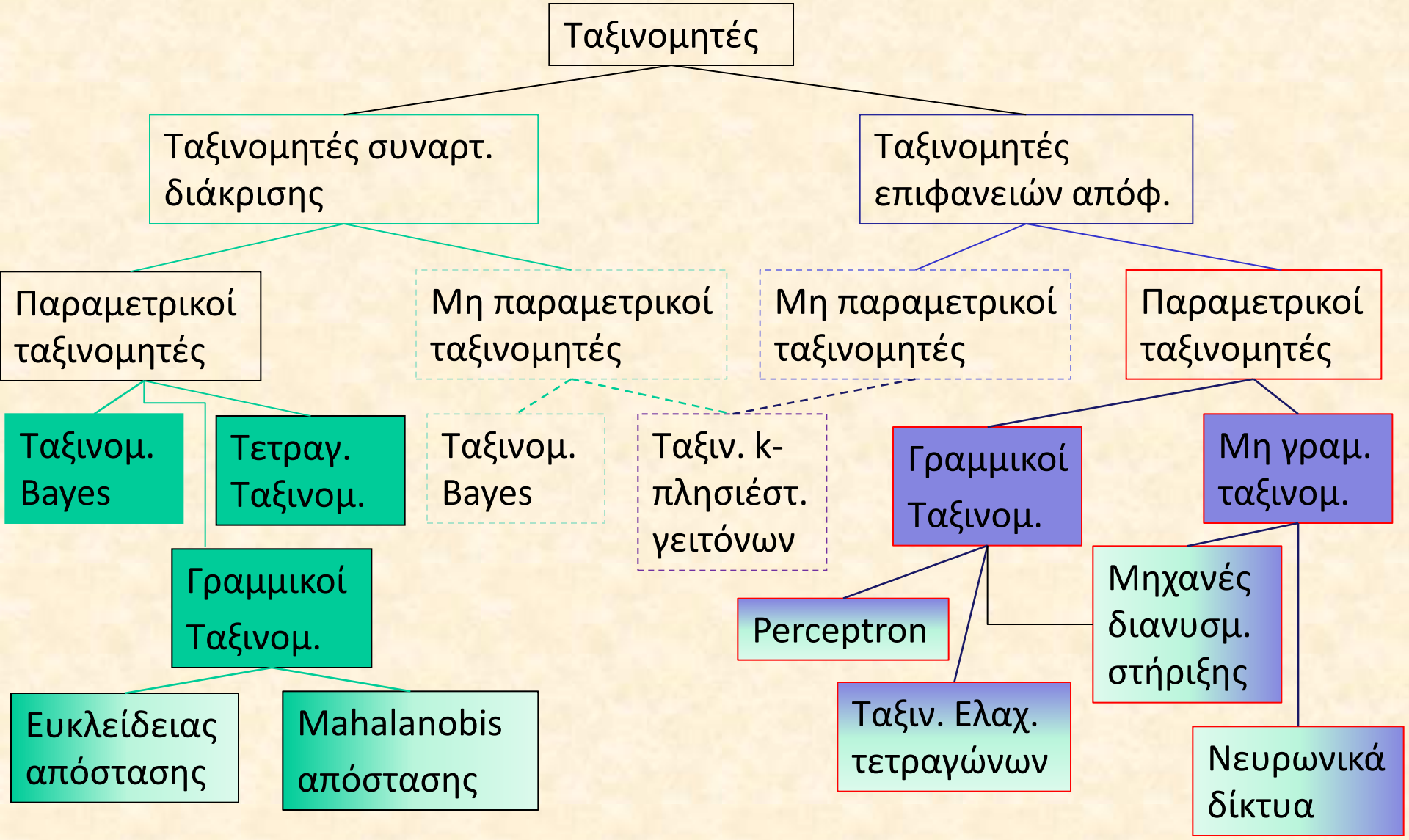
❖ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ
❖(PATTERN RECOGNITION)

Σέργιος Θεοδωρίδης

Κωνσταντίνος Κουτρούμπας

“ΧΑΡΤΟΓΡΑΦΗΣΗ” ΤΟΥ ΧΩΡΟΥ ΤΩΝ ΤΑΞΙΝΟΜΗΤΩΝ



"ΧΑΡΤΟΓΡΑΦΗΣΗ" ΤΟΥ ΧΩΡΟΥ ΤΩΝ ΤΑΞΙΝΟΜΗΤΩΝ

Υπενθ.: X είναι το σύνολο των δεδομένων σημείων όλων των κλάσεων

X_j είναι το υποσύνολο του X που περιέχει τα διανύσματα της κλάσης ω_j,

$X=X_1 \cup \ldots \cup X_M$

Ταξινομητές με βάση τις συναρτήσεις διάκρισης

Ταξινομ. Bayes

- $g_j(x) = f(P(\omega_j)p(x|\omega_j))$
- Εκτιμ. $p(x|\omega_j) \approx \hat{p}(x|\omega_j;\vartheta_j)$
- Εκτιμ. ϑ_j, με βάση το X_j

$$(ML,\ EM): X_j \to \hat{\vartheta}_j$$

- $x_i \to p(x_i|\omega_j) \approx \hat{p}(x_i|\omega_j;\hat{\vartheta}_j)$

Τετραγωνικός ταξινομητής

- $g_j(x) = (x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)$
- Υπόθεση : $N(\mu_j, \Sigma_j)$
- Εκτιμ. μ_j, Σ_j, με βάση το X_j

$$(ML): X_j \to \hat{\mu}_j, \hat{\Sigma}_j$$

- $x_i \to g_j(x_i) = (x_i-\hat{\mu}_j)^T \hat{\Sigma}_j^{-1}(x_i-\hat{\mu}_j)$

Γραμμικός Ευκλείδειος ταξινομητής

- $g_j(x) = (x-\mu_j)^T(x-\mu_j)$
- Υπόθεση : $N(\mu_j, I)$
- Εκτίμ. μ_j, με βάση το X_j

$$(ML): X_j \to \hat{\mu}_j$$

- $x_i \to g_j(x_i) = (x_i-\hat{\mu}_j)^T(x_i-\hat{\mu}_j)$

Γραμμικός Mahalanobis ταξινομητής

- $g_j(x) = (x-\mu_j)^T \Sigma^{-1}(x-\mu_j)$
- Υπόθεση : $N(\mu_j, \Sigma)$
- Εκτίμ. μ_j, Σ, με βάση το X_j

$$(ML): X_j \to \hat{\mu}_j, \hat{\Sigma}$$

- $x_i \to g_j(x_i) = (x_i-\hat{\mu}_j)^T \hat{\Sigma}^{-1}(x_i-\hat{\mu}_j)$

"ΧΑΡΤΟΓΡΑΦΗΣΗ" ΤΟΥ ΧΩΡΟΥ ΤΩΝ ΤΑΞΙΝΟΜΗΤΩΝ

Υπενθ.: X είναι το σύνολο των δεδομένων σημείων όλων των κλάσεων

X_j είναι το υποσύνολο του X που περιέχει τα διανύσματα της κλάσης ω_j,

$X=X_1 \cup \ldots \cup X_M$

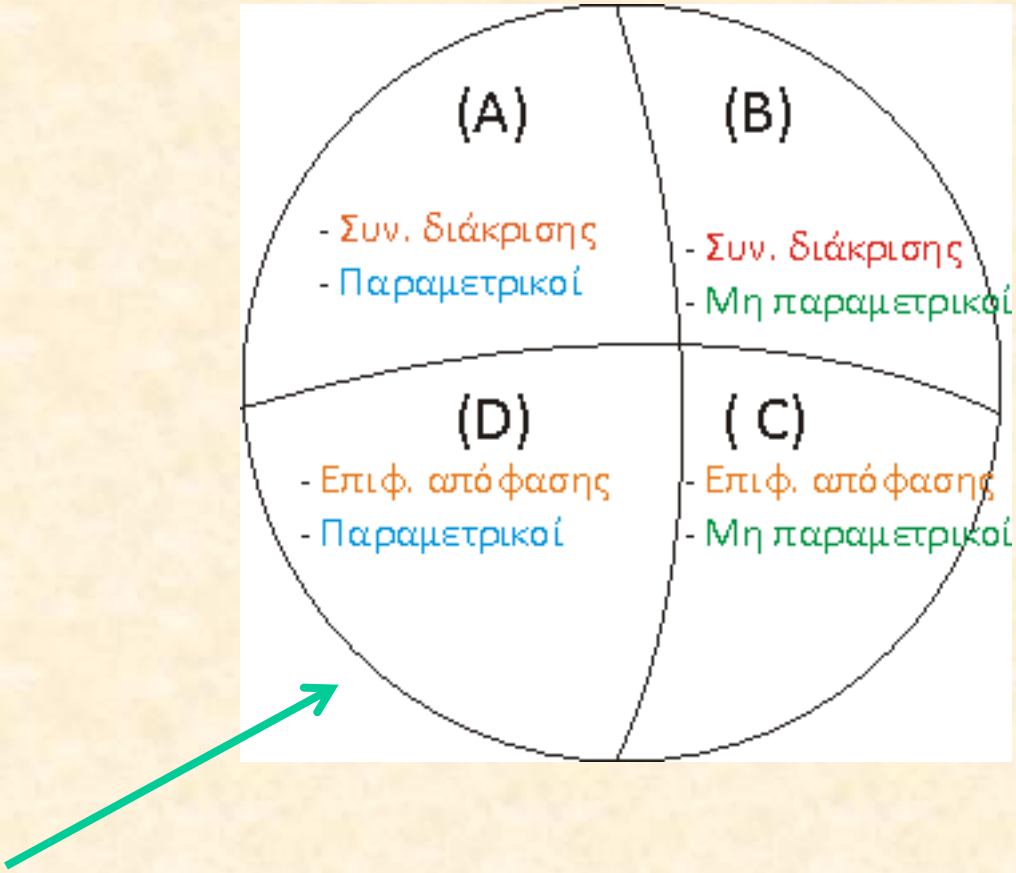
Μη παραμετρικοί ταξινομητές με βάση τις συναρτήσεις διάκρισης

Ταξινομητής Bayes

- $g_j(x) = f(P(\omega_j)p(x|\omega_j))$
- $x_i \rightarrow p(x_i|\omega_j) \approx \hat{p}(x_i|\omega_j; X_j)$
  $(παράθυρα\ Parzen,$
  $εκτίμ.\ πυκν.\ βάσει\ των\ k-πλησ.\ γειτ.$

Ταξινομητής k-πλησιέστερων γειτόνων

- $x_i \rightarrow g_j(x_i) = k_i^j$


(A)
- Συν. διάκρισης
- Παραμετρικοί
(B)
- Συν. διάκρισης
- Μη παραμετρικοί
(D)
- Επιφ. απόφασης
- Παραμετρικοί
(C)
- Επιφ. απόφασης
- Μη παραμετρικοί

ΜΗ ΓΡΑΜΜΙΚΟΙ ΠΑΡΑΜΕΤΡΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ ΕΠΙΦΑΝΕΙΩΝ ΑΠΟΦΑΣΗΣ

Μερικά προκαταρτικά:

- Στην πράξη έχουμε στη διάθεσή μας ένα σύνολο δεδομένων (training set)

όπου $$X = \{(x_i, d_i), x_i \in R^l, d_i \in \{1,2,...,M\}, i = 1,...N\}$$

$\boldsymbol{x}_i$ είναι η *l*-διάστατη αναπαράσταση της *i*-στής οντότητας ενός συνόλου *N* οντοτήτων (training vector)

d_i είναι η ετικέτα της κλάσης στην οποία ανήκει το $\boldsymbol{x}_i$ (1 για ω_1, 2 για ω_2,…).

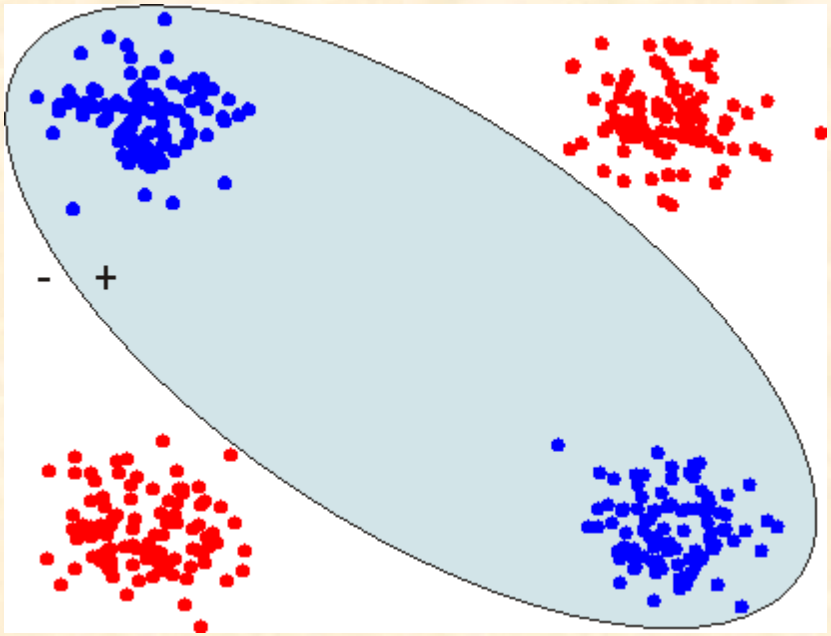
- Εστιάζουμε κυρίως στην περίπτωση δύο κλάσεων (συνήθως $\omega_1 \rightarrow +1$ ή A **και** $\omega_2 \rightarrow -1(0)$ ή B).

- Δεν υιοθετούμε κάποια υπόθεση σχετικά με τις συναρτήσεις πυκν. πιθ. (pdfs) που μοντελοποιούν τις κλάσεις.

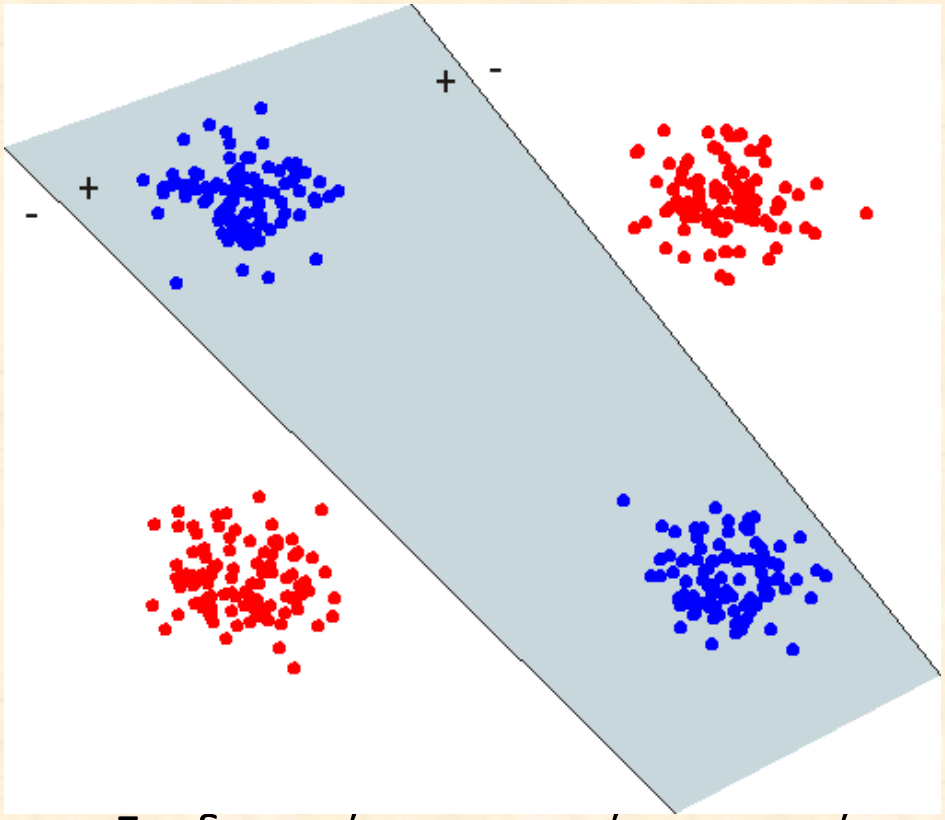
- Εστιάζουμε σε ταξινομητές που δύνανται να υλοποιήσουν μη γραμμικούς διαχωρισμούς μεταξύ των (δεδομένων των) κλάσεων.

-***ΣΗΜ.:*** Μη γραμμικοί διαχωρισμοί μπορούν να επιτευχθούν είτε μέσω μιας μη γραμμικής επιφανειας, είτε μέσω συνδυασμού μερικών γραμμικών η μη γραμμικών επιφανειών. Σε κάθε περίπτωση ορίζεται μια **επιφάνεια απόφασης**.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΠΑΡΑΜΕΤΡΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ ΕΠΙΦΑΝΕΙΩΝ ΑΠΟΦΑΣΗΣ



Μία μη γραμμική επιφάνεια



Συνδυασμός περισσοτέρων της μίας επιφανειών

ΜΗ ΓΡΑΜΜΙΚΟΙ ΠΑΡΑΜΕΤΡΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ ΕΠΙΦΑΝΕΙΩΝ ΑΠΟΦΑΣΗΣ

Σημ.: Εκτός αν ορίζεται διαφορετικά, θεωρούμε την περίπτωση των δύο κλάσεων,

i.e., ω_1 (+1) and ω_2 (-1).

Ορισμός του προβλήματος: Δοθέντος ενός συνόλου δεδομένων *X* σχεδίασε ένα ταξινομητή που επιτυγχάνει τον "βέλτιστο" δυνατό διαχωρισμό των (διανυσμάτων των) δύο κλάσεων.

Στρατηγική επίλυσης:

1. **Υιοθέτησε** ένα συγκεκριμένο (μη γραμμικό) παραμετρικό μοντέλο για τον ταξινομητή (***w*** είναι το διάνυσμα που περιέχει όλες τις παραμέτρους του).
2. **Όρισε** κατάλληλη συνάρτηση κόστους (cost function) του ***w***, $J(\boldsymbol{w})$, η οποία εμπλέκει επίσης τα διανύσματα του *X*, έτσι ώστε **οι θέσεις των βέλτιστών της να αντιστοιχούν στο βέλτιστο δυνατό διαχωρισμό για το πρόβλημα.** **Βελτιστοποίησε την** $J(\boldsymbol{w})$ ως προς ***w***. **Η θέση *w* όπου η** $J(\boldsymbol{w})$ **παρουσιάζει βέλτιστο ορίζει τον καλύτερο δυνατό διαχωρισμό**.

Σημαντική παρατήρηση: Η έννοια της φράσης "καλύτερος δυνατός διαχωρισμός" διαφέρει για διαφορετικές επιλογές της $J(\boldsymbol{w})$.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΠΑΡΑΜΕΤΡΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ ΕΠΙΦΑΝΕΙΩΝ ΑΠΟΦΑΣΗΣ

Σύντομη υπενθύμιση:

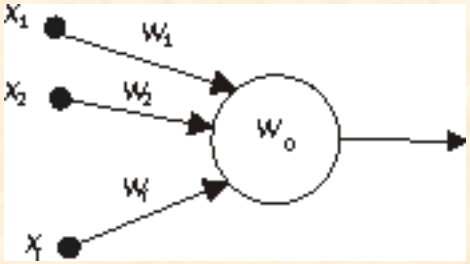
- Ένας γραμμικός ταξινομητής ορίζεται **μέσω** ενός υπερεπιπέδου

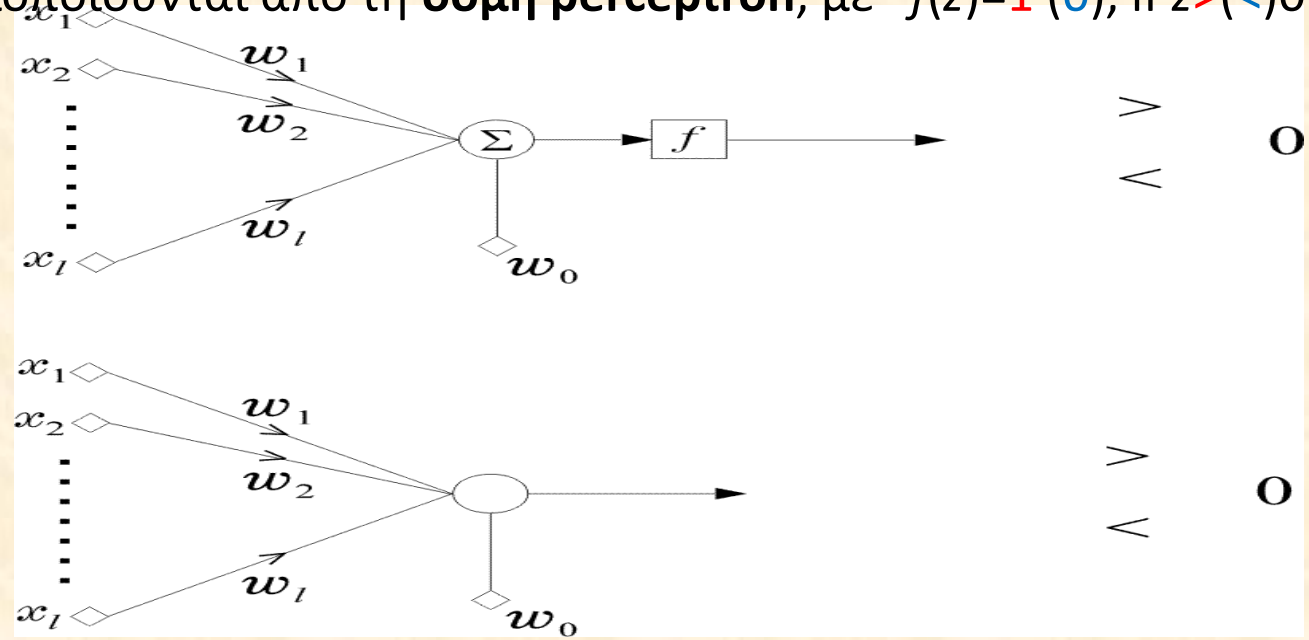
$$(H): h(x,w) = w_1 x_1 + \ldots + w_l x_l + w_0 = \sum_{k=1}^{l} w_k x_k + w_0 = w^T x + w_0 = 0$$

όπου $\boldsymbol{w}=[w_1,\ldots,w_l]^T$, $\boldsymbol{x}=[x_1,\ldots,x_l]^T$ (η γνώση των $\boldsymbol{w}$ και w_0 ορίζει πλήρως το (H))

- Αν για δεδομένο $\boldsymbol{x}$ είναι $h(\boldsymbol{x},\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x}+w_0 > (<) 0$, το $\boldsymbol{x}$ καταχωρείται στην κλάση +1(0).

-Τέτοιοι ταξινομητές υλοποιούνται από τη **δομή perceptron**, με $f(z)=1$ (0), if $z>(<)0$

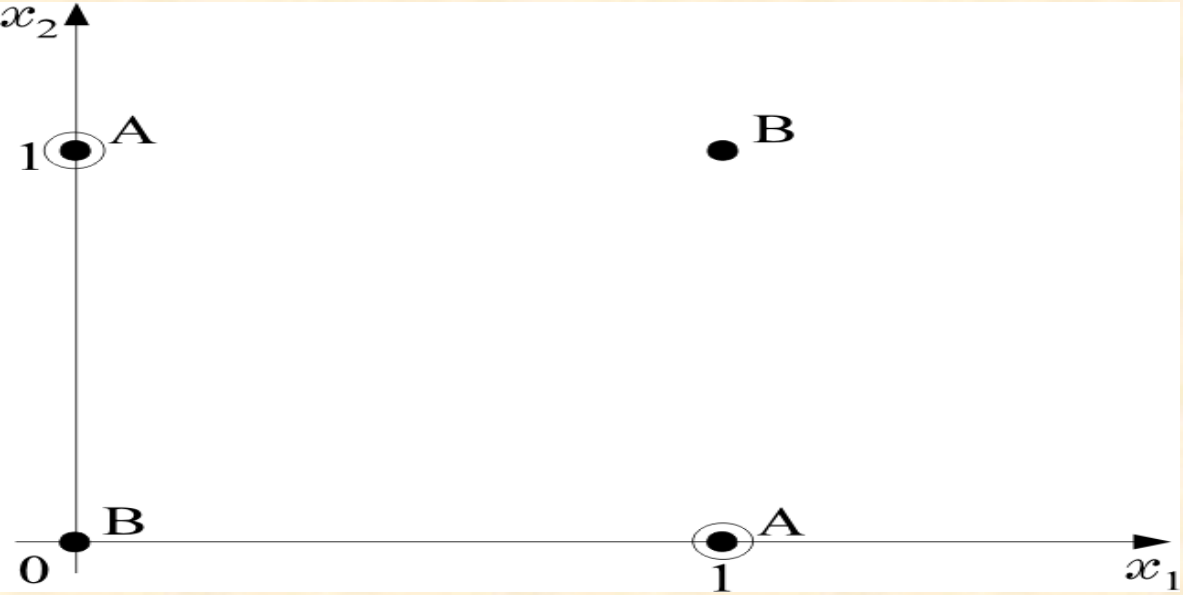




ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Το πρόβλημα exclusive OR (*XOR*)

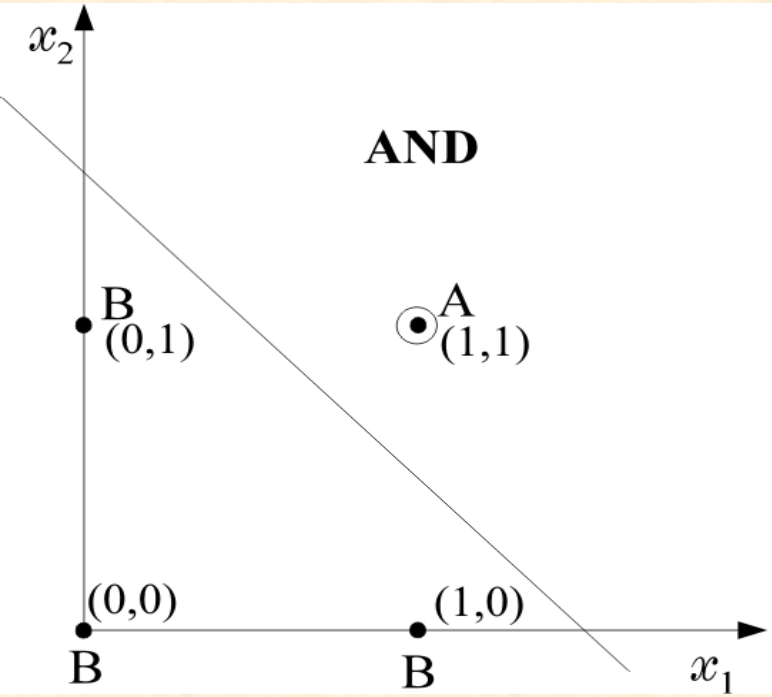
x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B

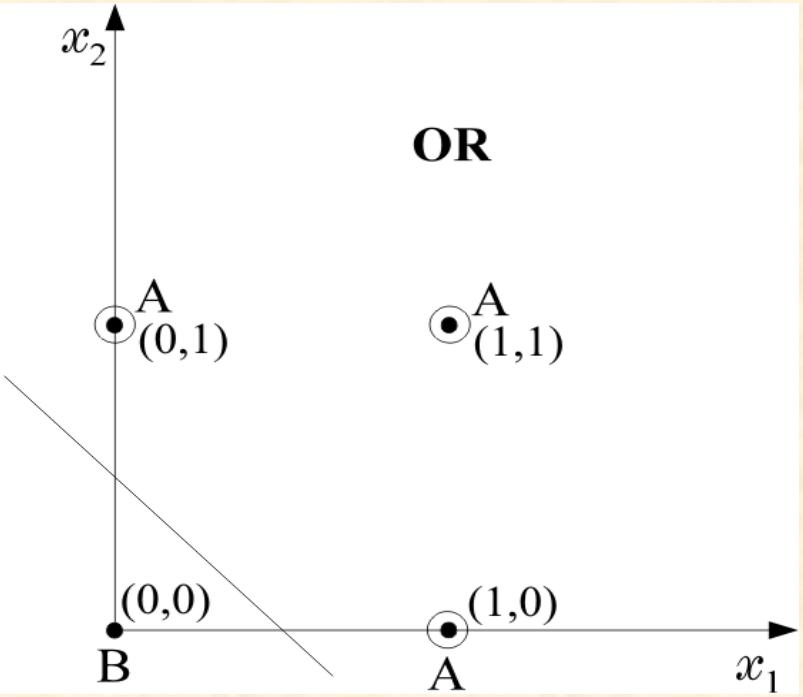


Δεν υπάρχει καμία γραμμή (υπερεπίπεδο) που διαχωρίζει την κλάση A από την κλάση B.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Αντίθετα, τα προβλήματα *AND* και *OR* είναι γραμμικώς διαχωρίσιμα.

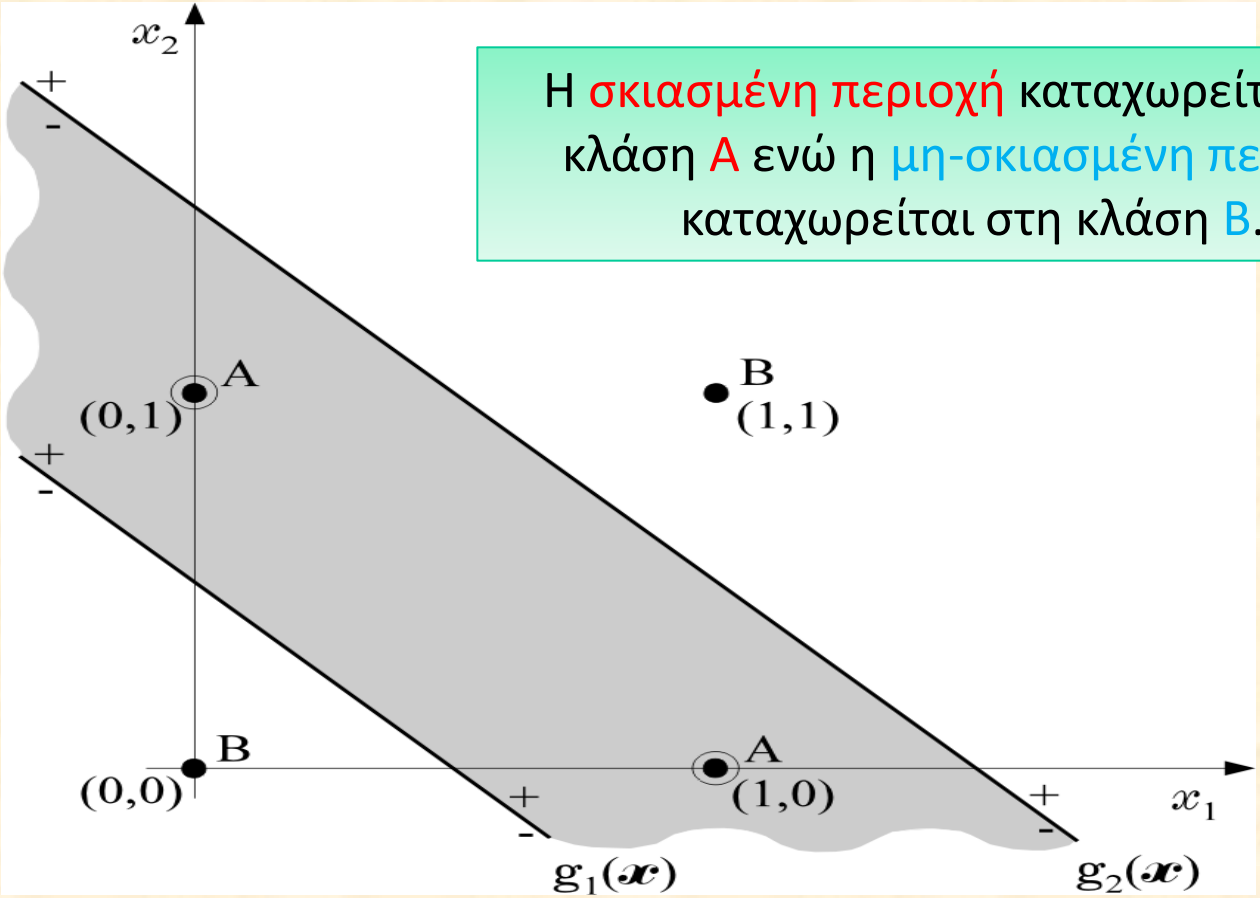




ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Η δομή δικτύου Perceptron δύο επιπέδων

Για το πρόβλημα *XOR*, σχεδίασε δύο, αντί για μία γραμμές.



Η σκιασμένη περιοχή καταχωρείται στη κλάση Α ενώ η μη-σκιασμένη περιοχή καταχωρείται στη κλάση Β.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Η δομή δικτύου Perceptron δύο επιπέδων

Έστω ότι το $\boldsymbol{x}=[x_1,x_2]^T$ απεικονίζεται στο διάνυσμα $\boldsymbol{y}=[y_1,y_2]^T$, όπου το y_i ισούται με 1(0), αν το $\boldsymbol{x}$ ανήκει στη θετική (αρνητική) πλευρά της i-στής γραμμής, i=1,2. Τότε

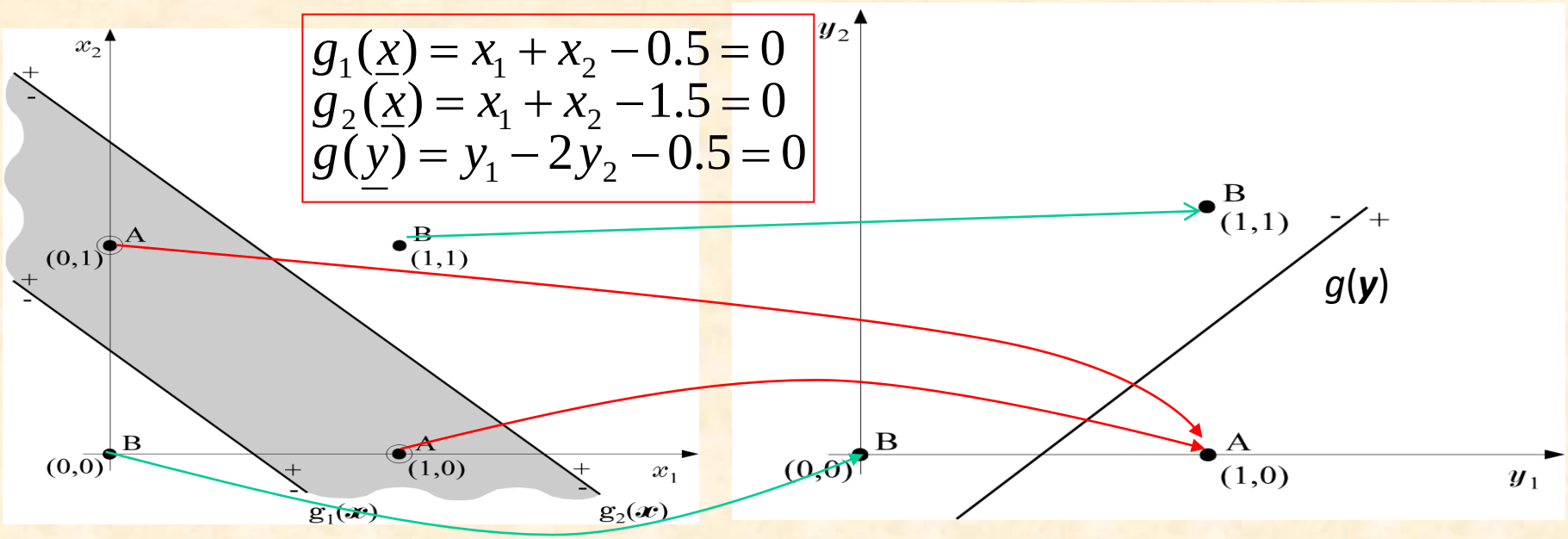
Απεικόνιση				Διαχωρισμός
x_1	x_2	y_1	y_2	
0	0	0(-)	0(-)	B(0)
0	1	1(+)	0(-)	A(1)
1	0	1(+)	0(-)	A(1)
1	1	1(+)	1(+)	B(0)

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Η δομή δικτύου Perceptron δύο επιπέδων

Έστω ότι το $\boldsymbol{x}=[x_1,x_2]^T$ απεικονίζεται στο διάνυσμα $\boldsymbol{y}=[y_1,y_2]^T$, όπου το y_i ισούται με 1(0), αν το $\boldsymbol{x}$ ανήκει στη θετική (αρνητική) πλευρά της i-στής γραμμής, i=1,2. Τότε,

- κάθε σημείο της σκιασμένης περιοχής απεικονίζεται στο σημείο (1,0) στο νέο χώρο
- κάθε σημείο της μη σκιασμένης περιοχής απεικονίζεται είτε στο (0,0) είτε στο (1,1).



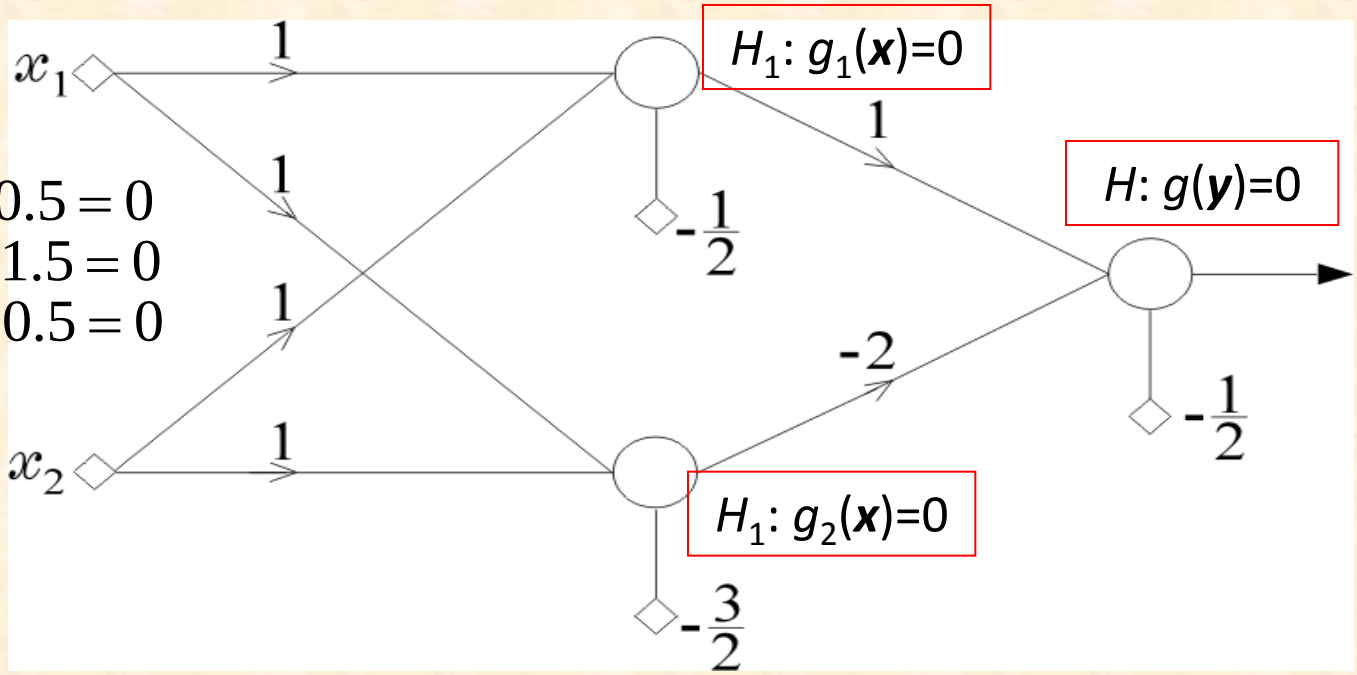
ΣΗΜ.: Το πρόβλημα γίνεται γραμμικώς διαχωρίσιμο στο μετασχηματισμένο χώρο.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Η δομή δικτύου Perceptron δύο επιπέδων - Υλοποίηση

- Τοποθέτησε δύο νευρώνες (perceptrons) στο ίδιο (**πρώτο**) **επίπεδο**. Ο πρώτος (δεύτερος) υλοποιεί το διαχωρισμό που ορίζεται από τη γραμμή $g_1(\boldsymbol{x})=0$ ($g_2(\boldsymbol{x})=0$).
- Τοποθέτησε έναν επιπλέον νευρώνα στο **δεύτερο επιπεδο**, που παίρνει σαν είσοδο τις εξόδους των προηγουμένων δύο νευρώνων και υλοποιεί την ταξινόμηση στο μετασχηματισμένο χώρο.

$$H_1 : g_1(\underline{x}) = x_1 + x_2 - 0.5 = 0$$
$$H_2 : g_2(\underline{x}) = x_1 + x_2 - 1.5 = 0$$
$$H : g(y) = y_1 - 2y_2 - 0.5 = 0$$

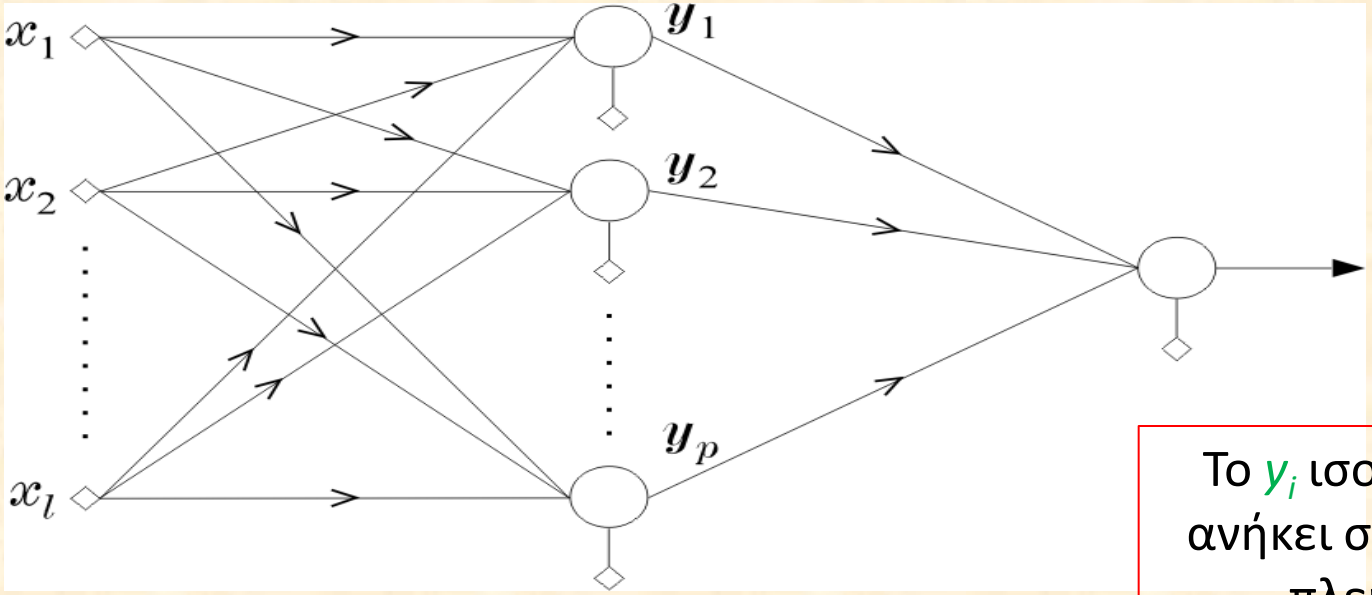


ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δυνατότητες ταξινόμησης των δικτύων perceptrons δύο επιπέδων

- Στην περίπτωση το προβλήματος *XOR*, οι νευρώνες του πρώτου επιπέδου απεικονίζουν τα διανύσματα εισόδου (***x***) στις κορυφές του κύβου με πλευρά ίση με 1, δηλ., στις (0, 0), (0, 1), (1, 0), (1, 1).
- Στη γενικότερη περίπτωση όπου $\boldsymbol{x} \in R^l$ και χρησιμοποιούνται p (πρώτου επιπέδου) νευρώνες, η απεικόνιση γίνεται στις κορυφές του H_p υπερκύβου, δηλ.,

$$x \rightarrow y = [y_1, \ldots y_p]^T,\ y_i \in \{0,1\}\ \ i = 1, 2, \ldots p$$



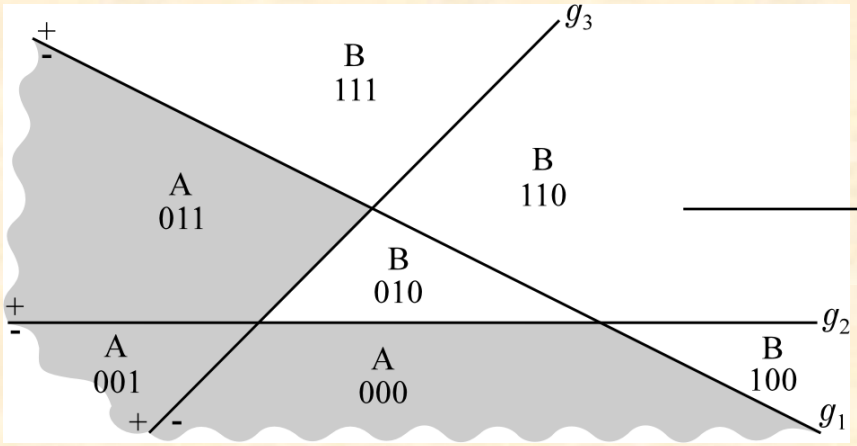
Το y_i ισούται με 1(0) αν το ***x*** ανήκει στη θετική (αρνητική) πλευρά του $g_i(\boldsymbol{x})=0$.

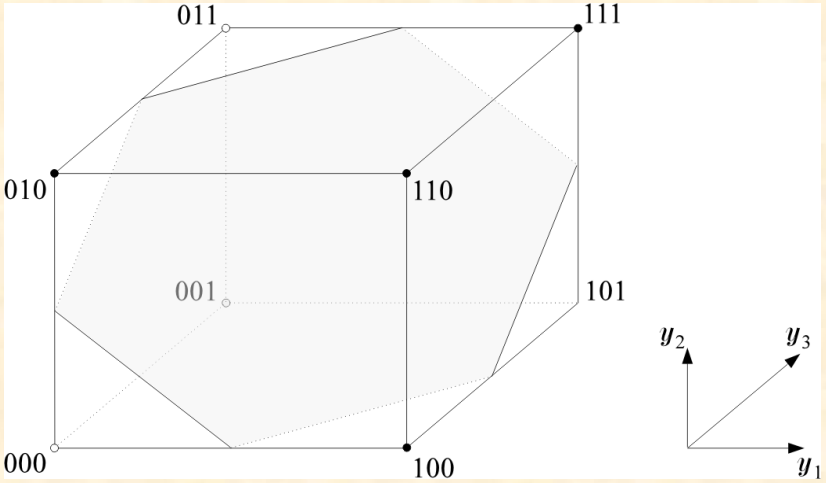
ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δυνατότητες ταξινόμησης των δικτύων perceptrons δύο επιπέδων

- Τεμνόμενα υπερεπίπεδα στο χώρο που αντιστοιχούν στους p νευρώνες του πρώτου επιπέδου, ορίζουν περιοχές με την ακόλουθη ιδιότητα:

  "ΟΛΑ τα σημεία μιας περιοχής έχουν την ίδια σχετική θέση ως προς τα p υπερεπίπεδα. Επομένως, ***απεικονίζονται στην ίδια κορυφή του*** H_p ***υπερκύβου***."





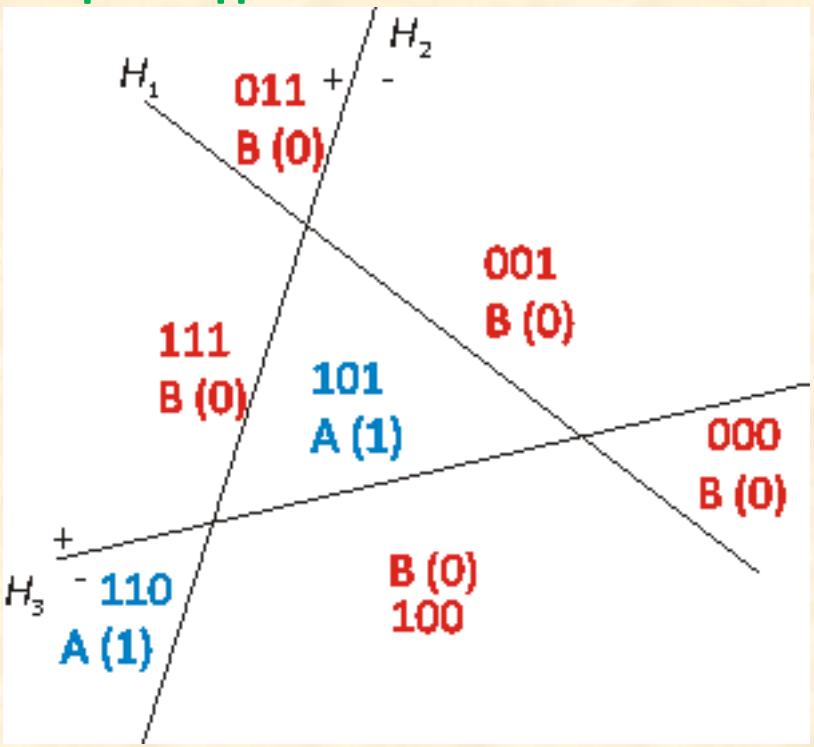
Παράδειγμα: Η κορυφή 001 αντιστοιχεί στην περιοχή που βρίσκεται στην (-) πλευρά της $g_1(\underline{x})=0$, στην (-) πλευρά της $g_2(\underline{x})=0$, στη (+) πλευρά της $g_3(\underline{x})=0$.

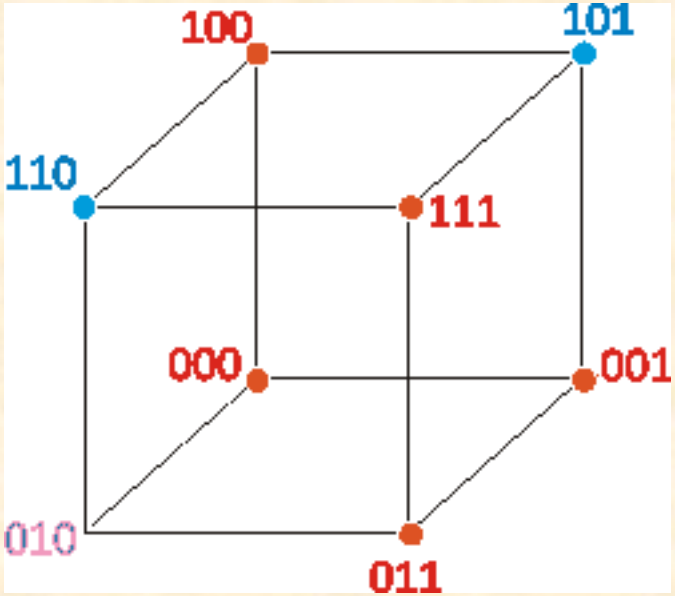
ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δυνατότητες ταξινόμησης των δικτύων perceptrons δύο επιπέδων

- Ο νευρώνας εξόδου αντιστοιχεί σε ένα υπερεπίπεδο στο μετασχηματισμένο χώρο το οποίο χωρίζει κάποιες κορυφές του H_p υπερκύβου από τις υπόλοιπες. Συνεπώς, ένα perceptron δύο επιπέδων έχει τη δυνατότητα να διαχωρίσει **κλάσεις που αποτελούνται από ενώσεις πολυεδρικών περιοχών**.
- Αλλά **ΌΧΙ ΟΠΟΙΕΣΔΗΠΟΤΕ** ενώσεις. Αυτό εξαρτάται από τη σχετική θέση των αντίστοιχων κορυφών του υπερκύβου.

Παράδειγμα:



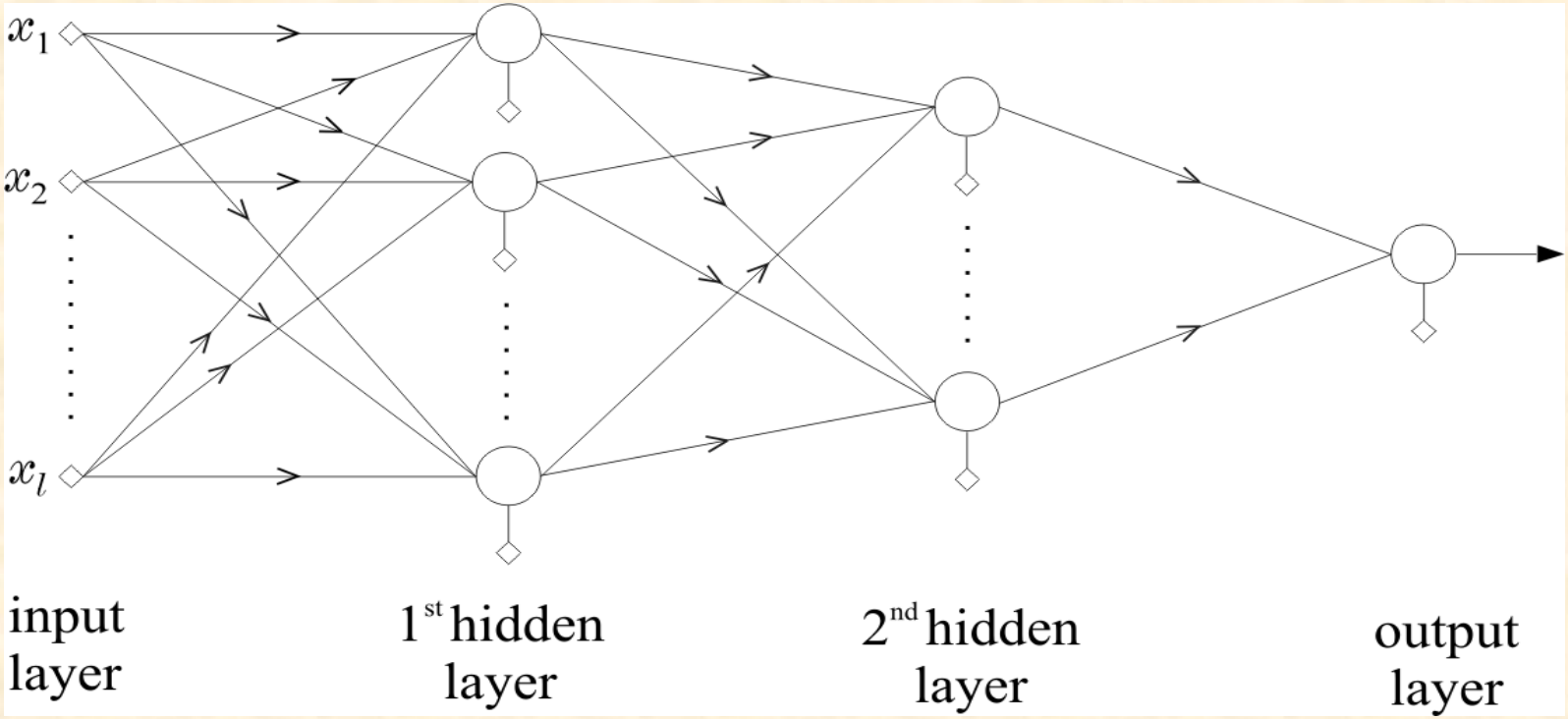


Δεν υπάρχει υπερεπίπεδο που να διαχωρίζει τις **μπλε** κορυφές από τις υπόλοιπες

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δίκτυα Perceptrons τριών επιπέδων

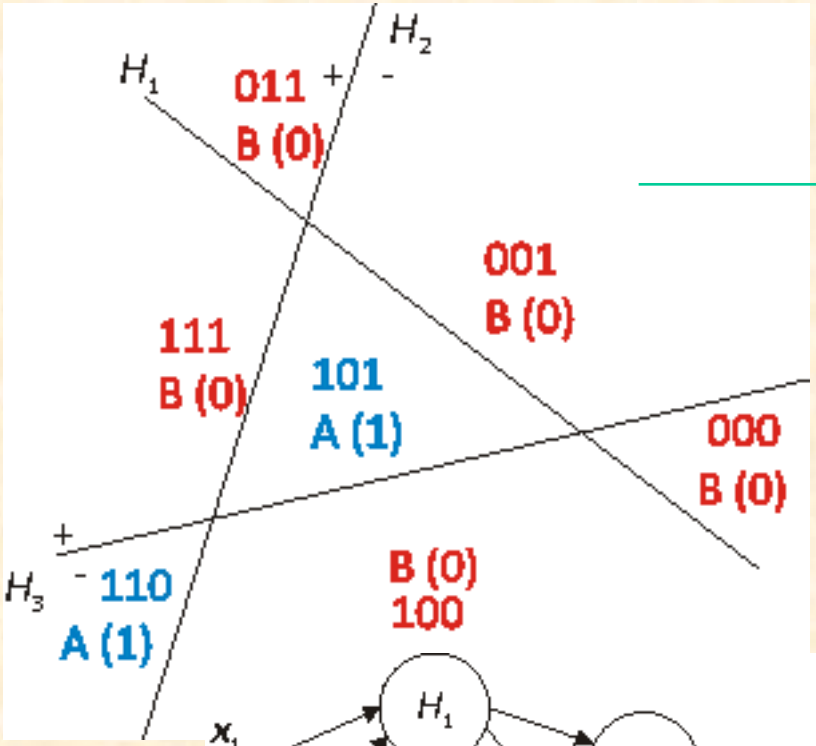
Η αρχιτεκτονική

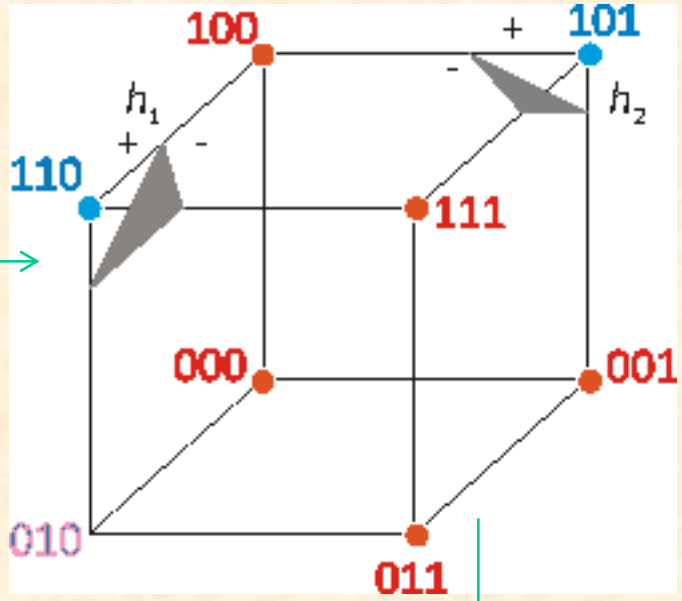


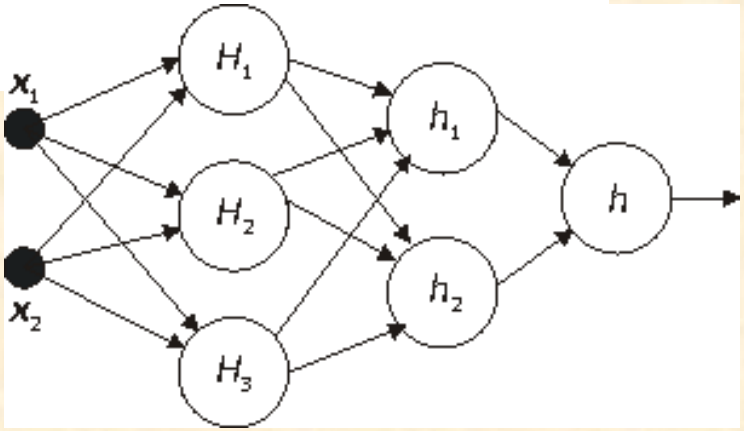
Τέτοιες δομές είναι ικανές να διαχωρίσουν κλάσεις που ορίζονται από **οποιαδήποτε** ένωση πολυεδρικών περιοχών.

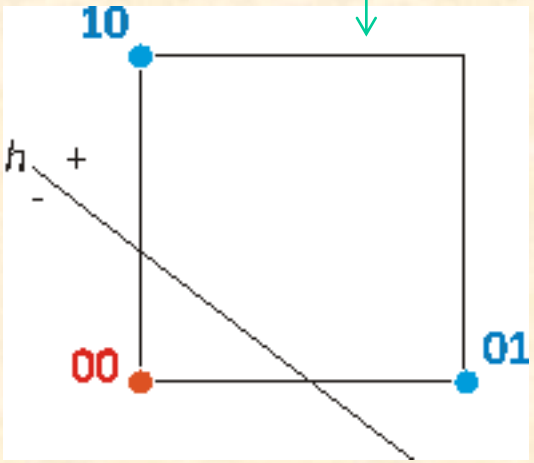
ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δίκτυα Perceptrons τριών επιπέδων– παράδειγμα









ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Δίκτυα Perceptrons τριών επιπέδων

Έστω $A=R_1 \cup \ldots \cup R_k$ (1) και $B=R_1' \cup \ldots \cup R_m'$ (0) ($A \cup B = R^l$), όπου οι περιοχές R_i και R_j' ορίζονται από την τομή p υπερεπιπέδων, $H_1,\ldots,H_p$.

Πώς ένα perceptron τριών επιπέδων μπορεί να υλοποιήσει ΟΠΟΙΟΝΔΗΠΟΤΕ διαχωρισμό που ορίζεται από υπερεπίπεδα:

- Τοποθέτησε p νευρώνες στο πρώτο επίπεδο, καθένας από τους οποίους αντιστοιχεί σε ένα από τα υπερεπίπεδα $H_1,\ldots,H_p$.
- Τοποθέτησε k nodes στο δεύτερο επίπεδο, καθένας από τους οποίους διαχωρίζει μια *κορυφή του* H_p *υπερκύβου* που αντιστοιχεί σε μια περιοχή R_i, από όλες τις υπόλοιπες κορυφές.
- Τοποθέτησε έναν *OR* νευρώνα στο τρίτο επίπεδο (αυτός επιστρέφει **0** για την κορυφή 00…0 του H_k υπερκύβου και **1** για όλες τις υπόλοιπες).

ΣΗΜ.: Το πρώτο επίπεδο του δικτύου ορίζει τα υπερεπίπεδα, το δεύτερο επίπεδο ορίζει τις περιοχές και ο νευρώνας εξόδου ορίζει τις κλάσεις.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Η παραπάνω ανάλυση ήταν χρήσιμη προκειμένου να κατανοήσουμε τη λογική των δικτύων perceptrons πολλών επιπέδων καθώς επίσης και τις δυνατότητές τους.

Ωστόσο, **στην πράξη** το μόνο που έχουμε στη διάθεσή μας είναι ένα σύνολο δεδομένων *X* και το πρόβλημά μας είναι το ακόλουθο:

Ορισμός προβλήματος:

1. **Υιοθέτησε/καθόρισε** δίκτυο perceptron πολλαπλών επιπέδων συγκεκριμένης δομής
2. **Εκτίμησε** το διάνυσμα παραμέτρων του ***w*** (το οποίο περιέχει όλες τις παραμέτρους των νευρώνων του δικτύου), ώστε να επιτευχθεί ο "βέλτιστος" δυνατός διαχωρισμός των κλάσεων, με βάση το *X*.

Δύο κύριες κατευθύνσεις:

- **Ανέπτυξε** μια δομή η οποία ταξινομεί **σωστά όλα** τα διανύσματα εκπαίδευσης.
- **Υιοθέτησε** μια δομή και υπολόγισε τις τιμές των παραμέτρων της μέσω της βελτιστοποίησης μιας κατάλληλα επιλεγμένης συνάρτησης κόστους.

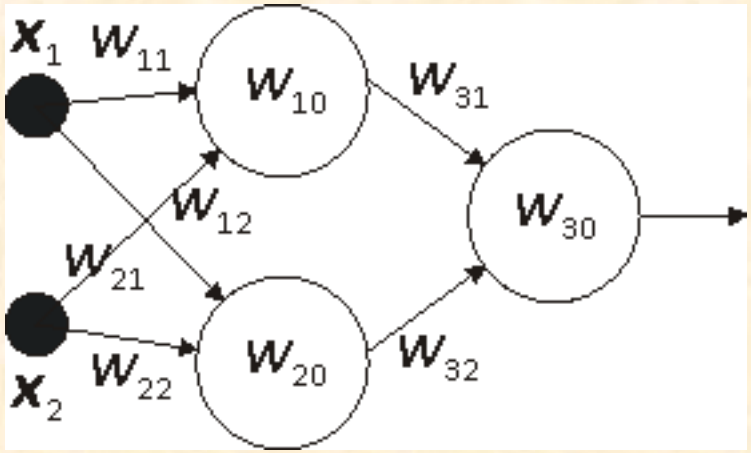
ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Θεωρείστε το ακόλουθο δίκτυο δύο επιπέδων με

-2 νευρώνες εισόδου (διάσταση χώρου εισόδου)

-2 νευρώνες πρώτου επιπέδου

-1 νευρώνας (εξόδου) δευτέρου επιπέδου.



Το δίκτυο αυτό περιγράφεται από τη συνάρτηση

$$\hat{y} = f(w_{31} f(w_{11}x_1 + w_{12}x_2 + w_{10}) + w_{32} f(w_{21}x_1 + w_{22}x_2 + w_{20}) + w_{30})$$

$$= f(\sum_{i=1}^{2} w_{3i} f(\sum_{j=1}^{2} w_{ij}x_j + w_{i0}) + w_{30}) = f(\sum_{i=1}^{2} w_{3i} f(\sum_{j=1}^{2} w_i^T x) + w_{30})$$

όπου $\boldsymbol{w}_1=[w_{11}, w_{12}, w_{10}]^T$, $\boldsymbol{w}_2=[w_{21}, w_{22}, w_{20}]^T$, $\boldsymbol{w}_3=[w_{31}, w_{32}, w_{30}]^T$, $\boldsymbol{x}=[x_1, x_2, 1]^T$ και $f(.)$ η συνάρτηση μοναδιαίου βήματος.

Το διάνυσμα παραμέτρων $\boldsymbol{w}$ του δικτύου είναι

$\mathbf{w}=[\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \boldsymbol{w}_3^T] = [w_{11}, w_{12}, w_{10}, w_{21}, w_{22}, w_{20}, w_{31}, w_{32}, w_{30}]^T$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (Backpropagation (BP) Algorithm)

- Είναι μια αλγοριθμική διαδικασία στη λογική της **οξύτερης καθόδου** (**gradient descent**) που υπολογίζει αναδρομικά τα συναπτικά βάρη (παράμετροι) του δικτύου, έτσι ώστε να επιτυγχάνεται η ελαχιστοποίηση της επιλεγμένης συνάρτησης κόστους.

- Σε ένα μεγάλο αριθμό διαδικασιών βελτιστοποίησης, απαιτείται ο υπολογισμός παραγώγων. Έτσι, ασυνεχείς συναρτήσεις ενεργοποίησης δημιουργούν προβλήματα δηλ.,

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

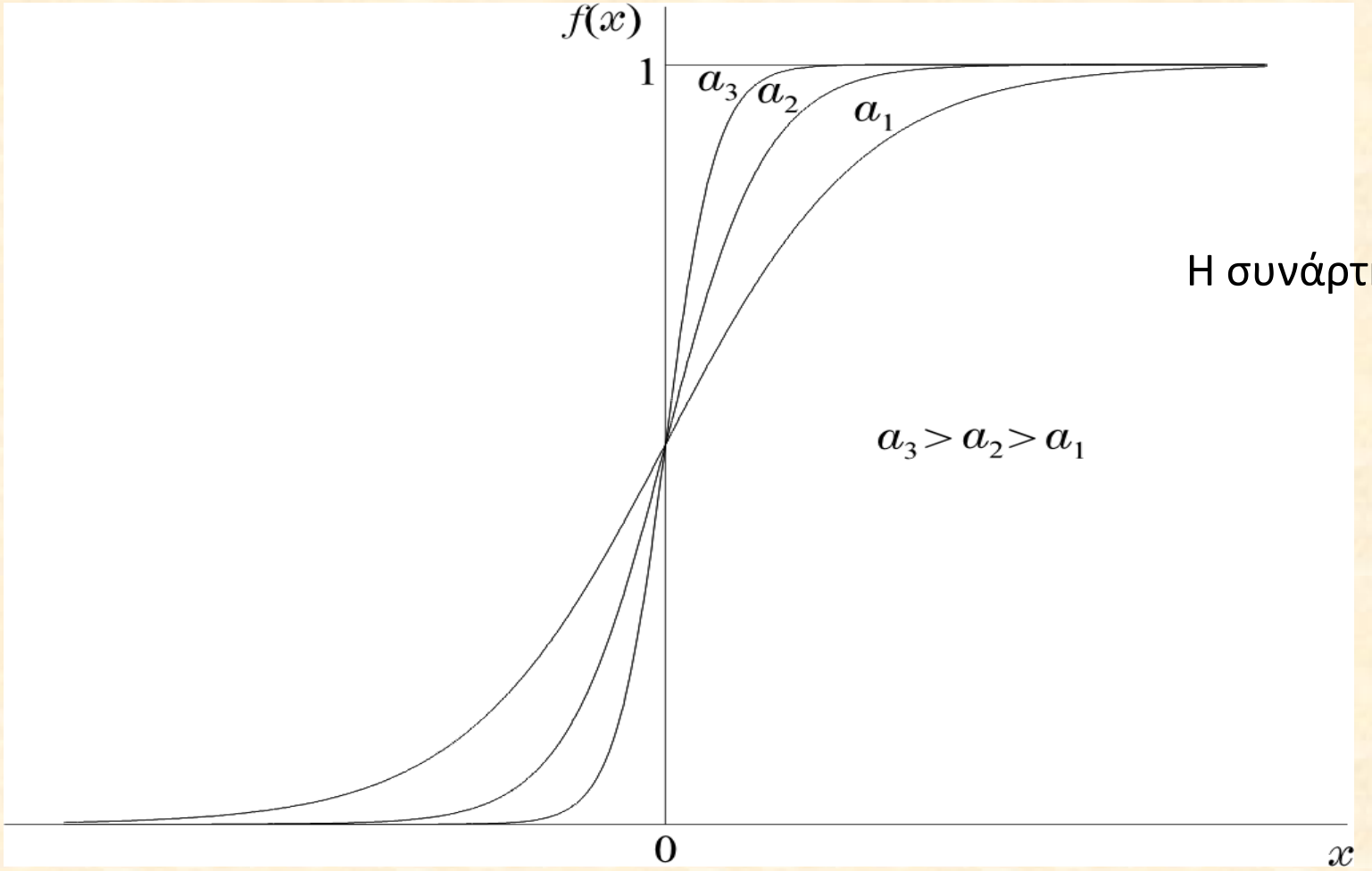
- Παρόλα αυτά, υπάρχουν δίοδοι διαφυγής!!! Η logistic συνάρτηση και η συνάρτηση υπερβολικής εφαπτομένης αποτελούν τέτοια παραδείγματα.

$$f(x) = \frac{1}{1+\exp(-ax)}$$

$$f(x) = \frac{1-\exp(-ax)}{1+\exp(-ax)}$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP)



Η συνάρτηση logistic

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP)

➢ Τα βήματα:

- Υιοθέτησε μία κατάλληλη **συνάρτηση κόστους** $J(\boldsymbol{w})$, π.χ.,

  Σφάλμα ελαχίστων τετραγώνων (Least Squares Error),

  Σχετική εντροπία (Relative Entropy)

  που ορίζεται λαμβάνοντας υπ' όψιν τις επιθυμητές αποκρίσεις και τις πραγματικές αποκρίσεις του δικτύου για τα διαθέσιμα διανύσματα εκπαίδευσης. Αυτό υπονοεί ότι από εδώ και στο εξής αποδεχόμαστε την ύπαρξη λαθών. Προσπαθούμε μόνο να τα ελαχιστοποιήσουμε, χρησιμοποιώντας συγκεκριμένα κριτήρια.

- Υιοθέτησε έναν **αλγόριθμο** για τη βελτιστοποίηση της συνάρτησης κόστους ως προς τα συναπτικά βάρη π.χ.,

  Αλγόριθμος οξύτερης καθόδου (Gradient descent)

  Αλγόριθμος Newton

  Αλγόριθμος συζυγών διευθύνσεων (Conjugate gradient)

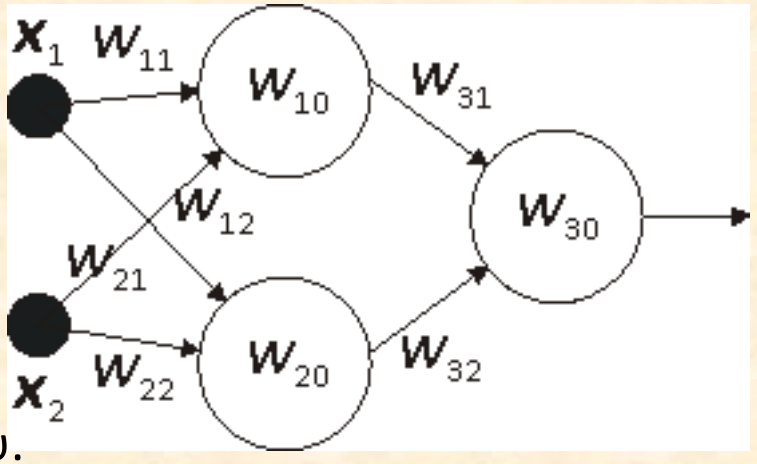
ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP)

Για την περίπτωση της οξύτερης καθόδου έχουμε

$$w_i(\mathrm{t}+1) = w_i(\mathrm{t}) + \Delta w_i(t)$$

$$\Delta w_i(t) = -\mu \frac{\partial J}{\partial w_i}\Big|_{w_i = w_i(t)}$$



Για τα διανύσματα όλων των νευρώνων του δικτύου.

Παράδειγμα: Η συν. κόστους ελαχίστων τετραγώνων για το παραπάνω δίκτυο είναι

$$J(w) = \sum_{n=1}^{N} (d_n - \hat{y}_n)^2 = \sum_{n=1}^{N} (d_n - f(\sum_{i=1}^{2} w_{3i} f(\sum_{j=1}^{2} w_i^T x_n) + w_{30}))^2 =$$

$$\sum_{n=1}^{N} (d_n - f(\sum_{i=1}^{2} w_{3i} f(\sum_{j=1}^{2} w_{ij} x_{nj} + w_{i0}) + w_{30}))^2$$

Batch mode

$$J(w) = (d_n - \hat{y}_n)^2 = (d_n - f(\sum_{i=1}^{2} w_{3i} f(\sum_{j=1}^{2} w_i^T x_n) + w_{30}))^2$$

Pattern (online) mode

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP)

Η **διαδικασία**:

1. **Αρχικοποίησε** τα άγνωστα βάρη σε τυχαίες μικρές τιμές.
2. **Υπολόγισε** τους όρους βαθμίδας (gradient terms) προς τα πίσω, ξεκινώντας με τα βάρη των νευρώνων του τελευταίου επιπέδου και κινούμενος προς αυτά των νευρώνων του πρώτου
3. **Ενημέρωσε** τα βάρη
4. **Επανέλαβε** τα βήματα 2 και 3 μέχρις ότου ικανοποιηθεί ένα κριτήριο τερματισμού.

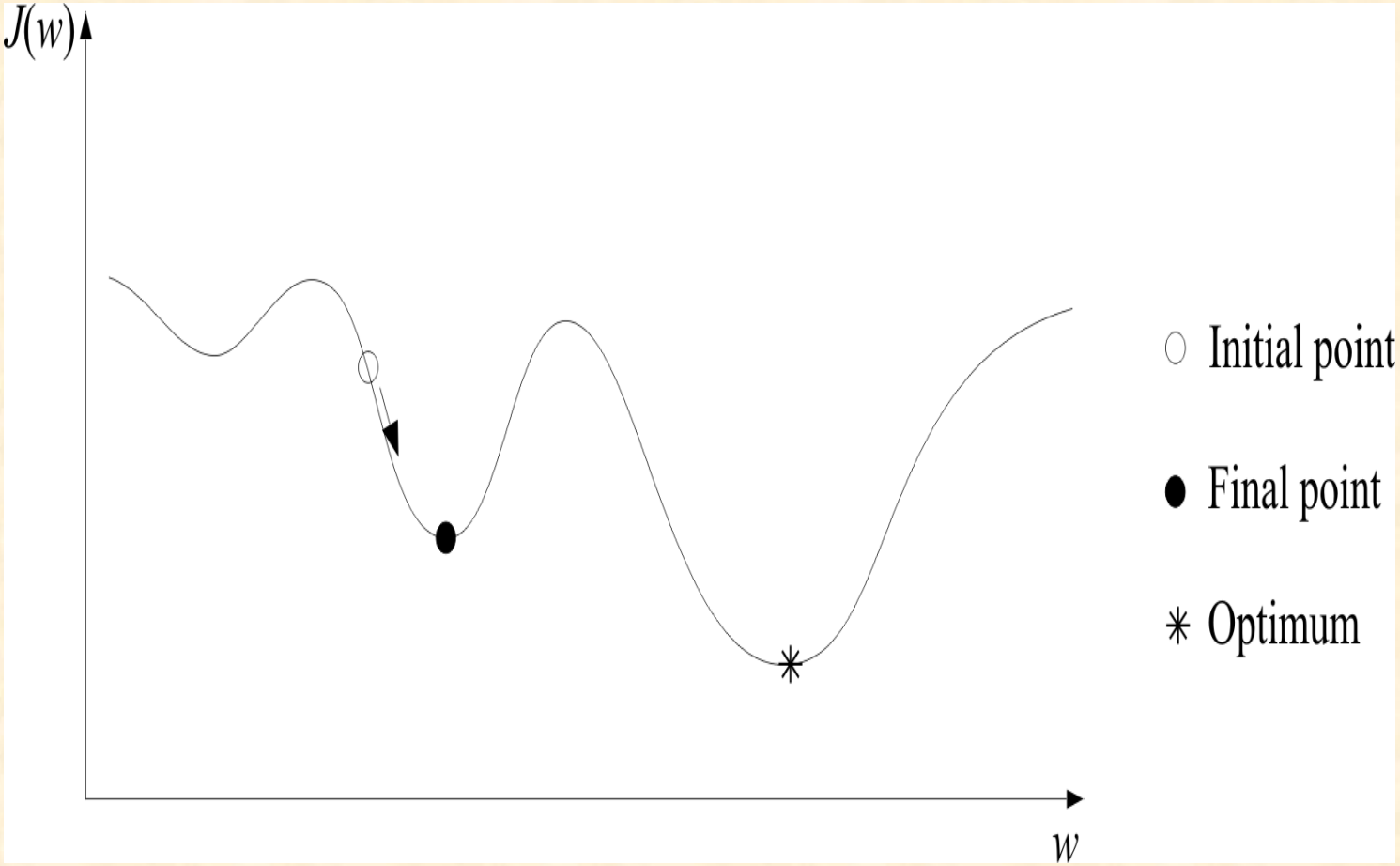
– **Δύο** κύριες **φιλοσοφίες**:

- Επεξεργασία κατά συρροή (Batch mode): Οι όροι βαθμίδας του τελευταίου επιπέδου υπολογίζονται αφού πρώτα υποστούν επεξεργασία ΌΛΑ τα διανύσματα εκπαίδευσης δηλ.,, θεωρώντας το άθροισμα των λαθών για όλα τα διανύσματα.
- Επεξεργασία κατά μόνας (Pattern mode): Οι όροι βαθμίδα υπολογίζονται για κάθε νέο διάνυσμα εκπαίδευσης. Έτσι, οι όροι αυτοί βασίζονται σε διαδοχικά μεμονωμένα λάθη.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP)

Ένα σημαντικό πρόβλημα: Ο αλγόριθμος μπορεί να συγκλίνει σε ένα τοπικό ελάχιστο.



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP) – Επιλογή συνάρτησης κόστους

Παραδείγματα:

- Η συνάρτηση ελαχίστων τετραγώνων (Least Squares)

$$J = \sum_{i=1}^{N} E(i)$$

$$E(i) = \sum_{m=1}^{k} e_m^2(i) = \sum_{m=1}^{k} (y_m(i) - \hat{y}_m(i))^2$$

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

$y_m(i) \rightarrow$ για το Επιθυμητή απόκριση του *m-στου* νευρώνα εξόδου (1 ο $x(i)$

$\hat{y}_m(i) \rightarrow$ Πραγματική απόκριση του *m-στου* νευρώνα εξόδου, στο $x(i)$

διάστημα [0, 1], για είσοδο

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP) – Επιλογή συνάρτησης κόστους

- Η συνάρτηση cross-entropy

$$J = \sum_{i=1}^{N} E(i)$$

$$E(i) = \sum_{m=1}^{k} \left\{ y_m(i) \ln \hat{y}_m(i) + (1 - y_m(i)) \ln(1 - \hat{y}_m(i)) \right\}$$

Αυτή προϋποθέτει την ερμηνεία των y and $\hat{y}$ ως **πιθανότητες.**

- Σφάλμα ταξινόμησης (Classification error rate).

Οι περισσότερες από τις σχετικές τεχνικές χρησιμοποιούν μία εξομαλυσμένη έκδοση του σφάλματος ταξινόμησης και όλες μαζί αποτελούν την κατηγορία αλγορίθμων διακριτικής μάθησης (discriminative learning).

Ο αλγόριθμος οπισθοδρομικής διάδοσης (BP) – Επιλογή συνάρτησης κόστους

- Σχόλιο 1: Ένα κοινό χαρακτηριστικό των παραπάνω συναρτήσεων είναι ο κίνδυνος σύγκλισης σε κάποιο τοπικό ελάχιστο. Οι **"καλώς ορισμένες"** ("**Well formed**") συναρτήσεις κόστους εγγυώνται σύγκλιση σε μία "καλή" λύση, δηλαδή σε μία λύση που να ταξινομεί σωστά ΟΛΑ τα δεδομένα εκπαίδευσης, υπό την προϋπόθεση ότι υπάρχει μία τέτοια λύση. Η cross-entropy συνάρτηση κόστους **είναι** καλώς ορισμένη. Η συνάρτηση ελαχίστων τετραγώνων **δεν είναι**.

- Σχόλιο 2: Αμφότερες οι συναρτήσεις κόστους ελαχίστων τετραγώνων και cross entropy οδηγούν σε τιμές εξόδου $\hat{y}_m(i)$ που προσεγγίζουν κατά βέλτιστο τρόπο τις εκ των υστέρων πιθανότητες για κάθε κλάση (Optimally class a-posteriori probabilities)!!!

$$\hat{y}_m(i) \cong P(\omega_m | \underline{x}(i))$$

Δηλ., την πιθανότητα της κλάσης ω_m δοθέντος του $x(i)$.

Πρόκειται για ένα πολύ ενδιαφέρον αποτέλεσμα. **Δεν** εξαρτάται από τις κατανομές των κλάσεων. Είναι ένα χαρακτηριστικό **ορισμένων** συναρτήσεων κόστους. Η ποιότητα της προσέγγισης εξαρτάται από το μοντέλο που υιοθετήθηκε. Επιπλέον, ισχύει μόνο στο ολικό ελάχιστο.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Παραλλαγές του αλγόριθμου οπισθοδρομικής διάδοσης (BP)

- Αλγόριθμος backpropagation με όρο ορμής (momentum term)
  - Προστατεύει τον αλγόριθμο από περιπτώσεις ταλάντωσης γύρω από το ελάχιστο και, κατά συνέπεια, αργής σύγκλισης
  - Εξισώσεις

$$w_i^r(\mathrm{t}+1) = w_i^r(\mathrm{t}) + \Delta w_i^r(t+1)$$

$$\Delta w_i^r(t+1) = a\Delta w_i^r(t) - \mu \frac{\partial J}{\partial w_i^r}\Big|_{w_i^r = w_i^r(t)}$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Παραλλαγές του αλγόριθμου οπισθοδρομικής διάδοσης (BP)

- Προσαρμοστικός (adaptive) αλγόριθμος backpropagation
  - Επιταχύνει ή επιβραδύνει ανάλογα με το είδος της περιοχής του landscape της συνάρτησης κόστους που βρίσκεται η τρέχουσα εκτίμηση
  - Εξισώσεις

$$\frac{J(t)}{J(t-1)} < 1, \quad \mu(t) = r_i \mu(t-1)$$

$$\frac{J(t)}{J(t-1)} > c, \quad \mu(t) = r_d \mu(t-1)$$

$$1 \leq \frac{J(t)}{J(t-1)} \leq c, \quad \mu(t) = \mu(t-1)$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Μέγεθος – αρχιτεκτονική δικτύου

Ένα σημαντικό ζήτημα: Πώς αποφασίζουμε για το μέγεθος και τη δομή του δικτύου;

Δύο κύριες **κατευθύνσεις**:

- **Σταθερή δομή δικτύου:** Υιοθέτησε ένα δίκτυο συγκεκριμένης σταθερής δομής και εφάρμοσε το αλγόριθμο BP. Αν ή απόδοση του δικτύου μετά την εκπαίδευση δεν είναι ικανοποιητική, υιοθέτησε μια άλλη αρχιτεκτονική και επανέλαβε την εκπαίδευση.
- **Μεταβλητή δομή δικτύου:** Στην περίπτωση αυτή η δομή του δικτύου μεταβάλλεται κατά την διάρκεια της εκπαίδευσης του δικτύου. Υπάρχουν δύο βασικές φιλοσοφίες:
  - Φιλοσοφία κλαδέματος (pruning): Ξεκίνα με ένα δίκτυο μεγάλου μεγέθους και «κλάδευέ το» σταδιακά (αφαιρώντας συναπτικά βάρη και/ή νευρώνες) σύμφωνα με κάποια κριτήρια.
  - Φιλοσοφία κατασκευής (Constructive philosophy): Ξεκίνα με ένα δίκτυο μικρού μεγέθους (ανίκανο να λύσει το υπό μελέτη πρόβλημα) και σταδιακά πρόσθετε νευρώνες έως ότου το δίκτυο μάθει τη διαδικασία ταξινόμησης.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Μέγεθος – αρχιτεκτονική δικτύου

Παράδειγμα φιλοσοφίας κλαδέματος:

Μέθοδοι που βασίζονται στην κανονικοποίηση συνάρτησης (function regularization)

$$J = \sum_{i=1}^{N} E(i) + aE_p(\underline{w})$$

Ο δεύτερος όρος ευνοεί μικρές τιμές για τα βάρη, π.χ.,

$$E_p(\underline{\omega}) = \sum_k h(w_k^2)$$

$$h(w_k^2) = \frac{w_k^2}{w_0^2 + w_k^2}$$

όπου $w_0 \cong 1$

Μετά από μερικά βήματα εκπαίδευσης, τα βάρη με μικρές τιμές απομακρύνονται.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος Backpropagation – θέματα ικανότητας γενίκευσης (Generalization)

Γιατί να μην ξεκινήσουμε με ένα δίκτυο μεγάλου μεγέθους και να αφήσουμε τον αλγόριθμο να αποφασίσει ποια βάρη είναι μικρά;;

Η προσέγγιση αυτή είναι απλοϊκή.

Παραβλέπει το γεγονός ότι οι ταξινομητές πρέπει να έχουν καλή δυνατότητα **γενίκευσης (generalization)**. Ένα μεγάλο δίκτυο μπορεί να δώσει μικρά σφάλματα για το σύνολο εκπαίδευσης, αφού μπορεί να μάθει τις συγκεκριμένες λεπτομέρειές του. Από την άλλη μεριά, δεν αναμένεται να παρουσιάζει καλή απόδοση για δεδομένα στα οποία δεν εκπαιδεύτηκε.

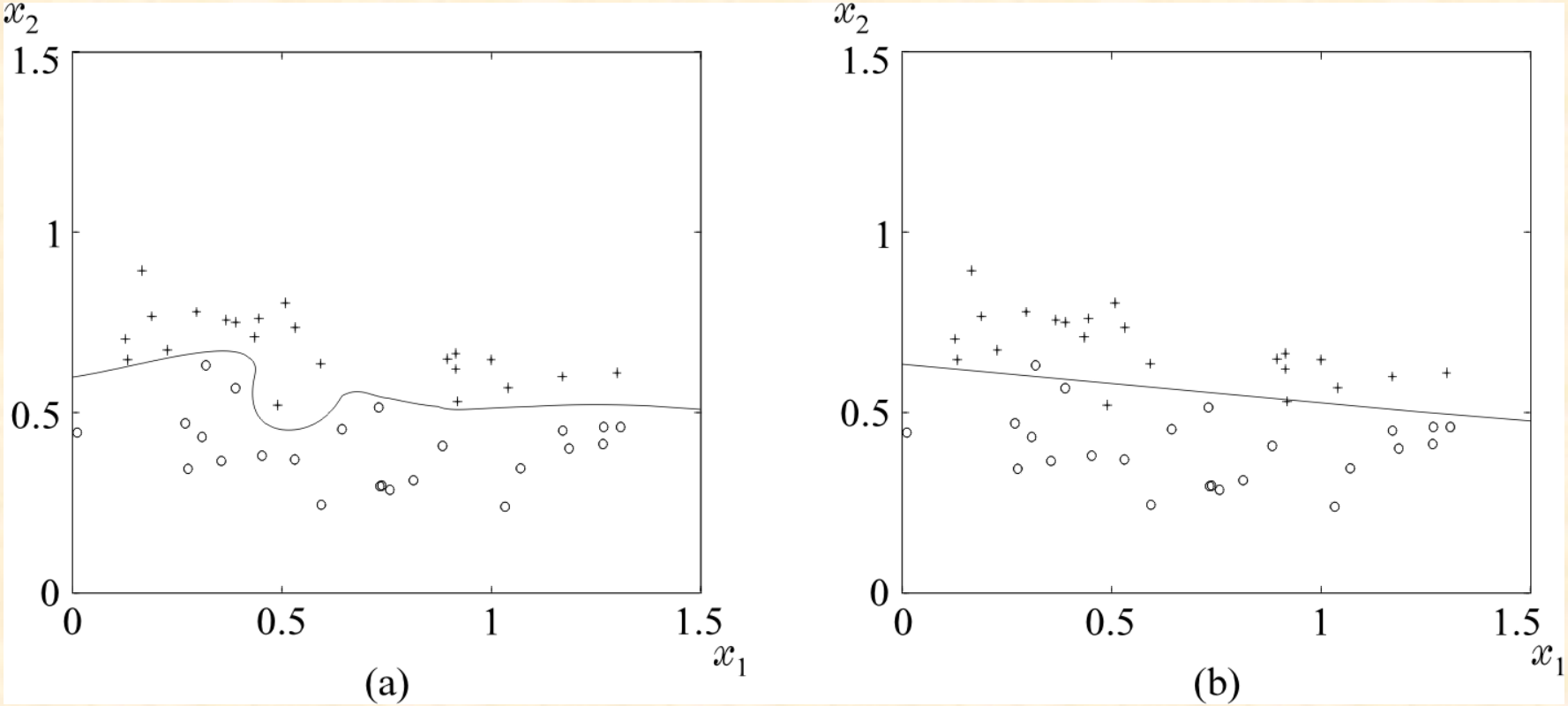
Γενικά, το μέγεθος του δικτύου πρέπει να είναι:

- Αρκούντως μεγάλο για να μπορεί να μάθει τι κάνει όμοια τα δεδομένα της ίδιας κλάσης και τι κάνει ανόμοια τα δεδομένα διαφορετικών κλάσεων.
- Αρκούντως μικρό για να μην μπορεί να μάθει τις διαφορές μεταξύ δεδομένων της ίδιας κλάσης. Το τελευταίο οδηγεί στο λεγόμενο υπερ-ταίριασμα (overfitting).

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ο αλγόριθμος Backpropagation – θέματα ικανότητας γενίκευσης

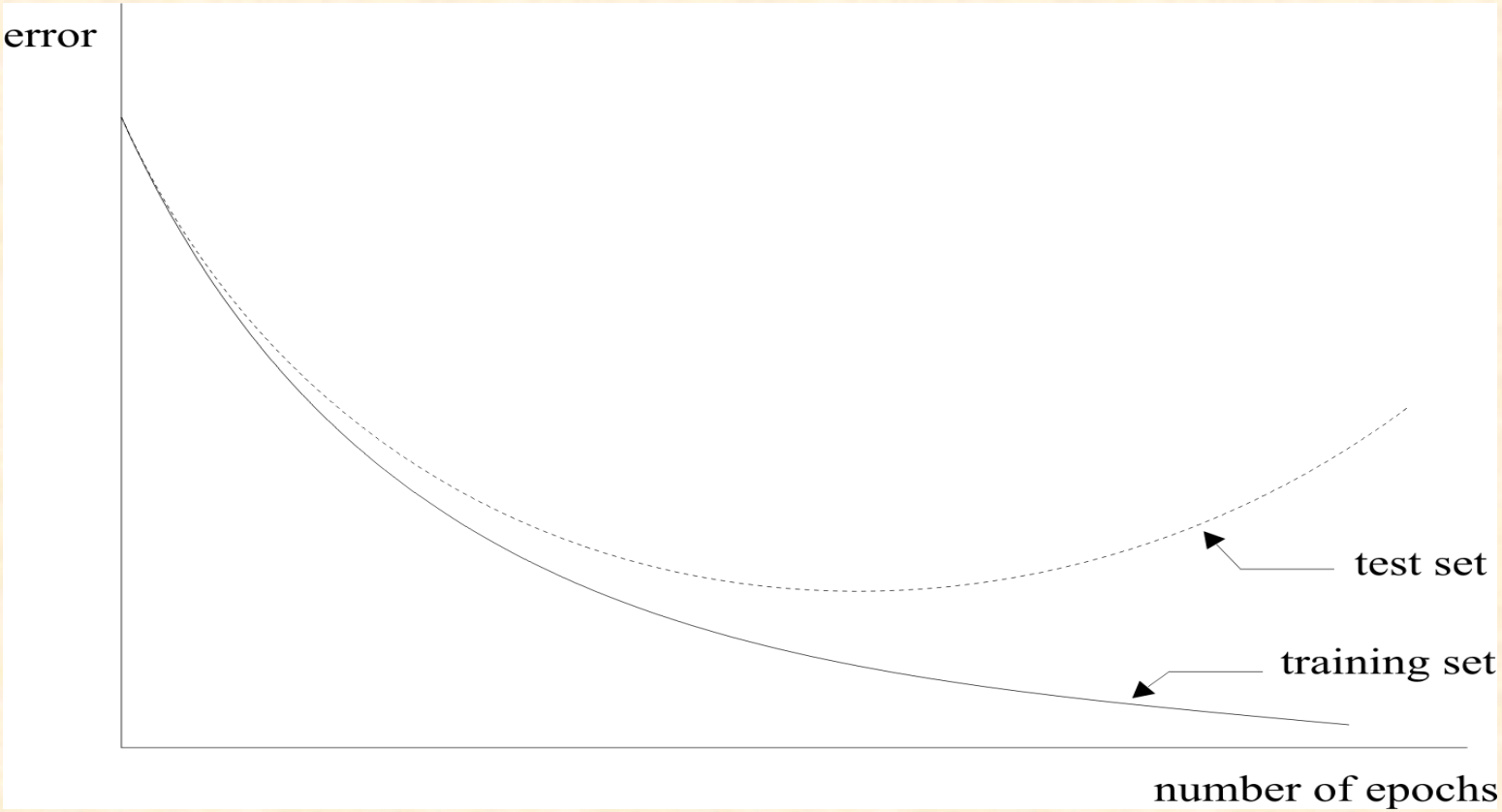
Παράδειγμα:



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

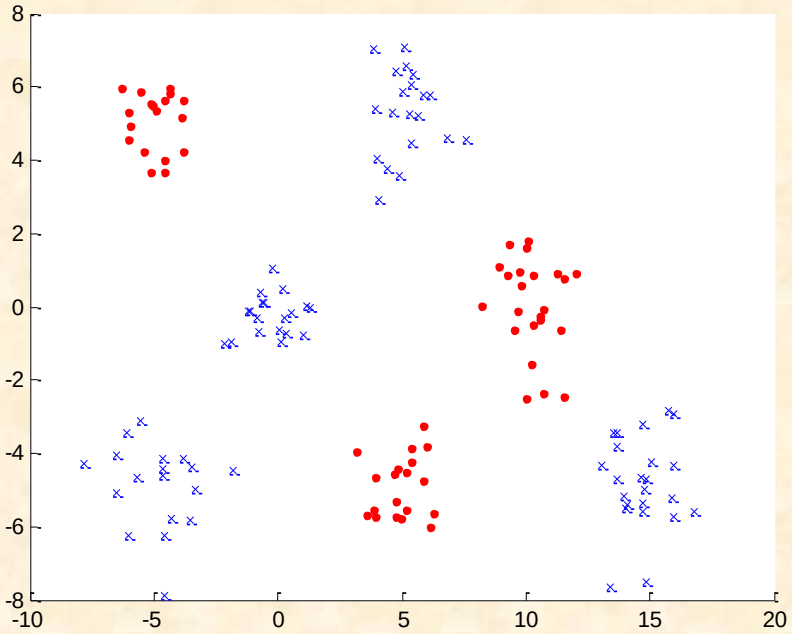
Ο αλγόριθμος Backpropagation

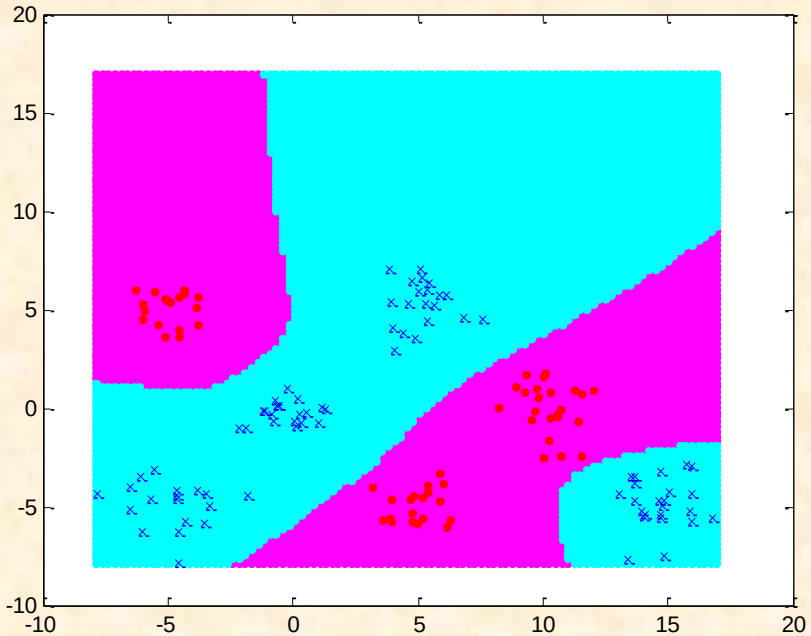
Υπερεκπαίδευση (Overtraining) είναι μία άλλη όψη του ίδιου νομίσματος, δηλ. το δίκτυο προσαρμόζεται στις ιδιαιτερότητες του συνόλου δεδομένων.



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Ένα παράδειγμα:





Όπως αναμενόταν, η επιφάνεια απόφασης δεν ορίζεται μέσω υπερεπιπέδων.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Στο πρόβλημα XOR προηγουμένως, ουσιαστικά μετασχηματίσαμε τον αρχικό χώρο όπου η διαδικασία ταξινόμησης ήταν μη γραμμική σε ένα νέο χώρο όπου η ταξινόμηση έγινε γραμμική.

Διατύπωση προβλήματος: Έστω ένα μη γραμμικώς διαχωρίσιμο πρόβλημα ταξινόμησης στο χώρο R^l. Υπάρχουν k συναρτήσεις $f_i(.)$ μέσω των οποίων ο αρχικός χώρος μπορεί να απεικονιστεί σε ένα νέο k-διάστατο χώρο, όπου το πρόβλημα να είναι γραμμικώς διαχωρίσιμο;

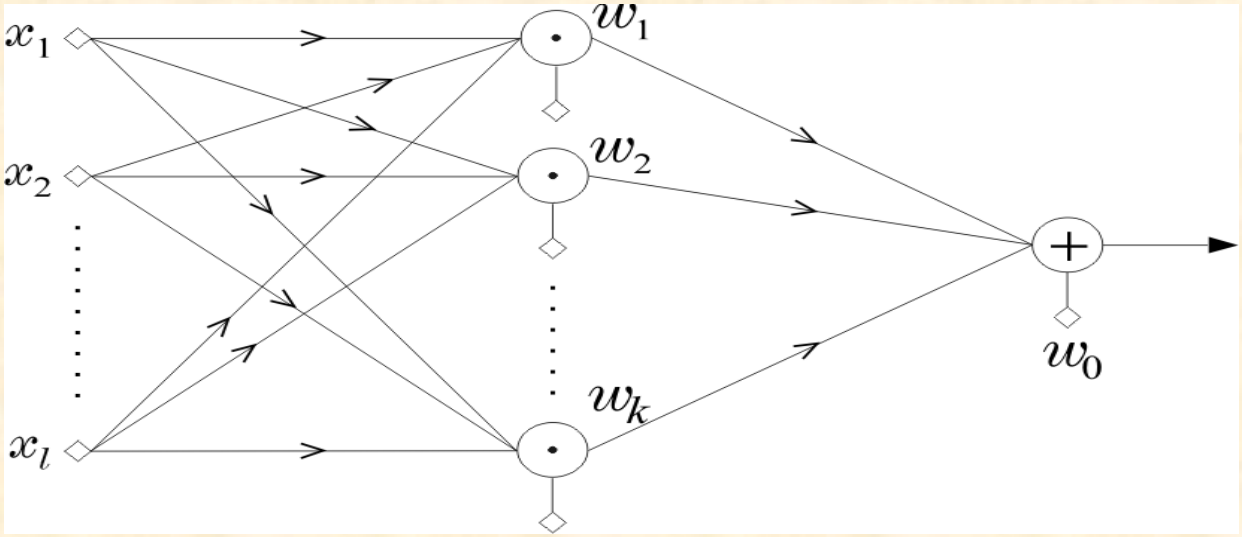
$$\underline{x} \rightarrow \underline{y} = \begin{bmatrix} f_1(\underline{x}) \\ \dots \\ f_k(\underline{x}) \end{bmatrix}$$

Αν συμβαίνει αυτό, η μη γραμμική επιφάνεια-σύνορο στον αρχικό χώρο (***x***) μπορεί να γραφεί ως γραμμική στο μετασχηματισμένο χώρο (***y***).

$$g(\underline{x}) \cong w_0 + \sum_{i=1}^{k} w_i f_i(\underline{x}) \quad (><) \quad 0$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

ΣΗΜ.: Η παραπάνω απεικόνιση $\boldsymbol{x} \rightarrow \boldsymbol{y}$ μπορεί να υλοποιηθεί από ένα δίκτυο δύο επιπέδων, όπου οι νευρώνες του πρώτου επιπέδου υλοποιούν τις συναρτήσεις $f_i(.)$ ενώ ο νευρώνας του δευτέρου επιπέδου πραγματοποιεί γραμμικό διαχωρισμό στο μετασχηματισμένο χώρο.



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

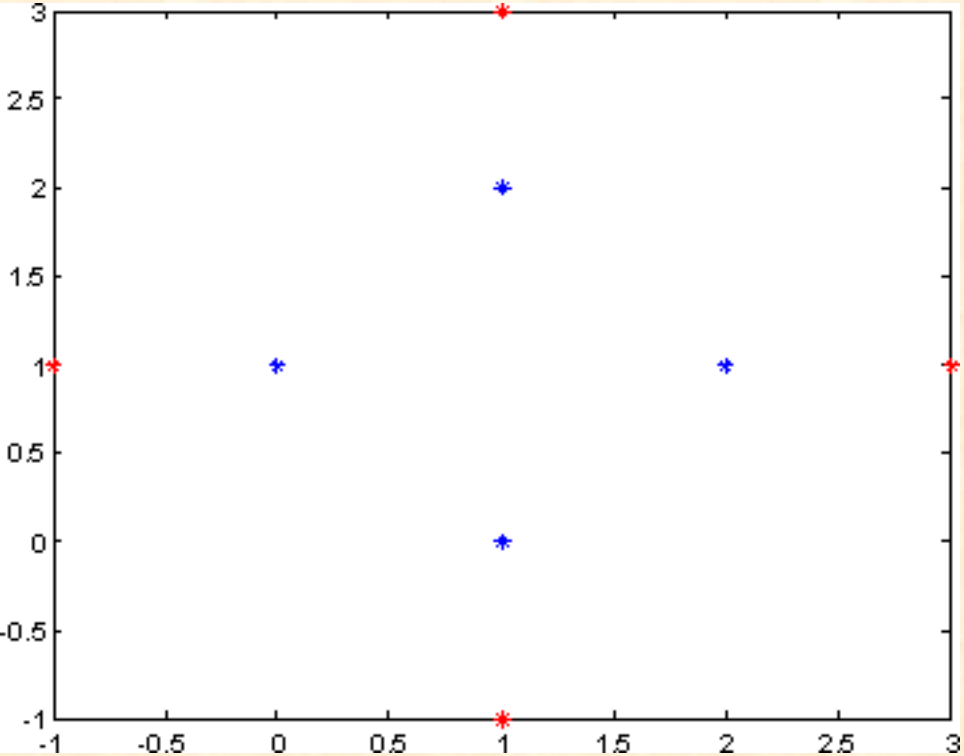
Παράδειγμα 1: Έστω

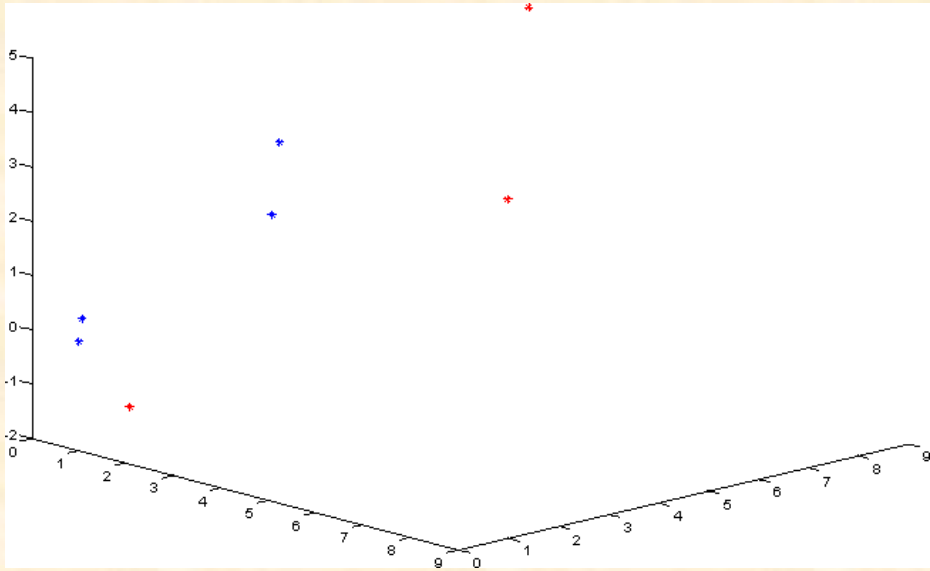
$\boldsymbol{x}_1$=(2,1), $\boldsymbol{x}_2$=(1,2), $\boldsymbol{x}_3$ =(0,1), $\boldsymbol{x}_4$ =(1,0), $\boldsymbol{x}_5$ =(3,1), $\boldsymbol{x}_6$ =(1,3), $\boldsymbol{x}_7$ =(-1,1), $\boldsymbol{x}_8$ =(1,-1).

Τα πρώτα 4 διανύσματα ανήκουν στην κλάση +1 ενώ τα υπόλοιπα στην κλάση -1.

Έστω ο μετασχηματισμός $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{bmatrix}$

Στο νέο χώρο η διαδικασία ταξινόμησης γίνεται γραμμική





ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

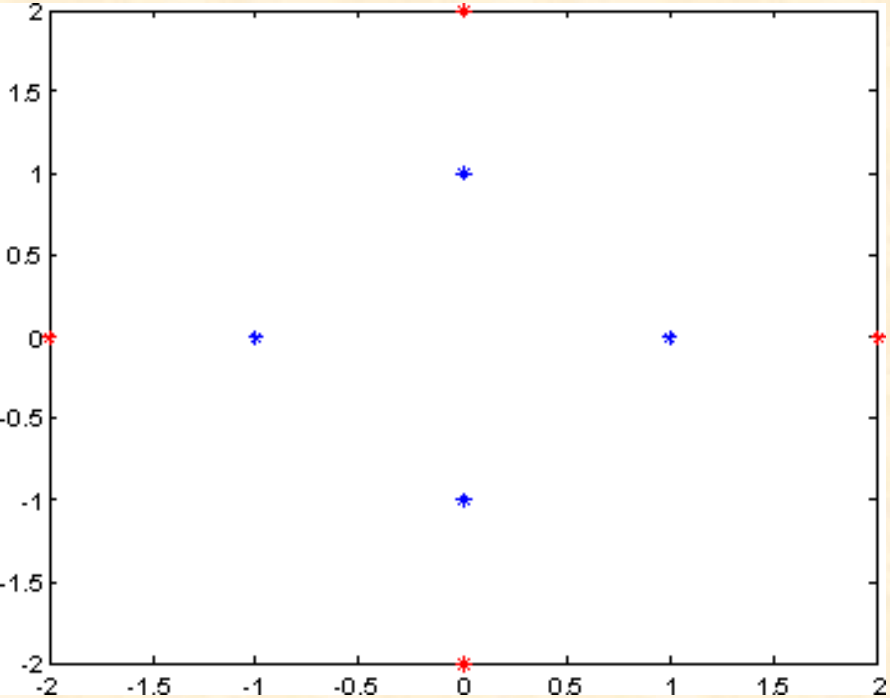
Παράδειγμα 2: Έστω

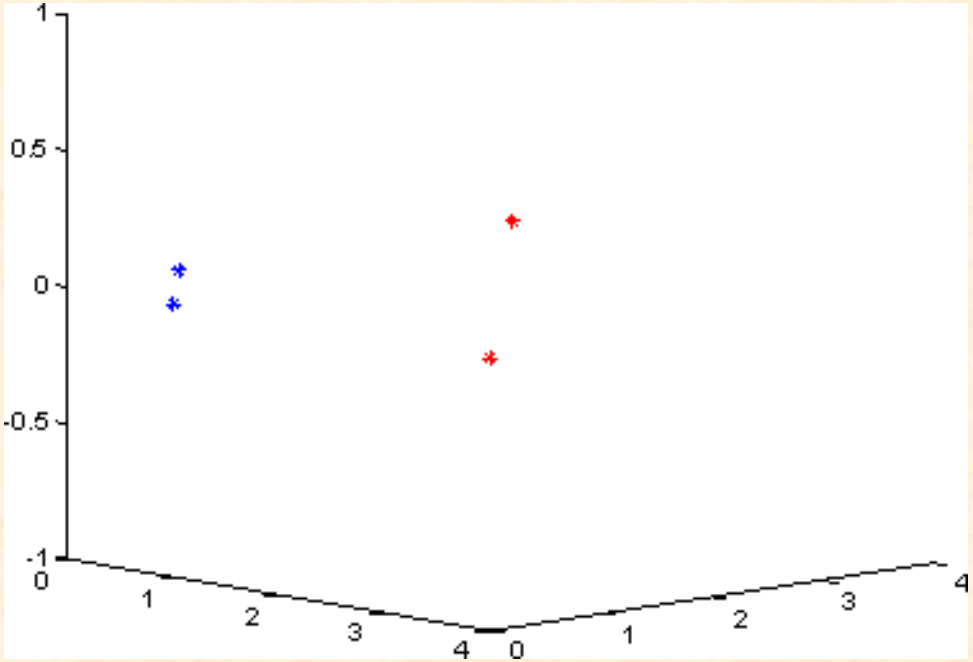
$\boldsymbol{x}_1$=(1,0), $\boldsymbol{x}_2$=(0,1), $\boldsymbol{x}_3$ =(-1,0), $\boldsymbol{x}_4$ =(0,-1), $\boldsymbol{x}_5$ =(2,0), $\boldsymbol{x}_6$ =(0,2), $\boldsymbol{x}_7$ =(-2,0), $\boldsymbol{x}_8$ =(0,-2).

Τα πρώτα 4 διανύσματα ανήκουν στην κλάση +1 ενώ τα υπόλοιπα στην κλάση -1.

Έστω ο μετασχηματισμός $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{bmatrix}$

Στο νέο χώρο η διαδικασία ταξινόμησης γίνεται γραμμική



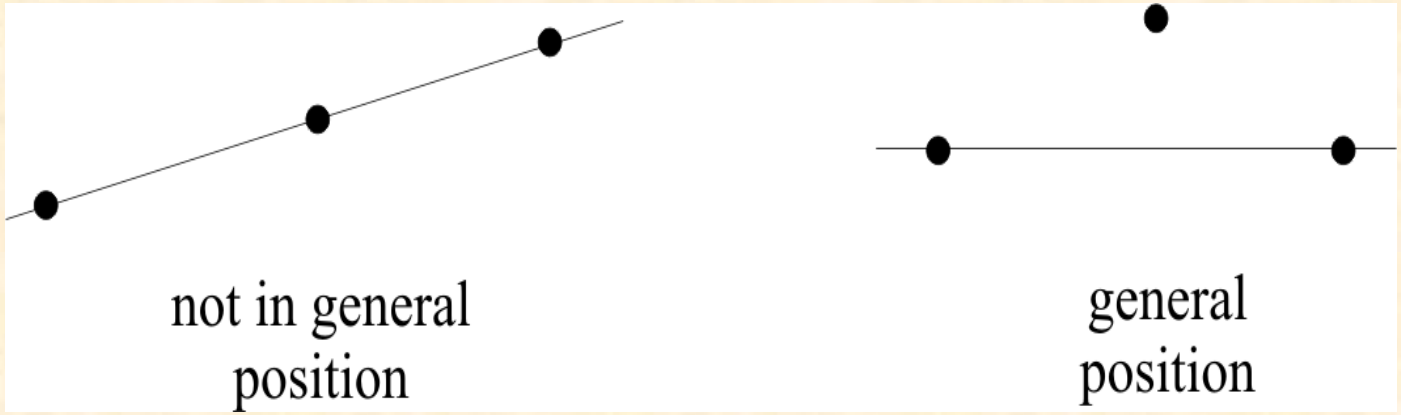


ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Χωρητικότητα του *l*-διάστατου χώρου σε γραμμικές διχοτομήσεις

Έστω *N* σημεία στον R^l χώρο που υποθέτουμε ότι βρίσκονται σε γενική θέση, δηλαδή:

Καμία ομάδα $\ell + 1$ τέτοιων σημείων δεν βρίσκεται σε έναν $\ell - 1$ διάστατο χώρο.



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

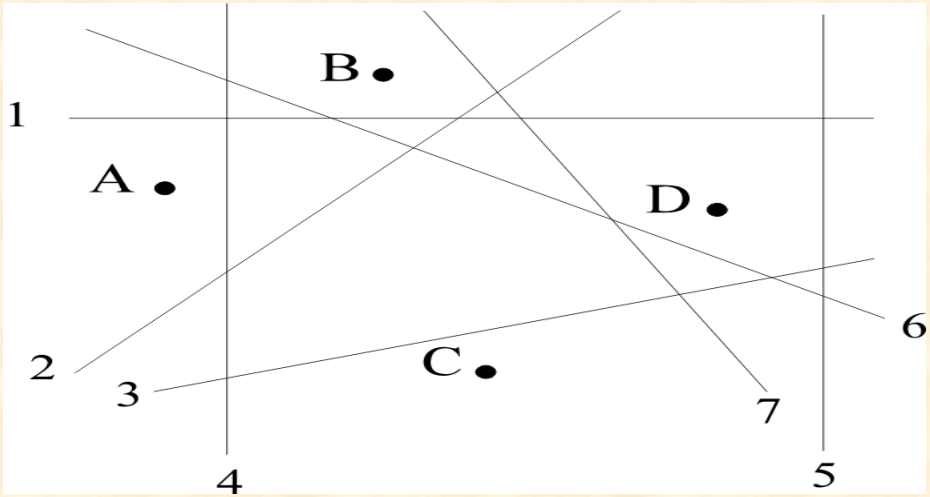
Χωρητικότητα του *l*-διάστατου χώρου σε γραμμικές διχοτομήσεις

Θεώρημα Cover: Ο αριθμός των ομαδοποιήσεων που μπορούν να υλοποιηθούν από (*l*-1)-διάστατα υπερεπίπεδα προκειμένου να διαχωριστούν *N* σημεία σε δύο κλάσεις είναι

$$O(N,l) = 2\sum_{i=0}^{l}\binom{N-1}{i}, \qquad \binom{N-1}{i} = \frac{(N-1)!}{(N-1-i)!i!}$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Παράδειγμα: Για $N=4$, $l=2$, είναι $O(4,2)=14$



Σημείωση:Ο συνολικός αριθμός δυνατών ομαδοποιήσεων είναι $2^4=16$

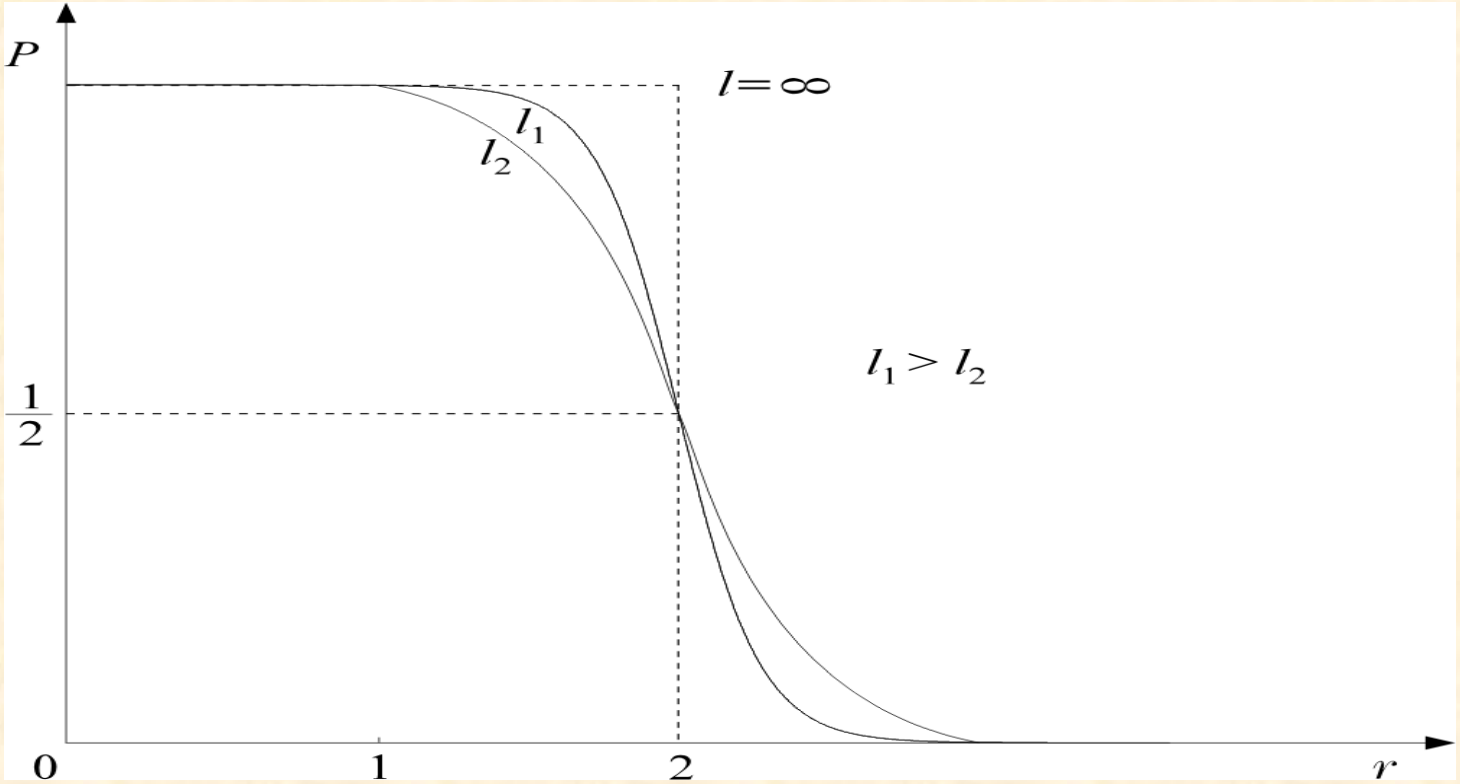
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	✓	✓	✓	✓	✓	✓	X	✓	✓	X	✓	✓	✓	✓	✓	✓

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Χωρητικότητα του *l*-διάστατου χώρου σε γραμμικές διχοτομήσεις

Η πιθανότητα ομαδοποίησης *N* σημείων σε δύο γραμμικώς διαχωρίσιμες κλάσεις είναι

$$\frac{O(N,l)}{2^N} = P_N^l$$



$$N = r(l+1)$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Χωρητικότητα του *l*-διάστατου χώρου σε γραμμικές διχοτομήσεις

Έτσι, η πιθανότητα να έχουμε *N* σημεία σε γραμμικώς διαχωρίσιμες κλάσεις τείνει στο 1, για μεγάλο *l* , υπό την προϋπόθεση ότι $N<2(l+1)$

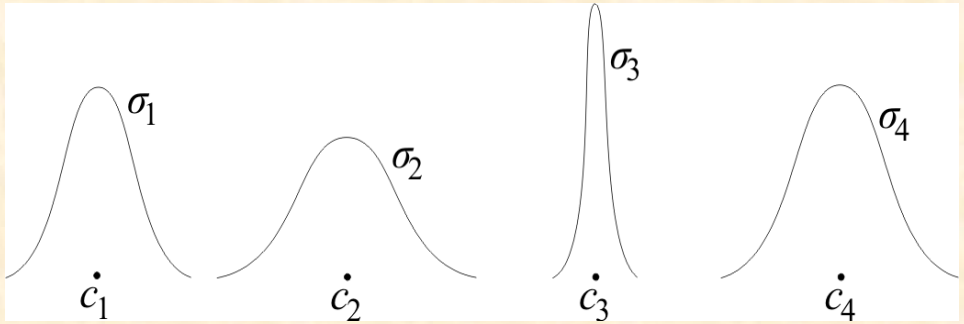
Συνεπώς, απεικονίζοντας σε χώρο υψηλότερης διάστασης, αυξάνουμε την πιθανότητα γραμμικού διαχωρισμού, υπό την προϋπόθεση ότι ο χώρος δεν παρουσιάζει μεγάλη πυκνότητα σε σημεία δεδομένων.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

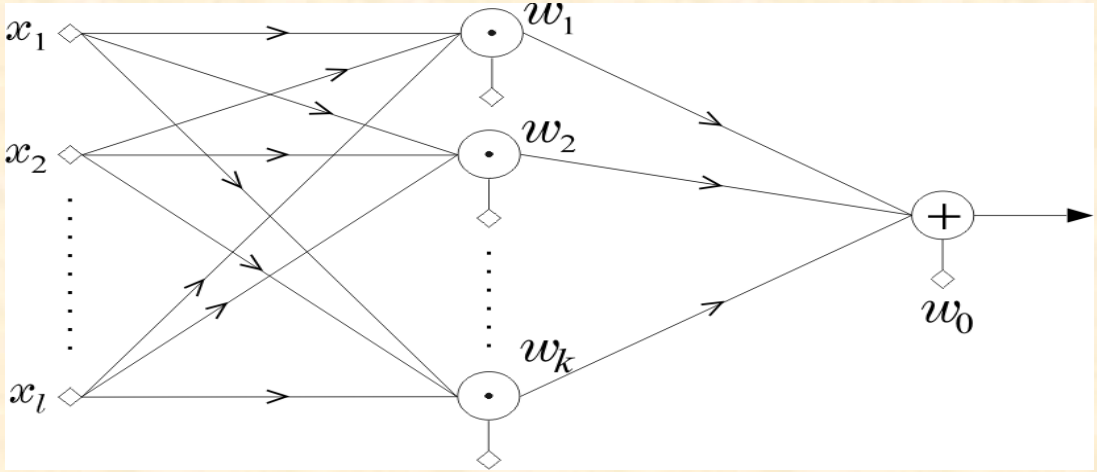
Συναρτήσεις ακτινωτής βάσης (Radial basis functions -RBFs)

➢Στην περίπτωση αυτή, οι $f_i(.)$ επιλέγονται ως $f_i(\underline{x}) = \exp\left(-\frac{\|x-\underline{c}_i\|^2}{2\sigma_i^2}\right)$

➢Έχουν σχήμα «καμπάνας», είναι κεντραρισμένες στο c_i και το πλάτος τους εξαρτάται από το σ_i.



➢Η παραπάνω απεικόνιση ***x*→*y*** μπορεί να υλοποιηθεί από ένα δίκτυο δύο επιπέδων, όπου οι νευρώνες του πρώτου επιπέδου υλοποιούν τις συναρτήσεις $f_i(.)$ ενώ ο νευρώνας του δευτέρου επιπέδου πραγματοποιεί γραμμικό διαχωρισμό στο μετασχηματισμένο χώρο.



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Συναρτήσεις ακτινωτής βάσης (Radial basis functions-RBFs)

Παράδειγμα: Το πρόβλημα *XOR*.

Έστω

$$c_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ c_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \sigma_1 = \sigma_2 = \frac{1}{\sqrt{2}}$$

Χρησιμοποιώντας συναρτήσεις RBF με τις παραπάνω παραμέτρους επιτυγχάνουμε την απεικόνιση του αρχικού 2-διάστατου χώρου σε ένα νέο μετασχηματισμένο χώρο ίδιας διάστασης.

$$x \to y = \begin{bmatrix} \exp(-\|x - c_1\|^2) \\ \exp(-\|x - c_2\|^2) \end{bmatrix}$$

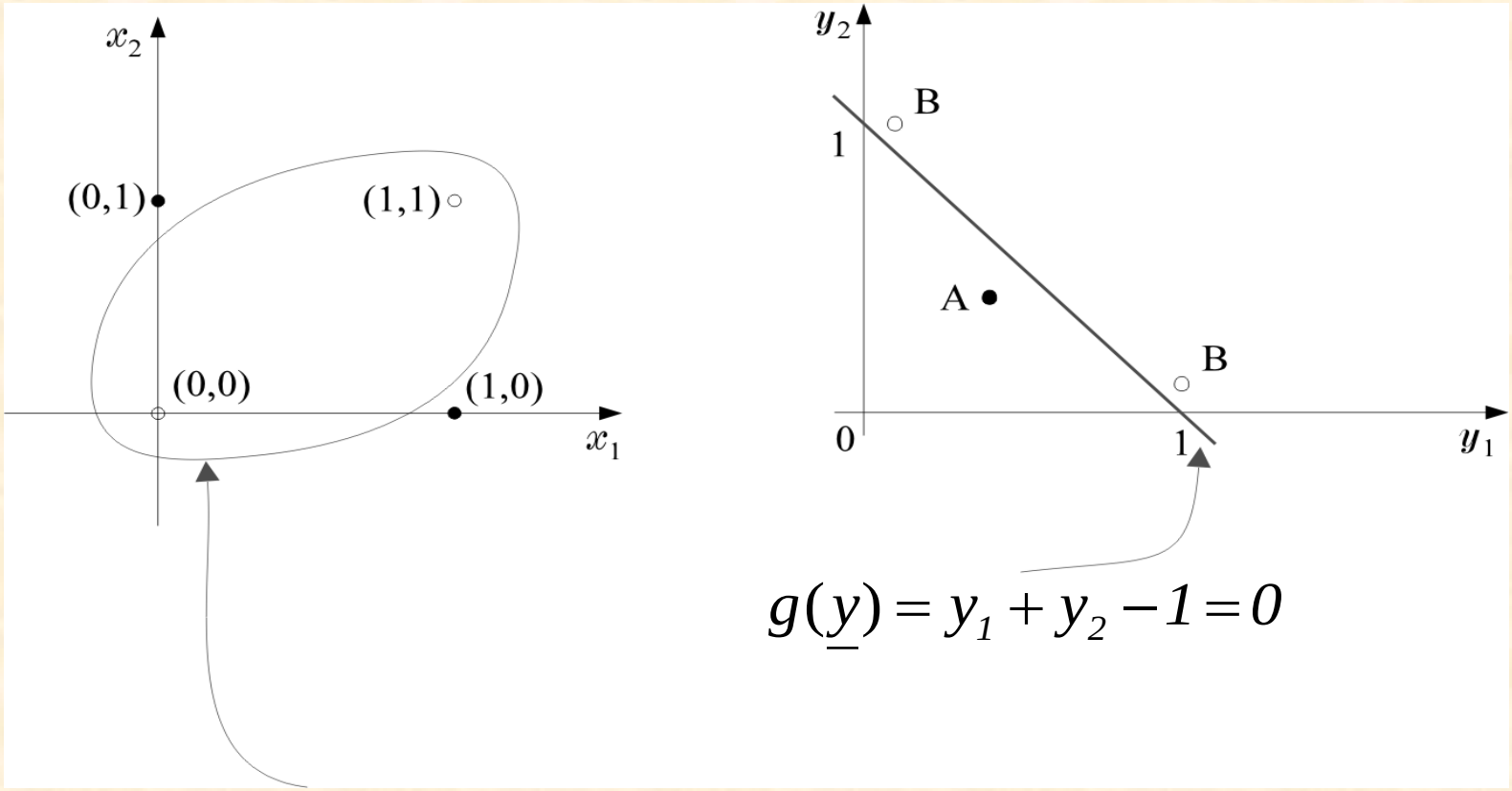
Μέσω αυτού του μετασχηματισμού έχουμε

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \to \begin{bmatrix} 0.135 \\ 1 \end{bmatrix},\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to \begin{bmatrix} 1 \\ 0.135 \end{bmatrix},\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \to \begin{bmatrix} 0.368 \\ 0.368 \end{bmatrix},\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \to \begin{bmatrix} 0.368 \\ 0.368 \end{bmatrix}$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Συναρτήσεις ακτινωτής βάσης (Radial basis functions-RBFs)

Παράδειγμα: Το πρόβλημα *XOR*.



$$g(\underline{x}) = \exp(-\|\underline{x} - \underline{c}_1\|^2) + \exp(-\|\underline{x} - \underline{c}_2\|^2) - 1 = 0$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Συναρτήσεις ακτινωτής βάσης - ζητήματα εκπαίδευσης

Οι εμπλεκόμενες παράμετροι είναι τα $\boldsymbol{c}_i$, τα σ_i και τα $\boldsymbol{w}_i$.

-σ_i: Συνήθως τίθενται εκ των προτέρων σε σταθερές τιμές (υπάρχουν όμως και άλλες εναλλακτικές).

-$\boldsymbol{c}_i$: Μπορεί

- είτε να τεθούν ίσα με τυχαία επιλεγμένα διανύσματα εκπαίδευσης ή
- είτε να τεθούν ίσα με τα κέντρα των ομάδων (clusters) που παράγονται μετά την εφαρμογή ενός αλγόριθμου ομαδοποίησης στα σημεία κάθε κλάσης.

-$\boldsymbol{w}_i$: Υπό την προϋπόθεση ότι τα $\boldsymbol{c}_i$ και σ_i έχουν εκτιμηθεί, η εκτίμηση των $\boldsymbol{w}_i$ είναι μία γραμμική διαδικασία (μπορεί π.χ. να χρησιμοποιηθεί μια μέθοδος ελαχίστων τετραγώνων).

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Ας θυμηθούμε ότι η πιθανότητα να έχουμε γραμμικώς διαχωρίσιμες κλάσεις αυξάνει καθώς **αυξάνει** και η **διάσταση** των χώρου χαρακτηριστικών. Έστω η απεικόνιση:

$$x \in R^l \rightarrow y \in R^k, \; k > l$$

Τότε μπορούμε να χρησιμοποιήσουμε SVM στο χώρο R^k.

Υπενθυμίζουμε ότι το δυϊκό πρόβλημα που θα λύσουμε θα είναι το

$$\underset{\lambda \geq 0}{\text{maximize}} \; (\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j d_i d_j y_i^T y_j)$$

$$\text{οπου } y_i \in R^k$$

Ο ταξινομητής στο μετασχηματισμένο χώρο (***y***) εκφράζεται ως

$$g(\underline{y}) = w^T y + w_0 = \sum_{i=1}^{N_s} \lambda_i d_i y_i^T y$$

$$\text{οπου } x \rightarrow y \in R^k$$

Έτσι, εμπλέκονται εσωτερικά γινόμενα σε ένα χώρο υψηλής διάστασης→ υψηλή υπολογιστική πολυπλοκότητα.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

- Κάτι έξυπνο: Υπολόγισε τα εσωτερικά γινόμενα στον χώρο **υψηλής** διάστασης σαν συνάρτηση των εσωτερικών γινομένων στο χώρο **χαμηλής** διάστασης!!!

- Είναι αυτό ΔΥΝΑΤΟΝ;; Ναι. Να ένα παράδειγμα

$$\text{Εστω } x = [x_1, x_2]^T \in R^2$$

$$\text{Εστω } x \to y = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix} \in R^3$$

Τότε είναι εύκολο να δείξουμε ότι

$$y_i^T y_j = (x_i^T x_j)^2$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

➢Θεώρημα του Mercer: Έστω $\boldsymbol{x} \in R^l$ και μια απεικόνιση f

$$\boldsymbol{x} \rightarrow f(\boldsymbol{x}) \in H,$$

όπου H είναι ένας χώρος Hilbert[(1)]. Για την πράξη του εσωτερικού γινομένου στο χώρο H, που συμβολίζεται με <.,.>, ισχύει

$$<f(\boldsymbol{x}), f(\boldsymbol{z})> = K(\boldsymbol{x},\boldsymbol{z}),$$

όπου $K(\boldsymbol{x},\boldsymbol{z})$, μία συνεχής συμμετρική συνάρτηση (πυρήνας – kernel) που ικανοποιεί τη συνθήκη

$$\int_C \int_C K(x,z) g(x) g(z) dx dz \geq 0$$

για κάθε $g(\boldsymbol{x})$, $\boldsymbol{x} \in C \subset R^l$ με $\int g^2(\underline{x}) d\underline{x} < +\infty,$

➢Αντίστροφα: Κάθε συνεχής, συμμετρική συνάρτηση $K(\boldsymbol{x},\boldsymbol{z})$ που ικανοποιεί τις παραπάνω συνθήκες, αντιστοιχεί στο εσωτερικό γινόμενο ΚΑΠΟΙΟΥ χώρου.

(1) Χώρος Hilbert: Πλήρης γραμμικός (πεπερασμένης ή άπειρης διάστασης) χώρος εξοπλισμένος με μια πράξη εσωτερικού γινομένου. Όταν η διάσταση είναι πεπερασμένη έχουμε έναν Ευκλείδειο χώρο.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Παραδείγματα συναρτήσεων πυρήνων

Συναρτήσεις ακτινωτής βάσης (Radial basis functions) (για κατάλληλες τιμές του σ)

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

Πολυωνυμικές συναρτήσεις (Polynomial functions) (για κατάλληλες τιμές του q)

$$K(\underline{x},\underline{z}) = (\underline{x}^T\underline{z}+1)^q,\ q>0$$

Συνάρτηση υπερβολικής εφαπτομένης (Hyperbolic tangent functions) (για κατάλληλες τιμές των β και γ)

$$K(\underline{x},\underline{z}) = \tanh(\beta\underline{x}^T\underline{z}+\gamma)$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Μια σημαντική παρατήρηση: Στο παρόν πλαίσιο, υπονοείται μια μη γραμμική απεικόνιση

$$f: \boldsymbol{x} \in R^l \rightarrow \boldsymbol{y} \in R^k$$

η οποία, ωστόσο, είναι άγνωστη. Έτσι, για δεδομένο διάνυσμα $\boldsymbol{x}$, δεν είναι γνωστή η εικόνα του $\boldsymbol{y}=f(\boldsymbol{x})$ στο μετασχηματισμένο χώρο (με άλλα λόγια, ο μετασχηματισμένος χώρος δεν μας είναι ρητά γνωστός).

Αυτό που μας είναι γνωστό είναι η συνάρτηση $K(.,.)$ (η οποία καλείται συνάρτηση πυρήνα – kernel function) η οποία εκφράζει το εσωτερικό γινόμενο δύο διανυσμάτων στο μετασχηματισμένο (υψηλής διάστασης) χώρο, σε συνάρτηση με το αντίστοιχο εσωτερικό γινόμενο στον αρχικό χώρο, δηλ.,

$$\boldsymbol{y}_i^T \boldsymbol{y}_j = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Μ’ άλλα λόγια, για δύο δεδομένα διανύσματα, μας είναι γνωστό το εσωτερικό γινόμενό τους στο μετασχηματισμένο χώρο.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Τα βήματα για τη λύση του SVM προβλήματος

- *Βήμα 1:* Επέλεξε κατάλληλη συνάρτηση πυρήνα. Αυτό υπονοεί απεικόνιση σε έναν (άγνωστο) χώρο υψηλότερης διάστασης.

- *Βήμα 2:*

$$\max_{\underline{\lambda}} \left(\sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j d_i d_j K(x_i, x_j)\right)$$

$$\text{subject to: } 0 \le \lambda_i \le C,\ i = 1,2,\ldots,N$$

$$\sum_i \lambda_i d_i = 0$$

Αυτό καταλήγει σε έναν υπονοούμενο συνδυασμό

$$w = \sum_{i=1}^{N_s} \lambda_i d_i f(x_i)$$

- *Βήμα 3:* Καταχώρησε ένα δεδομένο ***x*** στην κλάση +1 (-1), ανάλογα με το αν

$$+1(-1) \text{ if } g(x) = \sum_{i=1}^{N_s} d_i y_i K(x_i, x) + w_0 > (<) 0$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Η λογική πίσω από το βήμα 3:

- Έστω ***x*** δεδομένο διάνυσμα και ***y***=*f*(***x***) η εικόνα του στο μετασχηματισμένο χώρο.
- Προκειμένου να ταξινομήσουμε το ***x*** σε μία από τις κλάσεις +1 ή -1, πρέπει να ελέγξουμε αν ή ποσότητα

$$\boldsymbol{w}^T\boldsymbol{y}+w_0$$

είναι θετική ή αρνητική.

- Ωστόσο, αφού η απεικόνιση *f*(.) είναι άγνωστη, δεν γνωρίζουμε ούτε το ***y*** ούτε τα $\boldsymbol{y}_i=f(\boldsymbol{x}_i)$ που εμπλέκονται στον τύπο του ***w*** (που επαναλαμβάνεται παρακάτω).

$$w=\sum_{i=1}^{N_s}\lambda_i d_i f(x_i)$$

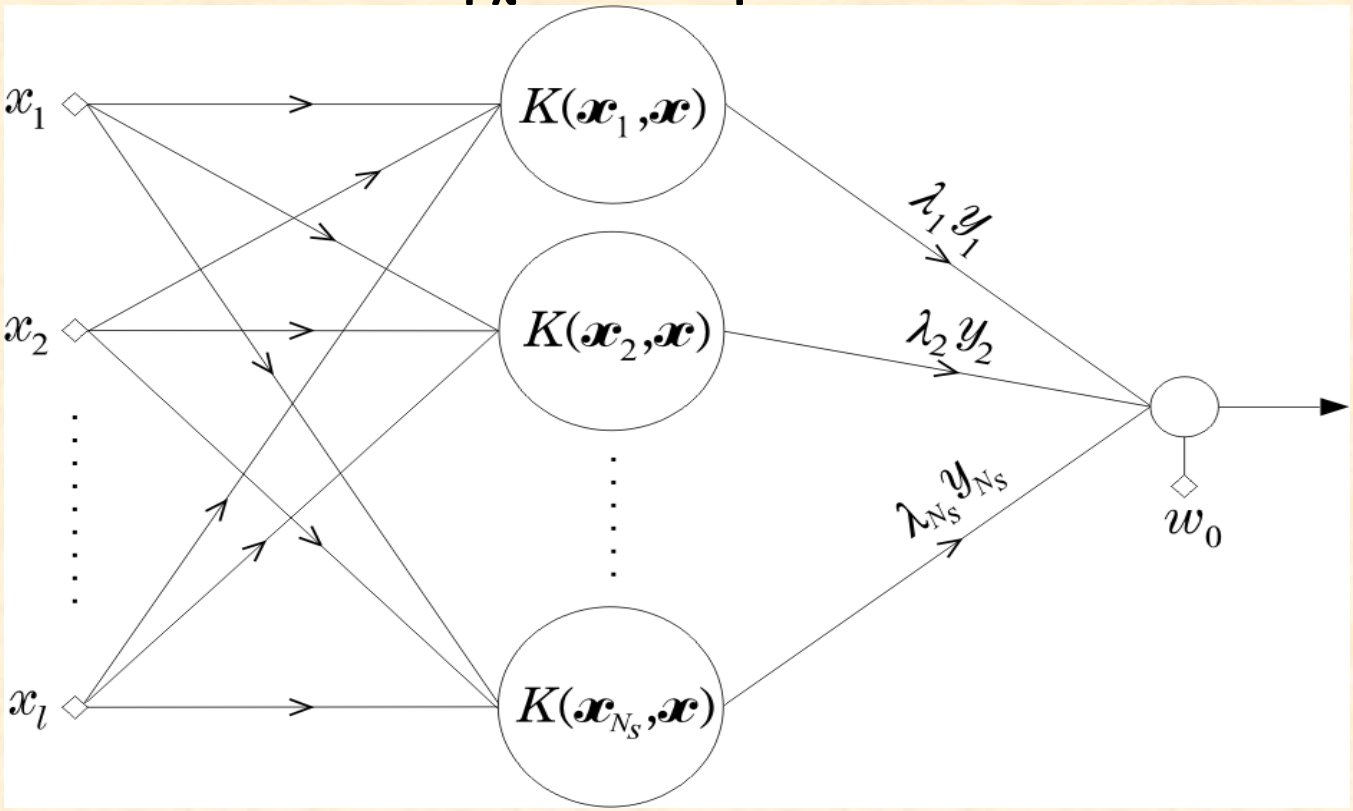
- Παρ' όλες τις παραπάνω δυσχέρειες, η ποσότητα $\boldsymbol{w}^T\boldsymbol{y}+w_0$ μπορεί να υπολογιστεί ως εξής

$$w^T y+w_0=\sum_{i=1}^{N_s}\lambda_i d_i y_i^T y+w_0=\sum_{i=1}^{N_s}\lambda_i d_i f(x_i)^T f(x)+w_0=$$
$$\sum_{i=1}^{N_s}\lambda_i d_i y_i^T y+w_0=\sum_{i=1}^{N_s}\lambda_i d_i K(x_i,x)+w_0$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Η αρχιτεκτονική SVM

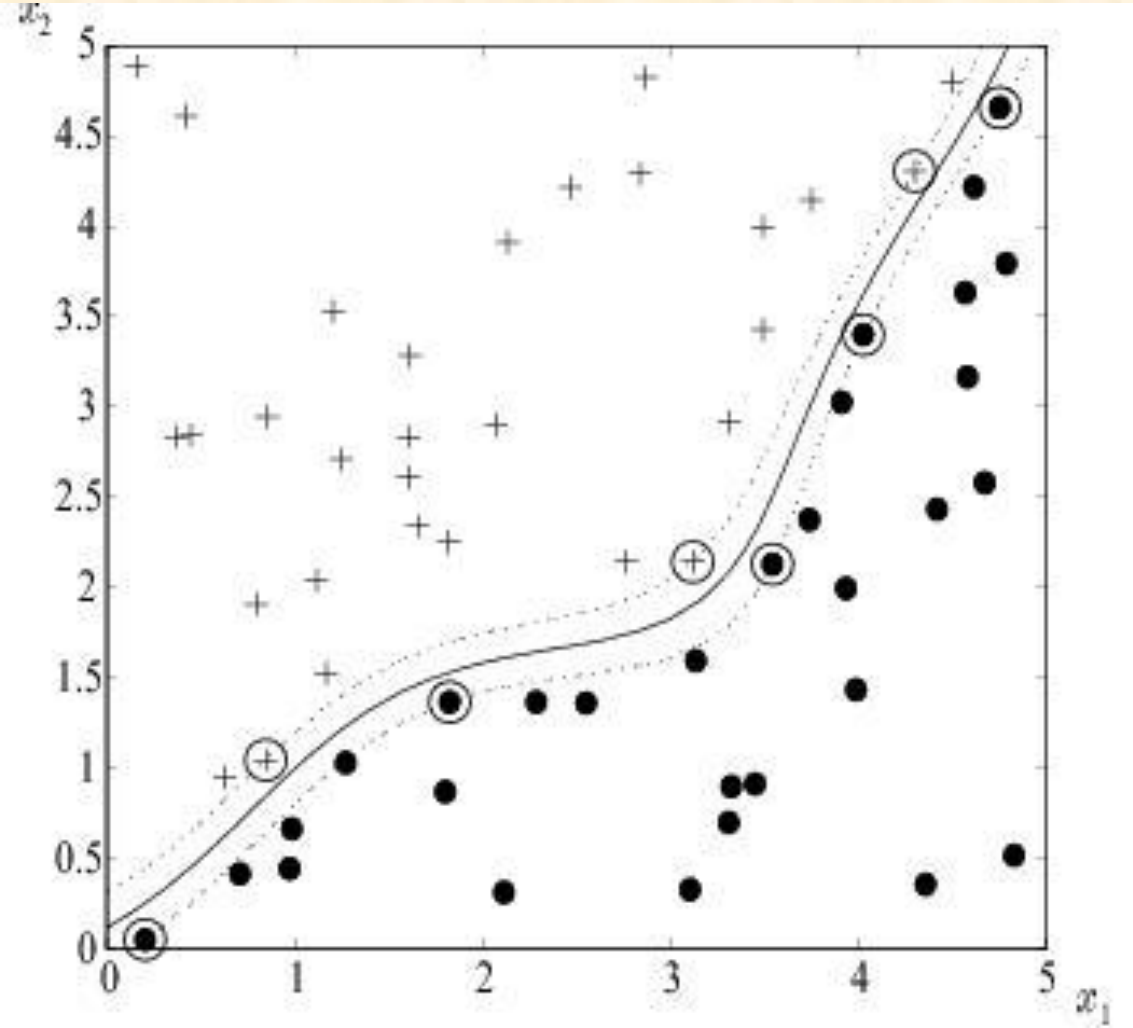


ΣΗΜ.: Οι μη γραμμικές SVM μπορεί να θεωρηθούν ως γενικευμένοι γραμμικοί ταξινομητές.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΓΕΝΙΚΕΥΜΕΝΟΙ ΓΡΑΜ. ΤΑΞΙΝΟΜΗΤΕΣ

Μηχανές διανυσματικής στήριξης (SVM): Η μη γραμμική περίπτωση

Παράδειγμα



ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

Πρόκειται για μια οικογένεια μη γραμμικών ταξινομητών. Είναι συστήματα απόφασης πολλών σταδίων (multistage), όπου οι κλάσεις απορρίπτονται διαδοχικά (sequentially), έως ότου φτάσουμε σε μια κλάση που θα είναι τελικά αποδεκτή. Για το λόγο αυτό:

- Ο χώρος των χαρακτηριστικών τεμαχίζεται σε μοναδικές περιοχές με έναν ακολουθιακό τρόπο.

- Με την άφιξη ενός διανύσματος χαρακτηριστικών, λαμβάνονται διαδοχικές αποφάσεις καταχώρησης χαρακτηριστικών σε συγκεκριμένες περιοχές, ακολουθώντας ένα μονοπάτι κόμβων (nodes) ενός κατάλληλα κατασκευασμένου δένδρου.

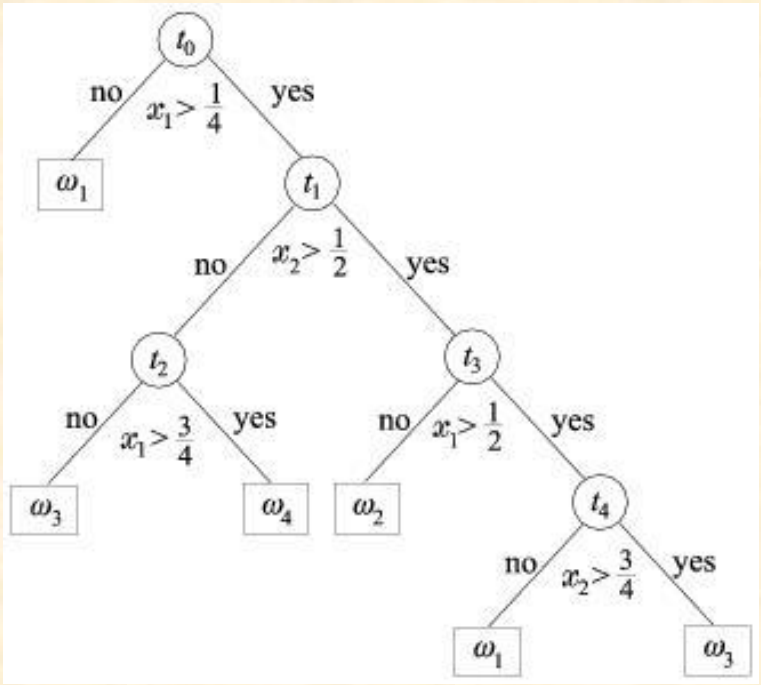
- Η ακολουθία των αποφάσεων εφαρμόζεται (συνήθως) σε μεμονωμένα χαρακτηριστικά και οι ερωτήσεις που εξετάζονται σε κάθε κόμβο είναι του τύπου:

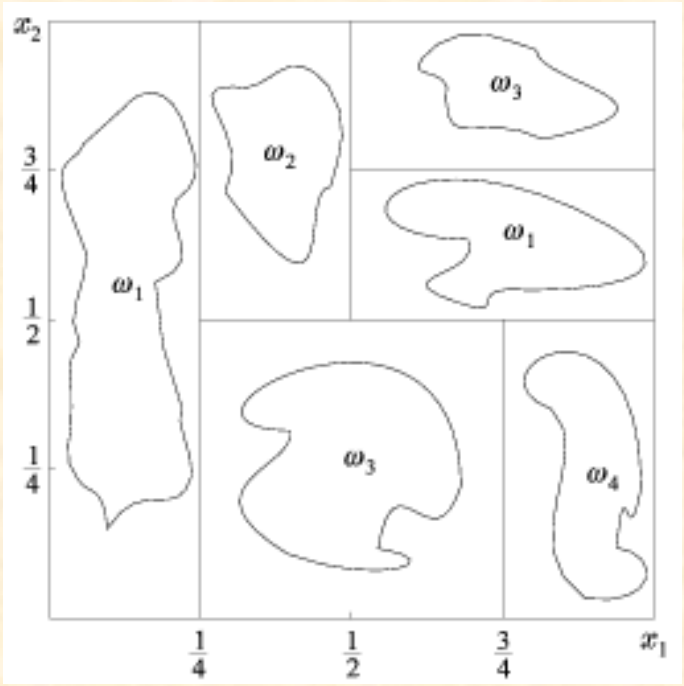
  είναι το χαρακτηριστικό $x_i \leq a$

όπου a είναι ένα προεπιλεγμένο κατώφλι (επιλέγεται κατά τη διάρκεια της εκπαίδευσης).

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

- Τα παρακάτω σχήματα αντιστοιχούν σε τέτοιο παράδειγμα. Τα δένδρα αυτού του τύπου είναι γνωστά ως Συνήθη Δυαδικά Δένδρα Ταξινόμησης (Ordinary Binary Classification Trees (OBCT)). Τα υπερπέπιπεδα απόφασης που διαιρούν το χώρο σε περιοχές, είναι παράλληλα στους άξονες του χώρου των δειγμάτων. Άλλοι τύποι διαίρεσης του χώροι είναι επίσης δυνατοί, παρότι είναι λιγότερο δημοφιλείς.





ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

- Σχεδιαστικά στοιχεία που ορίζουν ένα δένδρο απόφασης.
  - Κάθε κόμβος, *t*, αντιστοιχεί σε ένα υποσύνολο $X_t \subseteq X$, όπου *X* είναι το σύνολο εκπαίδευσης. Σε κάθε κόμβο, το αντίστοιχο σύνολο, X_t διαιρείται σε δύο (δυαδική διαίρεση) ξένα μεταξύ τους υποσύνολα-απογόνους $X_{t,Y}$ and $X_{t,N}$, ώστε

$$X_{t,Y} \cap X_{t,N} = \emptyset$$
$$X_{t,Y} \cup X_{t,N} = X_t$$

$X_{t,Y}$ είναι το υποσύνολο του X_t για το οποίο η απάντηση στην ερώτηση του κόμβου *t* είναι ΝΑΙ. $X_{t,N}$ είναι το υποσύνολο που αντιστοιχεί στο ΟΧΙ. Η διαίρεση αποφασίζεται με βάση την ερώτηση (query) που υιοθετείται.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

Προαπαιτούμενα για τη δημιουργία ενός δένδρου απόφασης

- **Κριτήριο διαμερισμού:** καθορίζει τη βέλτιστη δυνατή διαμέριση του X_t στα $X_{t,Y}$ και $X_{t,N}$.

- **Κριτήριο τερματισμού-διαμέρισης** (stop-splitting): ελέγχει την ανάπτυξη του δένδρου έως τους τερματικούς κόμβους (φύλλα – leafs).

- **Κανόνας καταχώρησης τερματικού κόμβου** σε κατηγορία.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

- Σύνολο ερωτήσεων: Στα δένδρα OBCT το σύνολο των ερωτήσεων είναι του τύπου

  είναι $x_i \leq a$;

  Η επιλογή του συγκεκριμένου χαρακτηριστικού x_i και της τιμής του κατωφλίου a, για κάθε κόμβο t, είναι το αποτέλεσμα αναζήτησης, κατά την εκπαίδευση, ανάμεσα στα χαρακτηριστικά και ένα σύνολο από δυνατές τιμές κατωφλίου. Ο τελικός συνδυασμός είναι αυτός που οδηγεί στη βέλτιστη τιμή ενός κατάλληλου κριτηρίου.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

- **Κριτήριο διαμερισμού:** Η κύρια ιδέα πίσω από τη διαίρεση σε κάθε κόμβο είναι τα υποσύνολα-απόγονοι $X_{t,Y}$ και $X_{t,N}$ που θα προκύψουν να παρουσιάζουν μεγαλύτερο βαθμό ομογενοποίησης ως προς τις κλάσεις, σε σχέση με αυτόν του X_t. Έτσι το κριτήριο που θα επιλεγεί θα πρέπει να είναι σε συμφωνία με αυτόν το στόχο. Ένα συχνά χρησιμοποιούμενο κριτήριο είναι ο βαθμός μη-καθαρότητας ενός κόμβου (node impurity):

$$I(t) = -\sum_{i=1}^{M} P(\omega_i \mid t) \log_2 P(\omega_t \mid t)$$

και

$$P(\omega_i \mid t) \approx \frac{N_t^i}{N_t}$$

όπου N_t^i είναι ο αριθμός των στοιχείων του συνόλου X_t, τα οποία ανήκουν στην κλάση ω_i. Η μείωση της μη-καθαρότητας ενός κόμβου (decrease in node impurity) ορίζεται ως:

$$\Delta I(t) = I(t) - \frac{N_{t,\mathrm{Y}}}{N_t} I(t_{\mathrm{Y}}) - \frac{N_{t,N}}{N_t} I(t_N)$$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

Κριτήριο διαμερισμού (συν.):

- Ο στόχος είναι να επιλεγούν σε κάθε κόμβο εκείνοι οι παράμετροι (**χαρακτηριστικό** και **κατώφλι**), που οδηγούν σε μία διαίρεση, η οποία παρουσιάζει τη μεγαλύτερη δυνατή μείωση της μη-καθαρότητας.

- Γιατί η μεγαλύτερη δυνατή μείωση; Παρατηρείστε ότι η μέγιστη τιμή για την *I(t)* λαμβάνεται όταν όλες οι κλάσεις είναι ισοπίθανες, δηλ. όταν το X_t παρουσιάζει τον ελάχιστο δυνατό βαθμό ομογενοποίησης.

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

Παράδειγμα: Έστω ένα παράδειγμα δύο κλάσεων με τα ακόλουθα πέντε σημεία: (1,10)-ω_1, (2,7) –ω_2, (3,6) -ω_1, (4,8) –ω_2, (5,9) –ω_2.

Για το σύνολο αυτό είναι: $P(\omega_1)=2/5=0.4$ και $P(\omega_2)=3/5=0.6$.

Άρα η εντροπία του είναι $I = -\left(P(\omega_1)\log_2 P(\omega_1) + P(\omega_2)\log_2 P(\omega_2) \right) = 0.9710$.

Υπολογισμός της μείωσης της εντροπίας για την 1η συνιστώσα και τιμή 1 ($x_1 \leq 1$):

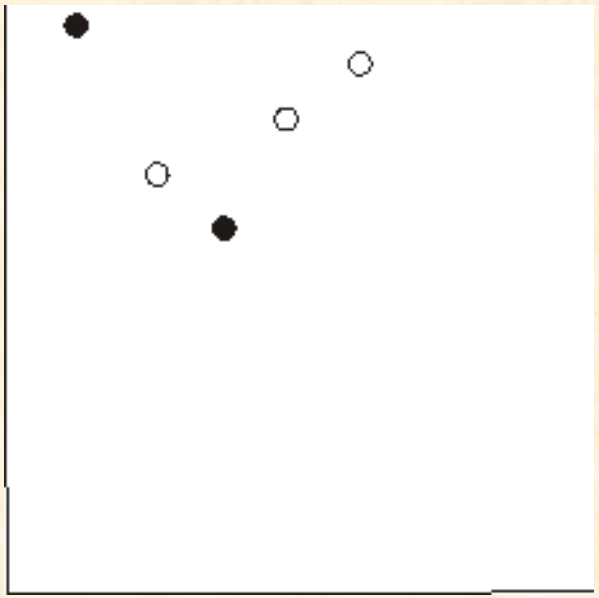
- $PY(\omega_1)=1/1=1$ και $PY(\omega_2)=0/1=0 \Rightarrow IY = -\left(PY(\omega_1)\log_2 PY(\omega_1) + PY(\omega_2)\log_2 PY(\omega_2)\right)= 0$.
- $PN(\omega_1)=1/4=0.25$ και $PN(\omega_2)=3/4=0.75$

  $\Rightarrow IN = -\left(PN(\omega_1)\log_2 PN(\omega_1) + PN(\omega_2)\log_2 PN(\omega_2)\right) = 0.8113$
- $\Delta I = I-(1/5)IY - (4/5)IN = 0.9710 - 0.2*0 - 0.8*0.8113 = 0.3219$.

Εργαζόμενοι ομοίως για τις υπόλοιπες περιπτώσεις έχουμε:

x_1	1	2	3	4	5
ΔI	0.32	0.02	**0.42**	0.17	0

x_2	6	7	8	9	10
ΔI	0	0.02	0.32	0.02	0.32



Η μέγιστη μείωση στην εντροπία επιτυγχάνεται για το 1ο Χαρακτηριστικό και τιμή κατωφλίου 3.

Άρα ο κανόνας για τον πρώτο κόμβο είναι: $x_1 \leq 3$

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

- **Κανόνας τερματισμού διαμέρισης**. Υιοθέτησε ένα κατώφλι *T* και σταμάτα την περαιτέρω διαίρεση ενός κόμβου (δηλ. καταχώρησέ τον σαν φύλλο-τερματικό κόμβο), αν η μείωση της μη-καθαρότητας είναι μικρότερη από *T*. Δηλ. όταν ο κόμβος *t* είναι "αρκετά καθαρός".

- **Κανόνας αντιστοίχισης σε κλάση:** Καταχώρησε ένα φύλλο στην κλάση ω_j , για την οποία: $$j = \arg\max_i P(\omega_i \mid t)$$

➢ Summary of an OBCT algorithmic scheme:

- Begin with the root node, i.e., $X_t = X$
- For each new node t
  - For every feature $x_k, k = 1, 2, \ldots, l$
    - For every value $\alpha_{kn}, n = 1, 2, \ldots, N_{tk}$
      - Generate X_{tY} and X_{tN} according to the answer in the question: is $x_k(i) \leq \alpha_{kn},\ i = 1, 2, \ldots, N_t$
      - Compute the impurity decrease
    - End
    - Choose α_{kn_0} leading to the maximum decrease w.r. to x_k
  - End
  - Choose x_{k_0} and associated $\alpha_{k_0 n_0}$ leading to the overall maximum decrease of impurity
  - If stop-splitting rule is met declare node t as a leaf and designate it with a class label
  - If not, generate two descendant nodes t_Y and t_N with associated subsets X_{tY} and X_{tN}, depending on the answer to the question: is $x_{k_0} \leq \alpha_{k_0 n_0}$
- End

ΜΗ ΓΡΑΜΜΙΚΟΙ ΤΑΞΙΝΟΜΗΤΕΣ – ΔΕΝΔΡΑ ΑΠΟΦΑΣΗΣ

Παρατηρήσεις:

- Ένας κρίσιμος παράγοντας κατά τη φάση σχεδιασμού είναι το μέγεθος του δένδρου. Συνήθως το δένδρο αναπτύσσεται έως ότου φτάσει σε μεγάλο μέγεθος και στη συνέχεια εφαρμόζονται διάφορες τεχνικές κλαδέματος (pruning).
- Τα δένδρα απόφασης ανήκουν στην κατηγορία των ασταθών (unstable) ταξινομητών. Αυτό μπορεί να αντιμετωπιστεί με τεχνικές «μέσου όρου» ("averaging" techniques). Π.χ. χρησιμοποιώντας τεχνικές bootstrapping στο X, κατασκευάζονται διάφορα δένδρα, T_i, $i=1, 2, \ldots, B$. Η απόφαση ταξινόμησης λαμβάνεται με βάση έναν κανόνα πλειοψηφίας (majority voting).

Example 1: A Simple Tree

Consider the following $n = 16$ points in two dimensions for training a binary CART tree ($B = 2$) using the entropy impurity (Eq. 1).

$$i(N) = -\sum_j P(\omega_j) \log_2 P(\omega_j), \qquad (1)$$

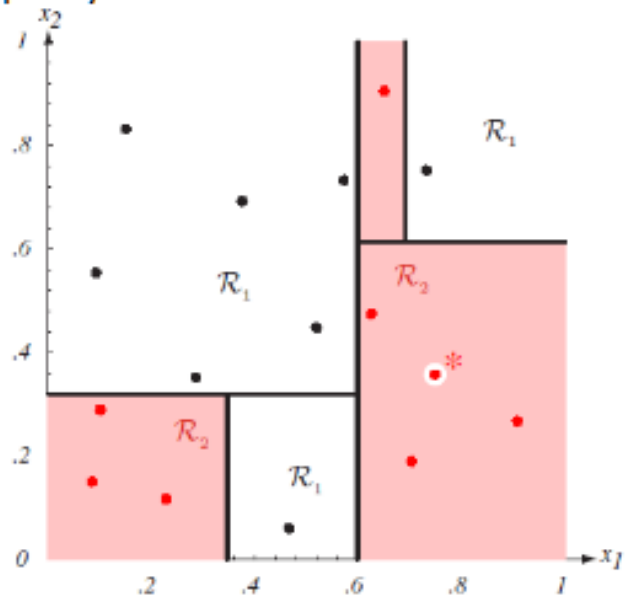
ω_1 (black)	
x_1	x_2
.15	.83
.09	.55
.29	.35
.38	.70
.52	.48
.57	.73
.73	.75
.47	.06

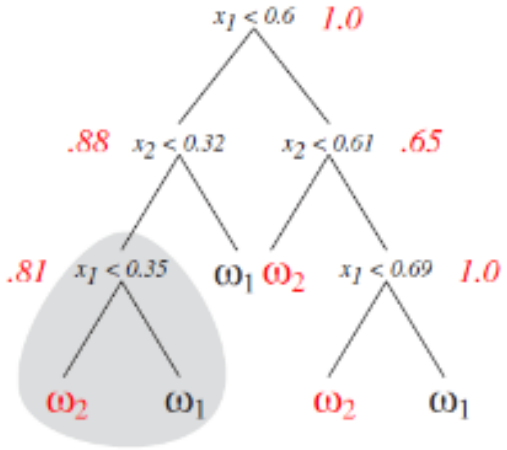
ω_2 (red)	
x_1	x_2
.10	.29
.08	.15
.23	.16
.70	.19
.62	.47
.91	.27
.65	.90
.75	.36* (.32†)

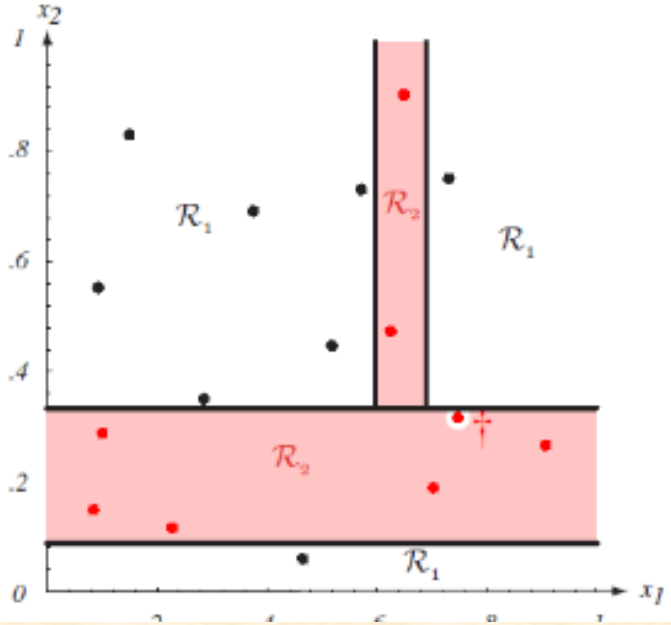
Example taken from http://www.cse.msu.edu/~cse802/DecisionTrees.pdf

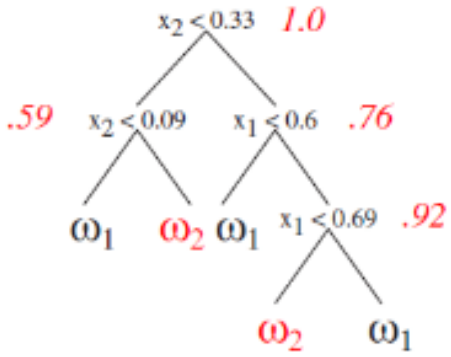
Example 1. Simple Tree

Entropy impurity at nonterminal nodes is shown in red and impurity at each leaf node is 0









Instability or sensitivity of tree to training points; alteration of a single point leads to a very different tree; due to discrete & greedy nature of CART

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

Η βασική φιλοσοφία πίσω από το συνδυασμό διαφορετικών ταξινομητών βασίζεται στο γεγονός ότι ακόμα και ο «καλύτερος» ταξινομητής αποτυγχάνει σε μερικά διανύσματα όπου άλλοι ταξινομητές μπορεί να δώσουν σωστή ταξινόμηση. Ο συνδυασμός ταξινομητών σκοπεύει στην εκμετάλλευση αυτής της συμπληρωματικής πληροφορίας (complementary information) που δίνεται από διάφορους ταξινομητές.

Έτσι κάποιος σχεδιάζει διαφορετικούς βέλτιστους ταξινομητές και στη συνέχεια συνδυάζει τα αποτελέσματα με βάση ένα συγκεκριμένο κανόνα.

- Έστω ότι καθένας από τους, *L* ταξινομητές που σχεδιάστηκαν δίνει στην έξοδό του τις εκ των υστέρων πιθανότητας

$$P(\omega_i \mid \underline{x}), i = 1, 2, ..., M$$

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

- Κανόνας γινομένου: Καταχώρησε το x στην κλάση ω_i :

$$i = \arg\max_k \prod_{j=1}^{L} P_j\left(\omega_k \mid \underline{x}\right)$$

όπου $P_j\left(\omega_k \mid \underline{x}\right)$ είναι η αντίστοιχη εκ των υστέρων πιθανότητα του j^{th} ταξινομητή.

- Κανόνας άθροισης: Καταχώρησε το x στην κλάσηω_i :

$$i = \arg\max_k \sum_{j=1}^{L} P_j\left(\omega_k \mid \underline{x}\right)$$

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

Παράδειγμα: Έστω ένα πρόβλημα **3** κλάσεων, ω_1, ω_2, ω_3, με $L=4$ ταξινομητές. Σχηματικά η ταξινόμηση ενός διανύσματος $\boldsymbol{x}$ σε μια από τις τρεις κλάσεις γίνεται ως εξής:

$\boldsymbol{x}$ →

	ω_1	ω_2	ω_3
Ταξ. 1	$P_1(\omega_1\|\boldsymbol{x})$	$P_1(\omega_2\|\boldsymbol{x})$	$P_1(\omega_3\|\boldsymbol{x})$
Ταξ. 2	$P_2(\omega_1\|\boldsymbol{x})$	$P_2(\omega_2\|\boldsymbol{x})$	$P_2(\omega_3\|\boldsymbol{x})$
Ταξ. 3	$P_3(\omega_1\|\boldsymbol{x})$	$P_3(\omega_2\|\boldsymbol{x})$	$P_3(\omega_3\|\boldsymbol{x})$
Ταξ.4	$P_4(\omega_1\|\boldsymbol{x})$	$P_4(\omega_2\|\boldsymbol{x})$	$P_4(\omega_3\|\boldsymbol{x})$
Γιν.	$\boldsymbol{\Pi_j P_j(\omega_1\|x)}$	$\boldsymbol{\Pi_j P_j(\omega_2\|x)}$	$\boldsymbol{\Pi_j P_j(\omega_3\|x)}$
Άθρ.	$\boldsymbol{\Sigma_j P_j(\omega_1\|x)}$	$\boldsymbol{\Sigma_j P_j(\omega_2\|x)}$	$\boldsymbol{\Sigma_j P_j(\omega_3\|x)}$

→ $\boldsymbol{i}$: $argmax_{k=1,2,3}\, \boldsymbol{\Pi_j P_j(\omega_k|x)}$

→ $\boldsymbol{i}$: $argmax_{k=1,2,3}\, \boldsymbol{\Sigma_j P_j(\omega_k|x)}$

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

- Κανόνας πλειοψηφίας: Καταχώρησε το x στην κλάση για την οποία υπάρχει ομοφωνία ή όταν τουλάχιστον ℓ_c από τους ταξινομητές συμφωνούν στην κλάση

$$\ell_c = \begin{cases} \frac{L}{2}+1, \text{ L even} \\ \frac{L+1}{2}, \text{ L odd} \end{cases}$$

Διαφορετικά η απόφαση είναι απόρριψη (rejection), δηλ. δεν λαμβάνεται καμία απόφαση.

Έτσι σωστή απόφαση λαμβάνεται όταν η πλειοψηφία των ταξινομητών συμφωνεί με τη σωστή κατηγορία και λάθος όταν η πλειοψηφία συμφωνεί με μία λάθος κατηγορία.

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

Εξαρτημένοι ή ανεξάρτητοι ταξινομητές;

- Παρότι δεν υπάρχουν γενικά θεωρητικά αποτελέσματα, το πείραμα έδειξε ότι όσο πιο ανεξάρτητοι είναι οι ταξινομητές ως προς τις αποφάσεις τους, τόσο πιο πολύ αναμένεται η λήψη βελτιωμένων αποτελεσμάτων μετά το συνδυασμό των επιμέρους αποφάσεων. Ωστόσο, δεν υπάρχει εγγύηση ότι ο συνδυασμός ταξινομητών οδηγεί σε καλύτερη απόδοση συγκρινόμενος με την απόδοση του "καλύτερου" ανάμεσα στους ταξινομητές.

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

Προς την ανεξαρτησία: Δυνατά σενάρια

- Εκπαίδευση μεμονωμένων ταξινομητών χρησιμοποιώντας διαφορετικά διανύσματα δεδομένων. Εδώ κάποιος έχει αρκετές επιλογές:
  - Bootstrapping: Πρόκειται για μία δημοφιλή τεχνική για το συνδυασμό ασταθών ταξινομητών, όπως είναι τα δένδρα απόφασης.
  - Stacking: Εκπαίδευση του συνδυαστή με δεδομένα που δεν έχουν χρησιμοποιηθεί για την εκπαίδευση των μεμονωμένων ταξινομητών.
  - Χρήση διαφορετικών υποχώρων για την εκπαίδευση μεμονωμένων ταξινομητών: Σύμφωνα μ' αυτή τη μέθοδο, κάθε μεμονωμένος ταξινομητής εκπαιδεύεται σε διαφορετικό υπόχωρο του χώρου των χαρακτηριστικών. Δηλ. χρησιμοποιούνται διαφορετικά χαρακτηριστικά για κάθε ταξινομητή.

ΣΥΝΔΥΑΖΟΝΤΑΣ ΤΑΞΙΝΟΜΗΤΕΣ

Παρατηρήσεις:

- Οι κανόνες πλειοψηφίας και αθροίσματος συγκαταλέγονται ανάμεσα στα πιο δημοφιλή συνδυαστικά σχήματα.

- Η εκπαίδευση των μεμονωμένων ταξινομητών σε διαφορετικούς υποχώρους του χώρου των χαρακτηριστικών φαίνεται να οδηγεί σε σημαντικά καλύτερα αποτελέσματα σε σχέση με την περίπτωση όπου οι ταξινομητές εκπαιδεύονται στον ίδιο υπόχωρο.

- Εκτός από τους τρεις παραπάνω κανόνες, μπορούν να υιοθετηθούν και άλλοι όπως η τιμή της διαμέσου (Median value) των εξόδων των μεμονωμένων ταξινομητών.

- Boosting προσέγγιση: Ένας «ισχυρός» ταξινομητής δημιουργείται με τη διαδοχική προσθήκη «ασθενών» (weak) ταξινομητών, οι οποίοι εκπαιδεύονται δίνοντας έμφαση στα σημεία που απέτυχε να ταξινομήσει σωστά ο συνδυασμός των προηγούμενων ασθενών ταξινομητών.

ΑΞΙΟΛΟΓΗΣΗ ΤΑΞΙΝΟΜΗΤΗ

Ο στόχος είναι η εκτίμηση της πιθανότητας λάθους ενός συστήματος ταξινόμησης

Τεχνική μέτρησης λαθών (Error Counting Technique)

- Έστω πρόβλημα ταξινόμησης σε M κλάσεις.
- Έστω N_i τα σημεία της κλάσης ω_i που χρησιμοποιούνται για δοκιμή.

$$\sum_{i=1}^{M} N_i = N$$

- Έστω P_i η πιθανότητα λάθους για την κλάση ω_i
- Υποθέτουμε ότι ο ταξινομητής έχει σχεδιαστεί χρησιμοποιώντας ένα διαφορετικό **ανεξάρτητο** σύνολο δεδομένων.
- Υποθέτοντας ότι τα διανύσματα του συνόλου δοκιμής είναι ανεξάρτητα, η πιθανότητα να έχουμε k_i διανύσματα από την ω_i λανθασμένα ταξινομημένα είναι

$$\text{prob}\{k_i \text{ in } \omega_i \text{ wrongly classified}\} = \begin{pmatrix} N_i \\ k_i \end{pmatrix} P_i^{k_i} (1-P_i)^{N_i - k_i}$$

ΑΞΙΟΛΟΓΗΣΗ ΤΑΞΙΝΟΜΗΤΗ

- Αφού τα P_i's είναι άγνωστα, εκτιμούνται μέσω της μεγιστοποίησης της παραπάνω διωνυμικής κατανομής και προκύπτει ότι

$$\hat{P}_i = \frac{k_i}{N_i}$$

- Δηλ. είναι ο αριθμός των λανθασμένα ταξινομημένων διανυσμάτων προς τον αριθμό των διανυσμάτων δοκιμής της εν λόγω κλάσης.
- Συνολική πιθανότητα λάθους

$$\hat{P} = \sum_{i=1}^{M} P(\omega_i) \frac{k_i}{N_i}$$

- ***Στατιστικές ιδιότητες***

  - $E[k_i] = N_i P_i$
  - $\text{Ετσι}, E[\hat{p}] = \sum_{i=1}^{M} P(\omega_i) P_i = P$
  - $\sigma_{k_i}^2 = N_i (1 - P_i) P_i$
  - $\sigma_{\hat{p}}^2 = \sum_{i=1}^{M} P^2(\omega_i) \frac{P_i (1 - P_i)}{N_i}$

  Ο εκτιμητής είναι αμερόληπτος αλλά ασυμπτωτικά συνεπής. Έτσι για **μικρές τιμές του *N*, μπορεί να μην είναι αξιόπιστος**

- Μια εκτίμηση που έχει προκύψει από θεωρητική ανάλυση σχετικά με τον ικανό αριθμό διανυσμάτων *N* του συνόλου δοκιμής, προκειμένου η πιθανότητα να είναι γύρω από δεδομένη τιμή *P* είναι

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

$\text{Ετσι}, \gamma\iota\alpha\ P \approx 0.01, N \approx 10000.\ \Gamma\iota\alpha\ P \approx 0.03, N \approx 3000$

ΑΞΙΟΛΟΓΗΣΗ ΤΑΞΙΝΟΜΗΤΗ

Εκμεταλλευόμενοι το πεπερασμένου μεγέθους σύνολο δοκιμής.

- Μέθοδος επανατοποθέτησης (Resubstitution method):

  Χρησιμοποίησε τα ίδια δεδομένα για εκπαίδευση και δοκιμή. Έχουμε υποεκτίμηση του σφάλματος. Η εκτίμηση βελτιώνεται για μεγάλες τιμές του N και μεγάλες τιμές του λόγου N/l.

- Holdout Method: Δοθέντος συνόλου N διανυσμάτων χώρισέ τα σε:

  N_1: διανύσματα εκπαίδευσης

  N_2: διανύσματα δοκιμής

  $N=N_1+N_2$

  - **Πρόβλημα:** Λιγότερα δεδομένα τόσο για εκπαίδευση, όσο και γα δοκιμή.

ΑΞΙΟΛΟΓΗΣΗ ΤΑΞΙΝΟΜΗΤΗ

- Leave-one-out Method

Τα βήματα:

- Επέλεξε ένα από τα *N* δείγματα. Εκπαίδευσε τον ταξινομητή χρησιμοποιώντας τα υπόλοιπα *N*-1 δείγματα. Έλεγξε την απόδοση του ταξινομητή χρησιμοποιώντας το επιλεγμένο δείγμα. Αν είναι λανθασμένα ταξινομημένο μέτρα ένα επιπλέον λάθος.
- Επανέλαβε το παραπάνω βήμα εξαιρώντας ένα διαφορετικό δείγμα κάθε φορά.
- Υπολόγισε την πιθανότητα λάθους παίρνοντας το μέσο όρο των καταμετρημένων λαθών.

- **Πλεονεκτήματα:**

- Χρήση όλων των διαθέσιμων δεδομένων για εκπαίδευση και δοκιμή
- Διασφάλιση ανεξαρτησίας ανάμεσα στα διανύσματα εκπαίδευσης και δοκιμής.

- **Μειονεκτήματα:**

- Complexity in computations high

- Παραλλαγές της μεθόδου εξαιρούν $k>1$ σημεία κάθε φορά, προκειμένου να μειώσουν την υπολογιστική πολυπλοκότητα.