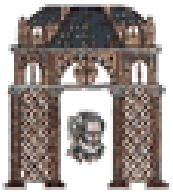
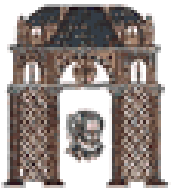


Αναγνώριση Προτύπων

Γραμμικές Συναρτήσεις Διάκρισης
(Linear Discriminant Functions)

Χριστόδουλος Χαμζάς

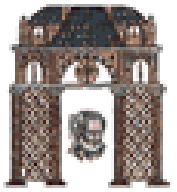
Τα περιεχόμενα των παρουσιάσεων προέρχονται από τις παρουσιάσεις του αντίστοιχου διδακτέου μαθήματος του καθ. Παναγιώτη Τσακαλίδη, Τμ. Επιστήμης Υπολογιστών, Παν. Κρήτης και του καθ. Σέργιου Θεοδωρίδη, Τμήμα Πληροφορικής, Πανεπιστήμιο Αθηνών. Βασίζεται στα βιβλία: "Pattern Classification", R.O. Duda, P.E. Hart, D.G. Stork, Wiley, 2nd Ed., 2001 και S. Theodoridis, K. Koutroumbas, Pattern Recognition, 3rd Edition, Academic Press, 2006



Γραμμικές Συναρτήσεις Διάκρισης

ΠΟΡΕΙΑ ΤΟΥ ΜΑΘΗΜΑΤΟΣ

1. Γνωρίζαμε τις κατανομές και βρίσκαμε τις **συναρτήσεις διάκρισης** (Εκτιμητές Bayesian)
2. Υποθέτουμε ότι γνωρίζουμε την μορφή της κατανομής πιθανότητας και από τα δεδομένα εκπαίδευσης βρίσκουμε τις τιμές των παραμέτρων τους (Parametric Estimation)
3. Από τα δεδομένα εκπαίδευσης βρίσκουμε (εκτιμούμε) τις κατανομές πιθανότητας (Παράθυρα Parzen και KNN)
4. Υποθέτουμε ότι γνωρίζουμε την μορφή των **συναρτήσεων διάκρισης** και θέλουμε να βρούμε τις τιμές των παραμέτρων τους από τα δεδομένα εκπαίδευσης. Δεν χρειαζόμαστε τις υποκείμενες κατανομές (non parametric estimation). Κίνδυνος no-robust estimators



Γραμμικές Συναρτήσεις Διάκρισης

- Στόχος: Η σχεδίαση γραμμικών ως προς το διάνυσμα χαρακτηριστικών $\boldsymbol{x}$ συναρτήσεων διάκρισης που ορίζουν υπερεπίπεδα ως επιφάνειες απόφασης.
- Γιατί; Απλή μορφή, εύκολη υλοποίηση, βέλτιστες για Γκαουσσιανές σ.π.π.
- Πώς: Διατυπώνοντας το πρόβλημα εύρεσης των παραμέτρων (βαρών) ως πρόβλημα βελτιστοποίησης μιας συνάρτησης κριτηρίου (κόστους).
- Τι είναι η συνάρτηση κριτηρίου; Μια βαθμωτή συνάρτηση των βαρών που θα πρέπει να ελαχιστοποιηθεί, π.χ. η πιθανότητα λάθος ταξινόμησης κατά την εκπαίδευση.
- Είναι δύσκολο να επιτευχθεί; Ναί, γενικώς είναι δύσκολη η σχεδίαση ενός γραμμικού ταξινομητή που να ελαχιστοποιεί το ρίσκο.
- Επομένως; Χρησιμοποιούμε εναλλακτικά κριτήρια (απλές συναρτήσεις των βαρών) και επαναληπτικές μεθόδους βελτιστοποίησης (καθόδου κατά την κλίση του κριτηρίου – gradient descent).

Γιατί γπαμμικοί ταξινομητέρ;

- Πξόβιεκα 2 θαηεγνξηώλ: Αλ ν αξηζκόο N ηωλ πξνηύπωλ είλαη κηθξόηεξνο από ηνλ αξηζκό D ηωλ ζπληζηωζώλ θάζε πξνηύπνπ, ηόηε ππάξρεη πάληα ππεξεπίπεδν πνπ ηα δηαρωξίδεη

- Επνκέλωο νη γξακκηθνί ηαμηλνκεηέο είλαη ρξήζηκνη
  - ζε πξνβιήκαηα πνιύ κεγάιεο δηαζηαηηθόηεηαο
  - Σε πξνβιήκαηα κέηξηαο δηαζηαηηθόηεηαο, όπνπ ππάξρεη έλαο ζρεηηθά κηθξόο αξηζκόο πξνηύπωλ εθπαίδεπζεο

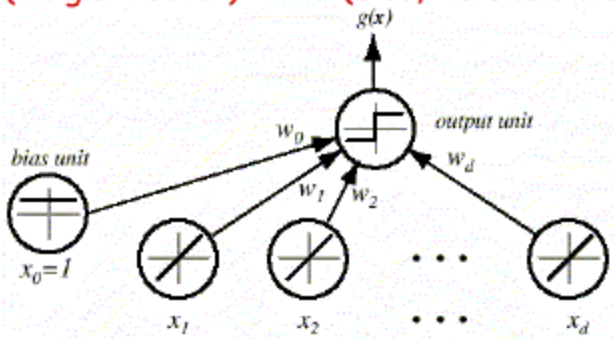
- Επηπιένλ ν αξηζκόο ηωλ ζπληζηωζώλ ελόο πξνηύπνπ κπνξεί λα απμεζεί απζαίξεηα κε ηελ πξνζζήθε λέωλ ζπληζηωζώλ πνπ είλαη κε γξακκηθέο ζπλαξηήζεηο ηωλ αξρηθώλ ζπληζηωζώλ (π.ρ. πνιπώλπκα)

Συναρτήσεις και Επιφάνειες Διάκρισης

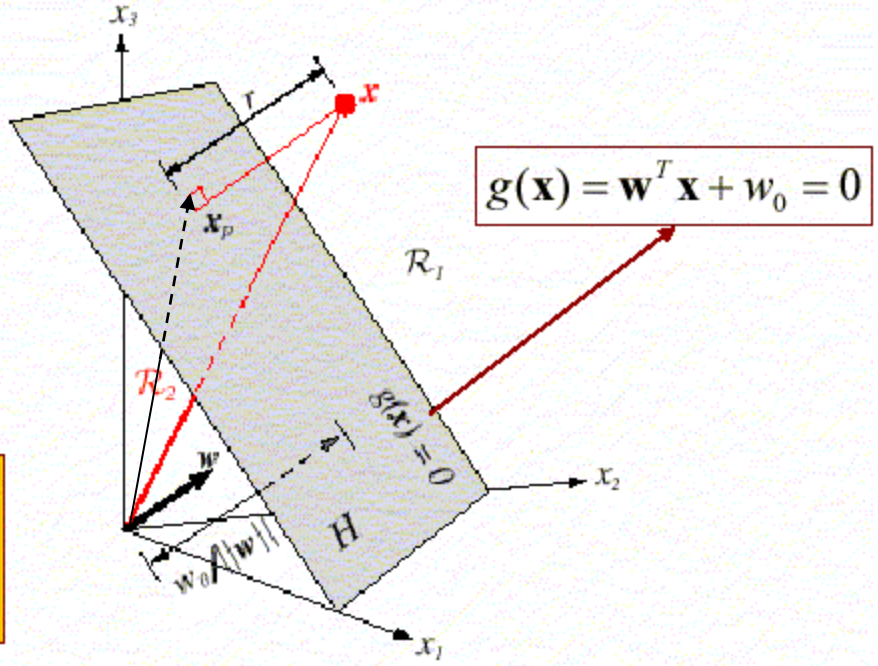
$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

Διάνυσμα βαρών (weight vector)

Βάρος κατωφλίου (bias, threhold weight)



Το διάνυσμα βαρών, **w**, καθορίζει τον προσανατολισμό του υπερεπιπέδου απόφασης και το βάρος κατωφλίου, w_0, καθορίζει τη σχετική θέση του ως προς την αρχή των αξόνων.



$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = 0$$

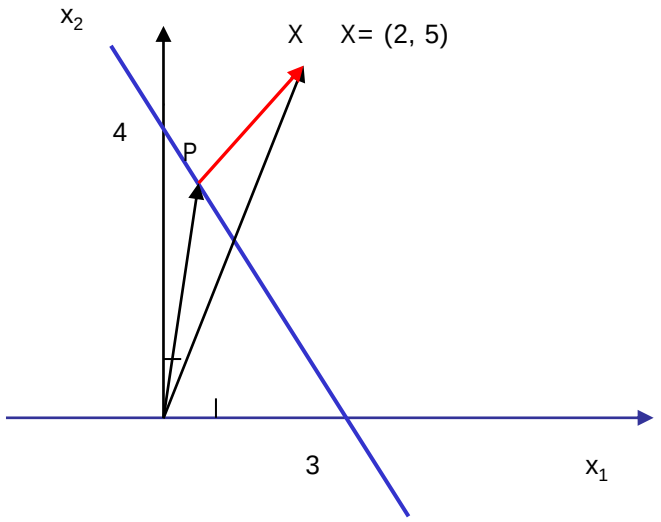
Έστω x_p η προβολή του x στο υπερεπίπεδο. Επειδή x_p είναι πάνω στο υπερεπίπεδο.

Άρα $g(x_p)=0=W^T x_p+w_0$ και

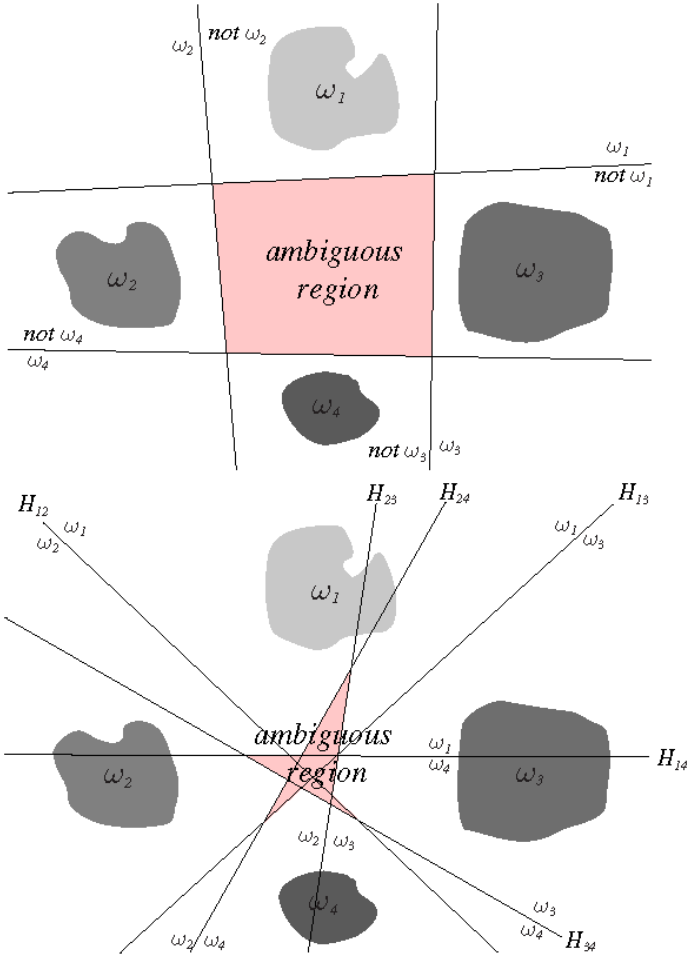
$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}, \text{ όπου } r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}$$

Παράδειγμα

- $g(x_1,x_2)=x_1+0.75x_2-3=[1 \;\; 0.75][x_1 \; x_2]^T+(-3)$
- $w=(1, .75)$, $w_0=-3$, $||w||=5/4$
- $X=(2, 5)$. Η απόσταση r του X από την ευθεία $g(X)=0$ είναι
- $r= (PX)=g(2, 5)/||w||=(11/4)/(5/4)=11/5$



Η Περίπτωση Πολλών Κλάσεων



Περίπτωση διαχωρισμού σε ω_1 / όχι ω_1

Μπορεί να υπάρχει περιοχή που δεν καθορίζεται.

Περίπτωση διαχωρισμού σε ω_i / ω_j

Μπορεί πάλι να υπάρχει περιοχή που δεν καθορίζεται.

Θα ακολουθήσουμε την φιλοσοφία των επιφανειών διαχωρισμού $g_i(x)=w^T x_i + w_{i0}$ και $x \in \omega_i$ εάν $g_i(x) > g_j(x)$ για κάθε $j \neq i$

Η Περίπτωση Πολλών Κλάσεων

Γραμμική Μηχανή (linear machine): $\boldsymbol{x}$ in ω_i αν $g_i(\boldsymbol{x})>g_j(\boldsymbol{x})$

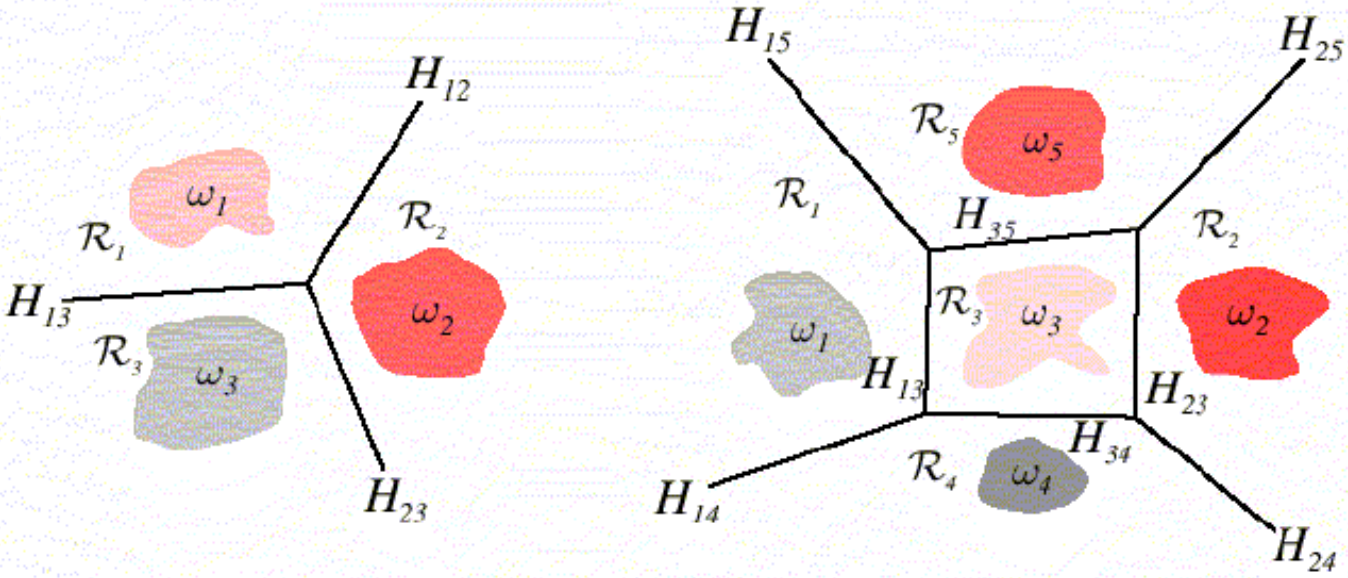
Σύνορα Απόφασης: H_{ij}: $g_i(\boldsymbol{x})=g_j(\boldsymbol{x}) \rightarrow (\boldsymbol{w}_i-\boldsymbol{w}_j)^t\boldsymbol{x} + (w_{i0}-w_{j0}) = 0$

- τμήμα υπερεπιπέδου κάθετο στο διάνυσμα $\boldsymbol{w}_i-\boldsymbol{w}_j$.

Απόσταση του $\boldsymbol{x}$ από το H_{ij}: $(g_i(\boldsymbol{x})-g_j(\boldsymbol{x})) / ||\boldsymbol{w}_i-\boldsymbol{w}_j||$

- σημαντικές είναι οι διαφορές των διανυσμάτων βαρών.

Κυρτές περιοχές απόφασης.



Γραμμικά Διαχωρίσιμες Κλάσεις: Διανύσματα και Περιοχές Λύσης

Επαυξημένα διανύσματα χαρακτηριστικών και βαρών (Augmented Vectors):

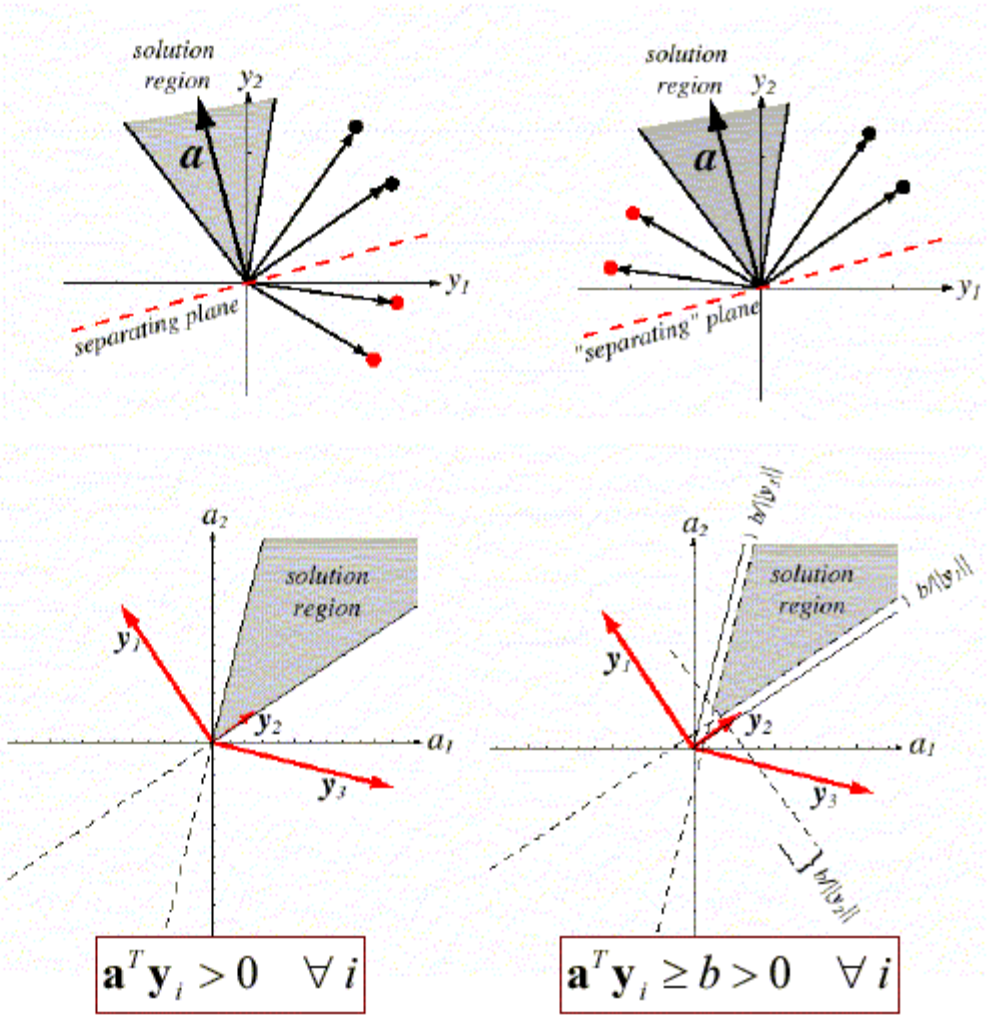
$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \\ x_d \end{bmatrix} = \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix}$$

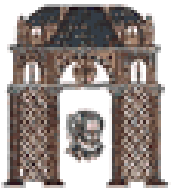
$$\mathrm{H}: \; g(\mathbf{y}) = \mathbf{a}^T\mathbf{y} = 0$$

$$\text{απόσταση } \mathbf{y} \text{ από H}: \; \left|\mathbf{a}^T\mathbf{y}\right| / \|\mathbf{a}\|$$

Κανονικοποίηση: αντικαθιστώντας όλα τα δείγματα εκπαίδευσης της κλάσης ω_2 με τα αρνητικά τους, ζητούμε τα διαχωριστικά διανύσματα (separating vectors) που πρέπει να ικανοποιούν την σχέση:

$$\mathbf{a}^T\mathbf{y}_i > 0 \quad \forall i$$





Διαδικασίες Βελτιστοποίησης

- Πρόβλημα: Εύρεση του $\boldsymbol{a}$ που ικανοποιεί το σύνολο των γραμμικών ανισοτήτων $\boldsymbol{a}^t\boldsymbol{y}_i > 0$ για κάθε $i=1,\ldots,n$.

- Πώς βρίσκουμε την κατάλληλη λύση;
  - Ορίζουμε μια συνάρτηση κριτηρίου, $J(\boldsymbol{a})$, και την ελαχιστοποιούμε ώστε το $\boldsymbol{a}$ να είναι ένα διάνυσμα λύσης.
  - Με αυτό τον τρόπο μετασχηματίζουμε το πρόβλημα της εξαντλητικής αναζήτησης σε πρόβλημα ελαχιστοποίησης μιας βαθμωτής συνάρτησης.

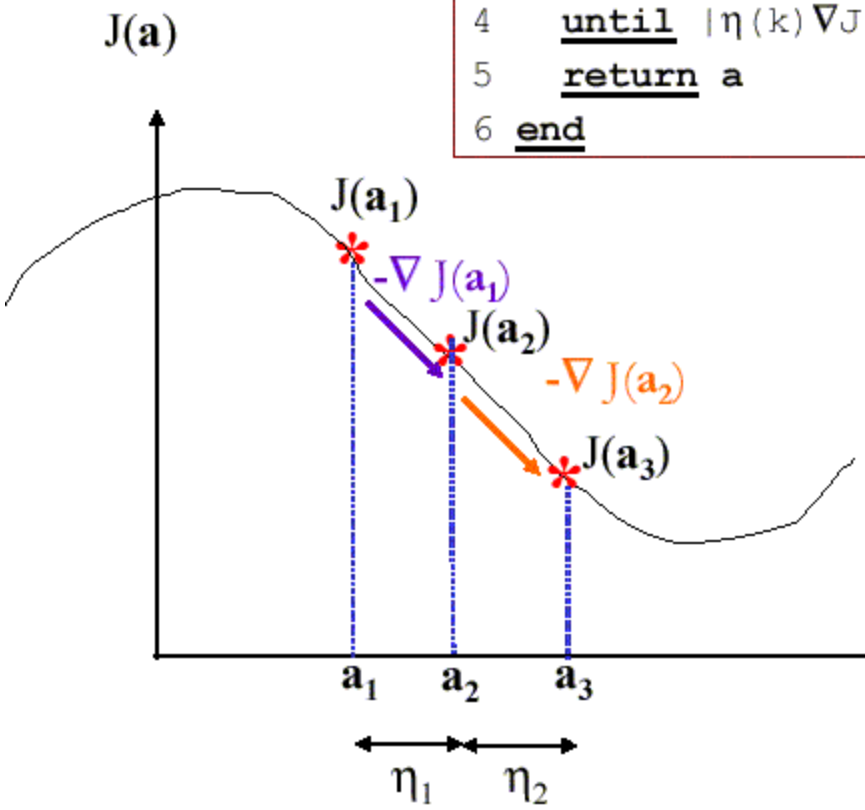
- Πώς ελαχιστοποιούμε την $J(\boldsymbol{a})$;
  - Επιλέγουμε κάποιο αρχικό σημείο $\boldsymbol{a}_1$ και υπολογίζουμε την τιμή $J(\boldsymbol{a}_1)$.
  - Υπολογίζουμε την κλίση στο $J(\boldsymbol{a}_1)$: $\nabla J(\boldsymbol{a}_1)$.
  - Παίρνουμε το επόμενο σημείο $\boldsymbol{a}_2$ κινούμενοι στην κατεύθυνση αρνητικής κλίσης (steepest descent), $-\nabla J(\boldsymbol{a}_1)$, κατά μια ποσότητα $\eta(k)$, τον λεγόμενο ρυθμό μάθησης (learning rate) ή το βήμα (stepsize).

Αλγόριθμος της πιο Απότομης Καθόδου (Steepest Descent)

Αλγόριθμος 1. Πιο Απότομη Κάθοδος (Steepest Descent)

```
1 begin initialize a, threshold θ, η(0)>0, k=0
2   do k ← k+1
3   a ← a-η(k)∇J(a)
4   until |η(k)∇J(a)|< θ
5   return a
6 end
```

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \eta_k \nabla J(\mathbf{a}_k)$$



Ζητήματα:

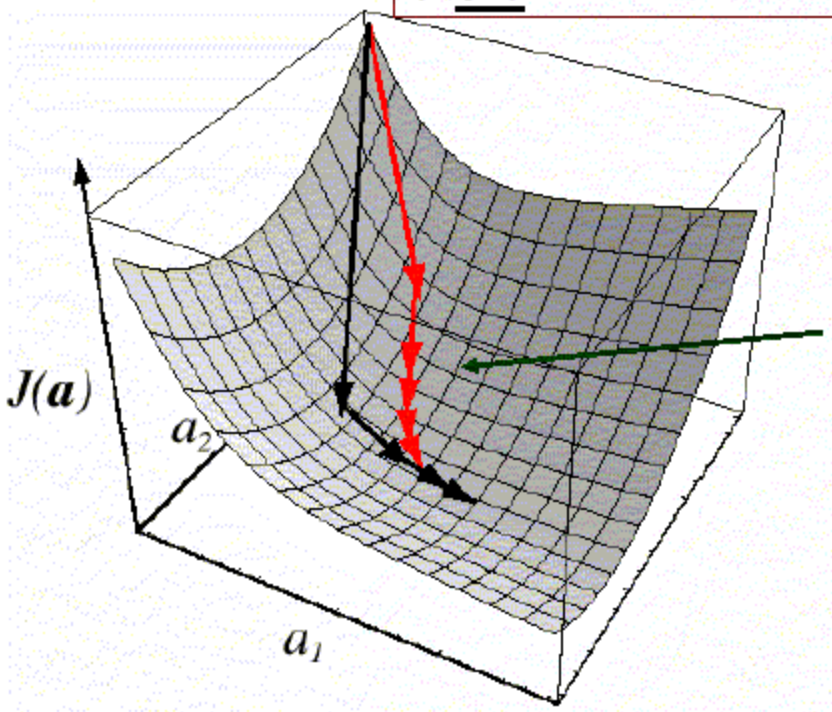
- **Πώς επιλέγουμε την συνάρτηση κριτηρίου;**
- **Πώς επιλέγουμε τον ρυθμό μάθησης *η(k)*;**
- **Σύγκλιση σε τοπικό/ολικό ελάχιστο;**
- **Πόσο γρήγορα συγκλίνουμε, πόσο ομαλά;**
- **Πότε σταματάμε;**

Αλγόριθμος Καθόδου Newton (Newton Descent)

Αλγόριθμος 2. Κάθοδος Newton (Newton Descent)

```
1 begin initialize a, threshold θ
2   a ← a-H^-1∇J(a)
3   until |H^-1∇J(a)| < θ
4   return a
5 end
```

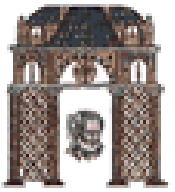
$$\mathbf{a}_{k+1} = \mathbf{a}_k - \mathbf{H}_k^{-1}\nabla J(\mathbf{a}_k)$$



$$\mathbf{H}_k = \left[\partial^2 J(\mathbf{a}) / \partial a_i \partial a_j\right]_{\mathbf{a}=\mathbf{a}_k}$$

Κόκκινο: Steepest Descent
Μαύρο: Newton Descent

Newton: μεγαλύτερη βελτίωση σε κάθε βήμα πληρώνοντας το υπολογιστικό κόστος της αντιστροφής του Hessian πίνακα *H*.



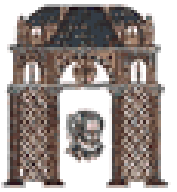
Το κριτήριο Perceptron

- Ποια μπορεί να είναι η συνάρτηση κριτηρίου;
  - Μία πρώτη επιλογή: Πλήθος των λάθος ταξινομημένων δειγμάτων εκπαίδευσης. Αλλά: Αυτή η συνάρτηση είναι ασυνεχής οπότε δεν είναι διαφόρισιμη.
  - Μια καλύτερη επιλογή: Η συνάρτηση κριτηρίου perceptron:

$$J_p(\mathbf{a}) = \sum_{\mathbf{y}\in\mathcal{Y}} \left(-\mathbf{a}^t\mathbf{y}\right)$$

όπου $\mathcal{Y}(\mathbf{a})$ είναι το σύνολο των δειγμάτων που δεν έχουν ταξινομηθεί σωστά από το $\mathbf{a}$.

  - Αν το $\mathcal{Y}(\mathbf{a})$ είναι κενό, τότε $J_p(\mathbf{a})=0$. Αφού $\mathbf{a}^t\mathbf{y}<0$ όταν το $\mathbf{y}$ δεν είναι σωστά ταξινομημένο, η $J_p(\mathbf{a})$ δεν είναι ποτέ αρνητική και μηδενίζεται όταν το $\mathbf{a}$ είναι διάνυσμα λύσης.
  - Γεωμετρικά, η $J_p(\mathbf{a})$ είναι ανάλογη του αθροίσματος των αποστάσεων των λάθος ταξινομημένων δειγμάτων από το σύνορο απόφασης.



Ο Αλγόριθμος Batch Perceptron

- Το διάνυσμα κλίσεων είναι: $$\nabla J_p(\mathbf{a}) = \left[\frac{\partial J_p}{\partial a_i}\right] = \sum_{y \in Y} -\mathbf{y}$$

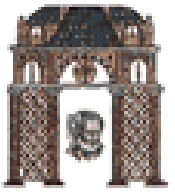
- Αναδρομική σχέση: $$\mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k) \sum_{y \in Y_k} \mathbf{y}$$

όπου Y_k είναι το σύνολο των δειγμάτων που έχουν ταξινομηθεί λάθος από το $\boldsymbol{a}_k$.

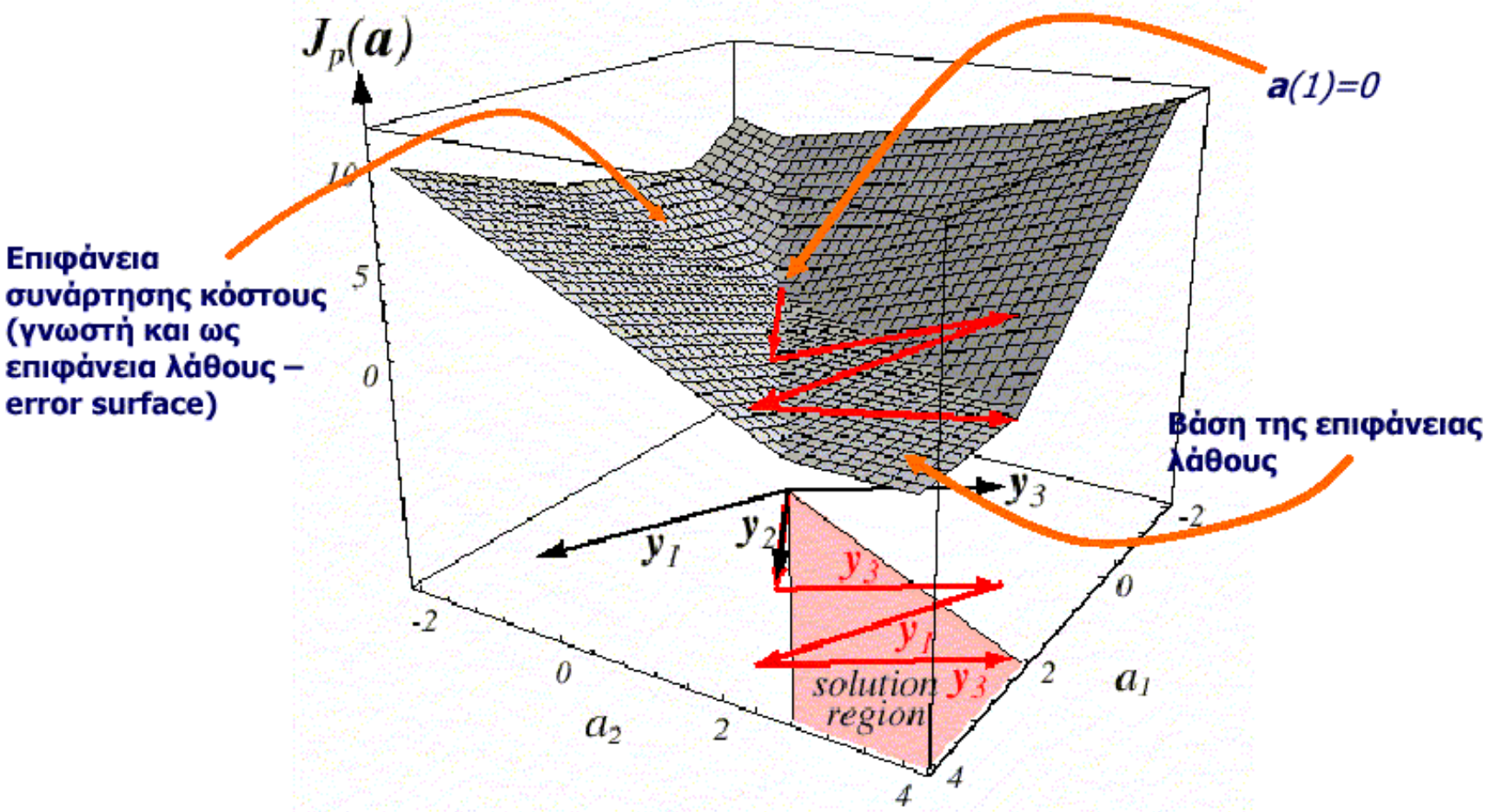
- Το επόμενο διάνυσμα βάρους (δ.β.) προκύπτει ως το άθροισμα του τρέχοντος δ.β. και ενός πολλαπλασίου του αθροίσματος των λάθος ταξινομημένων δειγμάτων.

Αλγόριθμος 3. Batch Perceptron

```
1 begin initialize a, κριτήριο θ, η(0)>0, k=0
2    do  k ← k+1
3        a ← a + η(k) Σ_{y∈Y_k} y
4    until |η(k) Σ_{y∈Y_k} y| < θ
5    return a
6 end
```



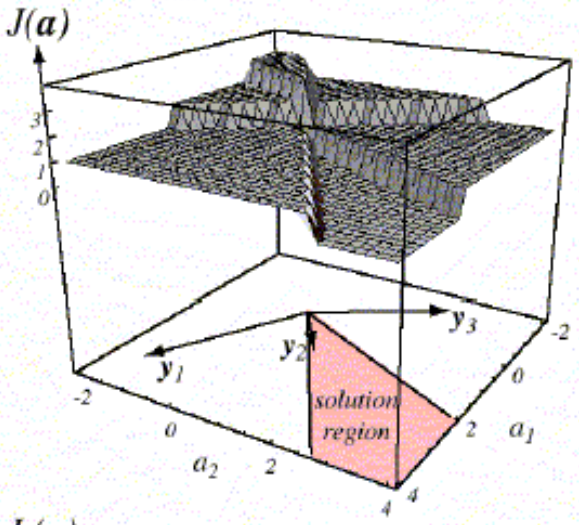
Βήματα του Batch Perceptron

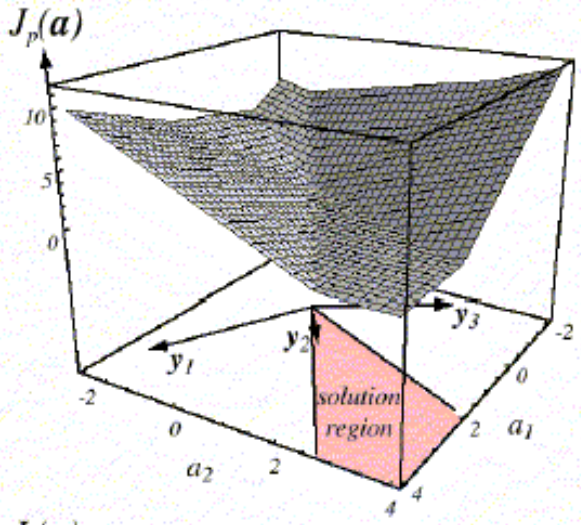


Τέσσερις Συναρτήσεις Κόστους

Πλήθος εσφαλμένων ταξινομήσεων

Bad





Κριτήριο Perceptron

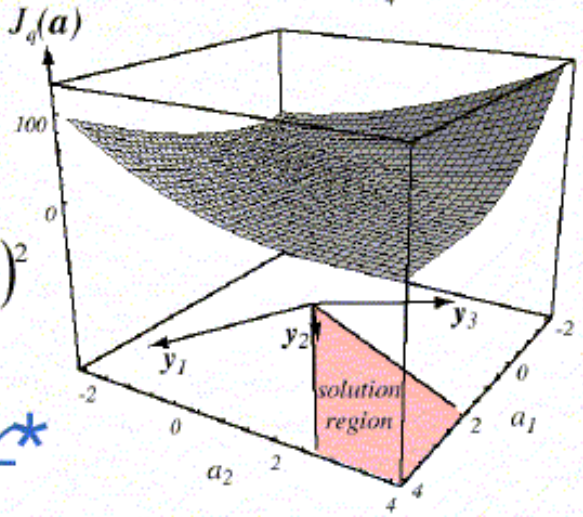
$$J_p(\mathbf{a}) = \sum_{\mathbf{y}\in\mathcal{Y}} \left(-\mathbf{a}^t\mathbf{y}\right)$$

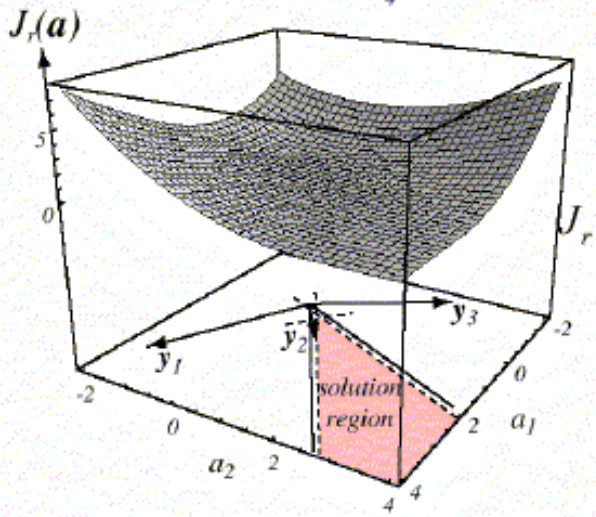
Good!

Συνολικό τετραγωνικό λάθος - Total square error (TSE)

$$J_q(\mathbf{a}) = \sum_{\mathbf{y}\in\mathcal{Y}} \left(\mathbf{a}^t\mathbf{y}\right)^2$$

Better*

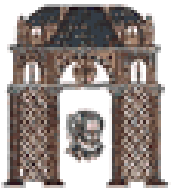




TSE with margin

$$J_r(\mathbf{a}) = \frac{1}{2}\sum_{\mathbf{y}\in\mathcal{Y}} \frac{\left(\mathbf{a}^t\mathbf{y}-b\right)^2}{\|\mathbf{y}\|}$$

Best*



Fixed Increment Single-Sample Perceptron

- Αντί να δοκιμάζουμε το διάνυσμα βαρών ***a(k)*** σε όλα τα δείγματα και να το διορθώνουμε βάσει του συνόλου Y_k των λάθος ταξινομημένων δειγμάτων, χρησιμοποιούμε τα δείγματα ένα κάθε φορά και ανάλογα με την ταξινόμηση του ανανεώνουμε ή όχι το διάνυσμα βαρών.
- Αν επιπλέον, χρησιμοποιήσουμε ένα σταθερό βήμα *η(k)*, τότε προκύπτει ο αλγόριθμος:

Αλγόριθμος 4. Fixed-Increment Single-Sample Perceptron

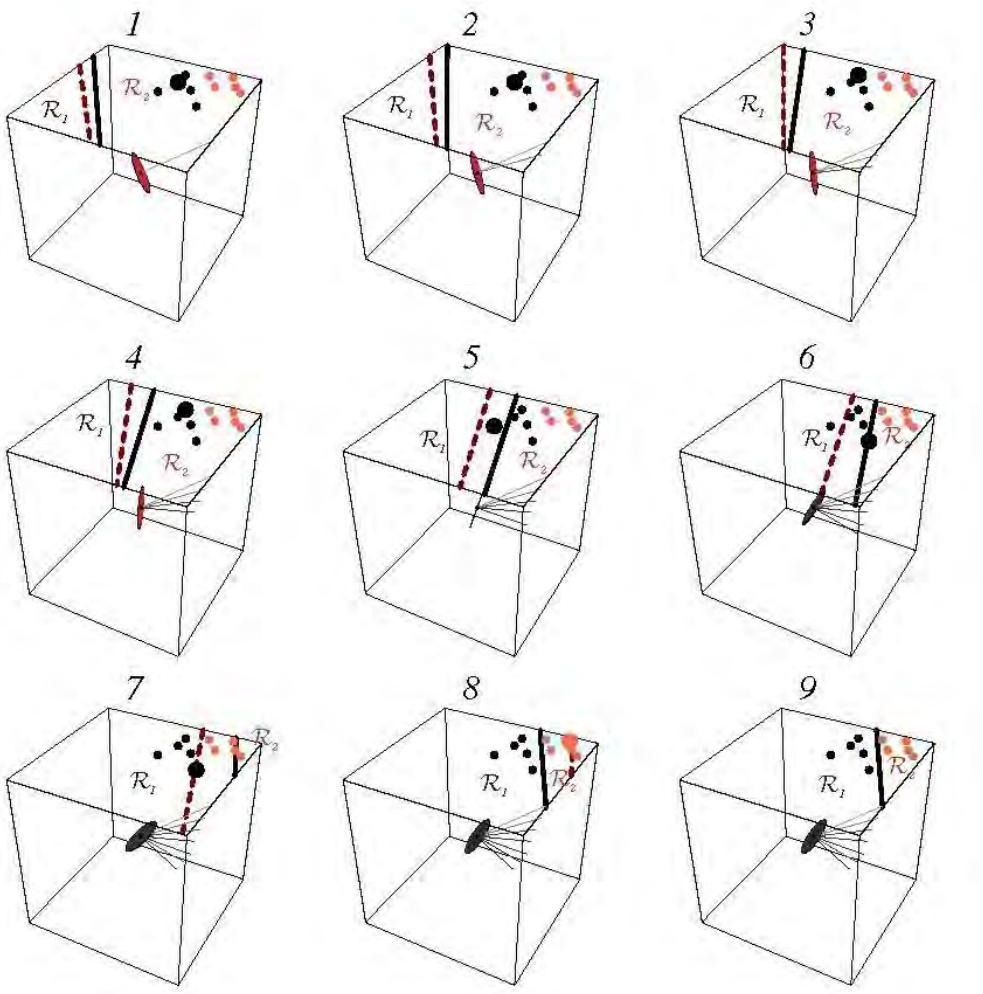
```
1 begin initialize a, k=0
2   do k ← (k+1) mod n
3     If y^k is misclassified by a, then a ← a + y^k
4   until all patterns properly classified
5   return a
6 end
```

Κυκλική Σειρά Δεδομένων (με πράσινο υποδηλώνονται τα λάθος ταξινομημένα δείγματα):

y_1 y_2 y_3 y_4 y_1 y_2 y_3 y_4 y_1 y_2 y_3 y_4

➔ y^1 y^2 y^3 y^0 $y^1 \ldots = y_2$ y_1 y_3 y_2 y_3

Σύγκλιση

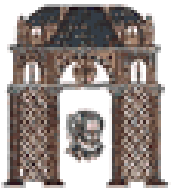


Πεπερασμένος αριθμός βημάτων, k_0, για σύγκλιση στη λύση a_0:

$$k_0 = \frac{\max_i \|\mathbf{y}_i\|^2 \|\mathbf{a}_0\|^2}{\min_i [\mathbf{y}_i^t \mathbf{a}_0]^2}$$

Ο παρονομαστής δηλώνει ότι η δυσκολία στη σύγκλιση καθορίζεται από τα δείγματα εκπαίδευσης y_i τα οποία είναι σχεδόν κάθετα στο διάνυσμα λύσης a_0: Γραμμικά διαχωρίσιμα προβλήματα είναι δύσκολα επιλύσιμα όταν τα πρότυπα είναι σχεδόν ομοεπίπεδα.

ΔΥΣΤΥΧΩΣ ΤΟ ΟΡΙΟ ΕΚΦΡΑΖΕΤΑΙ ΣΥΝΑΡΤΗΣΕΙ ΤΗΣ ΛΥΣΗΣ α_0



Variable-Increment Perceptron with Margin

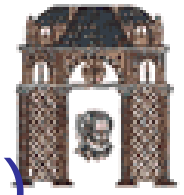
- Χρησιμοποιούμε τα δείγματα ένα κάθε φορά και διορθώνουμε το διάνυσμα βαρών $\boldsymbol{a}(k)$ όταν το εσωτερικό γινόμενό του με το δείγμα $\boldsymbol{y}^k$ είναι μικρότερο από κάποιο προκαθορισμένο θετικό όριο b: $\boldsymbol{a}(k)\boldsymbol{y}^k < b$.
- Αν επιπλέον, χρησιμοποιήσουμε ένα μεταβαλλόμενο βήμα $\eta(k)$, τότε προκύπτει ο αλγόριθμος:

Αλγόριθμος 5. Variable-Increment Perceptron w. Margin

```
1 begin initialize a, threshold θ, margin b, η(0), k=0
2   do  k ← (k+1) mod n
3     if a^t y^k<b, then a ← a + η(k) y^k
4   until a^t y^k>b for all k
5   return a
6 end
```

Συνθήκες Σύγκλισης:

$$\eta(k) \geq 0, \quad \lim_{m\to\infty} \sum_{k=1}^{m} \eta(k) = \infty, \quad \lim_{m\to\infty} \frac{\sum_{k=1}^{m} \eta^2(k)}{\left(\sum_{k=1}^{m} \eta(k)\right)^2} = 0$$



Μέθοδοι Χαλάρωσης (Relaxation Procedures)

- Συνάρτηση κριτηρίου:

$$J_r(\mathbf{a}) = \frac{1}{2}\sum_{\mathbf{y}\in Y}\frac{\left(\mathbf{a}^t\mathbf{y}-b\right)^2}{\|\mathbf{y}\|^2}$$

όπου $Y(\mathbf{a})$ είναι το σύνολο των δειγμάτων για τα οποία $\mathbf{a}^t\mathbf{y}<b$.

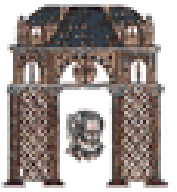
- Αν το $Y(\mathbf{a})$ είναι κενό, τότε $J_r(\mathbf{a})=0$. Η $J_r(\mathbf{a})$ δεν είναι ποτέ αρνητική και μηδενίζεται αν και μόνο αν $\mathbf{a}^t\mathbf{y}>b$ για όλα τα δείγματα εκπαίδευσης.

- Το διάνυσμα κλίσεων είναι:

$$\nabla J_r(\mathbf{a}) = \left[\frac{\partial J_r}{\partial a_i}\right] = \sum_{y\in Y}\frac{\mathbf{a}^t\mathbf{y}-b}{\|\mathbf{y}\|^2}\mathbf{y}$$

- Αναδρομική σχέση:

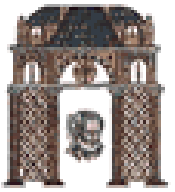
$$\mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k)\sum_{y\in Y_k}\frac{b-\mathbf{a}^t\mathbf{y}}{\|\mathbf{y}\|^2}\mathbf{y}$$



Batch Relaxation with Margin

Αλγόριθμος 6. Batch Relaxation with Margin

1 **begin initialize** **a**, margin b, $\eta(0)$, $k=0$
2 **do** $k \leftarrow (k+1)$ mod n
3 $Y_k=\{\}$
4 $j=0$
5 **do** $j \leftarrow j+1$
6 **if** $\mathbf{a}^t\mathbf{y}^j<b$, **then** append $\mathbf{y}^j$ to Y_k
7 **until** $j=n$
8 $\mathbf{a} \leftarrow \mathbf{a}+\eta(k)\sum_{y\in Y}\frac{b-\mathbf{a}^t\mathbf{y}}{\|\mathbf{y}\|^2}\mathbf{y}$
9 **until** $Y_k=\{\}$
10 **return** **a**
11 **end**



Single-Sample Relaxation with Margin

Αλγόριθμος 7. Single-Sample Relaxation with Margin

1 **begin initialize** **a**, margin b, $\eta(0)$, $k=0$

2 **do** $k \leftarrow (k+1) \bmod n$

3 **if** $\mathbf{a}^t\mathbf{y}^k<b$, **then** $\mathbf{a} \leftarrow \mathbf{a}+\eta(k)\dfrac{b-\mathbf{a}^t\mathbf{y}^k}{\|\mathbf{y}^k\|^2}\mathbf{y}^k$

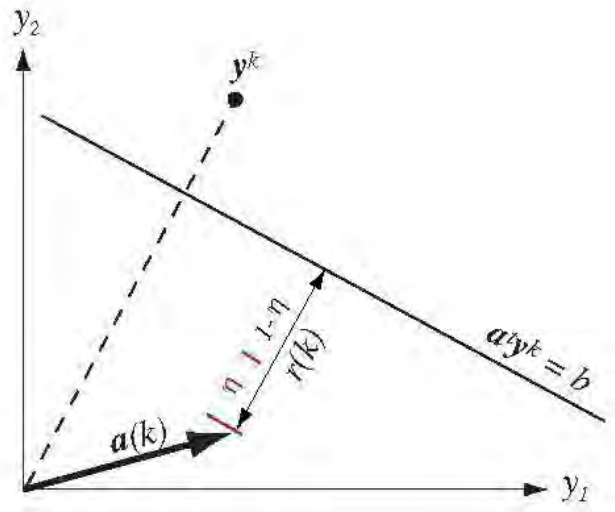
4 **until** $\mathbf{a}^t\mathbf{y}^k>b$ for all $\mathbf{y}^k$

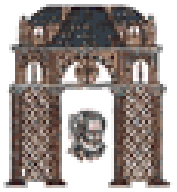
5 **return** **a**

6 **end**

Σε κάθε βήμα, το διάνυσμα βαρών ***a(k)***, μετατοπίζεται προς το υπερεπίπεδο $\mathbf{a}^t\mathbf{y}^k=b$ κατά ένα ποσοστό, $\eta(k)$, της απόστασής του, $r(k)$, από αυτό.

$\eta(k)<1$ → underrelaxation
$\eta(k)>1$ → overrelaxation
$0<\eta(k)<2$ για σύγκλιση





Μη Διαχωρίσιμες Κατηγορίες

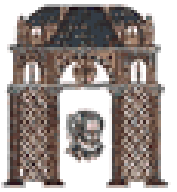
- Οι προηγούμενοι αλγόριθμοι βασίζονται στην υπόθεση ότι οι κατηγορίες είναι γραμμικά διαχωρίσιμες.
- Αλλά ακόμα και εάν το δείγμα εκπαίδευσης είναι γραμμικά διαχωρίσιμο αυτό δεν εγγυάται καλή συμπεριφορά στην πραγματικότητα

ΠΩΣ ΘΑ ΣΥΜΠΕΡΙΦΕΡΟΝΤΑΙ ΣΕ ΜΗ ΓΡΑΜΜΙΚΑ ΔΙΑΧΩΡΙΣΙΜΕΣ ΚΑΤΗΓΟΡΙΕΣ

Ας βάλλουμε όλα τα δεδομένα με ένα margin $a^T y_i - b_i > 0$

και ας προσπαθήσουμε να ελαχιστοποιήσουμε μια συνάρτηση κριτηρίου που βασίζεται στα ελάχιστα τετράγωνα ->

Μέθοδο Ελαχίστων Τετραγώνων



Μέθοδοι Ελαχίστων Τετραγώνων (Minimum Square Error – MSE)

- Στόχος: Καλή απόδοση τόσο στις γραμμικά διαχωρίσιμες όσο και στις μη γραμμικά διαχωρίσιμες περιπτώσεις.
- Πώς: Κριτήριο που περιλαμβάνει όλα τα πρότυπα. Επίσης,
  - Πρίν: Βρες $\boldsymbol{a}$ έτσι ώστε $\boldsymbol{a}^t\boldsymbol{y}_i>0$ για κάθε πρότυπο $\boldsymbol{y}_i$.
  - Τώρα: Βρες $\boldsymbol{a}$ έτσι ώστε $\boldsymbol{a}^t\boldsymbol{y}_i=b_i$ για κάθε πρότυπο $\boldsymbol{y}_i$. (b_i θετικές σταθερές).
- Δηλαδή; Μετατρέπουμε το πρόβλημα επίλυσης ενός συνόλου γραμμικών ανισοτήτων σε πρόβλημα επίλυσης γραμμικών εξισώσεων.
- Συμβολισμοί:

$$\mathbf{Y}_{n\times\hat{d}}=\begin{bmatrix}\mathbf{y}_1^t\\ \mathbf{y}_2^t\\ \vdots\\ \mathbf{y}_n^t\end{bmatrix}$$

Πίνακας Προτύπων

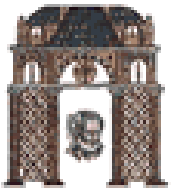
$$\mathbf{b}_{n\times 1}=\begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_n\end{bmatrix}$$

Διάνυσμα θετικών παραμέτρων

$$\mathbf{a}_{\hat{d}\times 1}=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_d\end{bmatrix}$$

Διάνυσμα βαρών

- Πρόβλημα: Εύρεση $\boldsymbol{a}$ έτσι ώστε: $\boldsymbol{Ya}=\boldsymbol{b}$.



Μέθοδοι Ελαχίστων Τετραγώνων (Minimum Square Error – MSE)

- Συνήθως $n>d+1$, περισσότερα πρότυπα από διαστάσεις → σύστημα υπερπροσδιορισμένο, δεν έχει ακριβή λύση.
- Επομένως; Ελαχιστοποίηση του τετραγώνου του μήκους του διανύσματος σφάλματος, $\boldsymbol{e=Ya-b}$:

$$J_s(\mathbf{a}) = \|\mathbf{Ya}-\mathbf{b}\|^2 = \sum_{i=1}^{n}\left(\mathbf{a}^t\mathbf{y}_i - b_i\right)^2$$

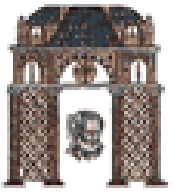
- Κλασσικό Πρόβλημα:

$$\nabla J_s(\mathbf{a}) = \mathbf{0} \Rightarrow \mathbf{Y}^t\mathbf{Ya} = \mathbf{Y}^t\mathbf{b}$$

Κανονικές Εξισώσεις
Normal Equations

- Αν ο $\boldsymbol{Y^tY}$ είναι ομαλός,

$$\mathbf{a} = \underbrace{\left(\mathbf{Y}^t\mathbf{Y}\right)^{-1}\mathbf{Y}^t}_{\text{ψευδοαντίστροφος}}\ \mathbf{b}$$

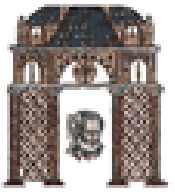


Widrow-Hoff (Least Mean Squares - LMS)

- Η $J_s(\boldsymbol{a})$ μπορεί να ελαχιστοποιηθεί μέσω αναδρομικών αλγορίθμων που δεν απαιτούν την αντιστροφή πινάκων.
- Διάνυσμα κλίσεων: $\nabla J_s(\mathbf{a}) = 2\mathbf{Y}^t\left(\mathbf{Y}\mathbf{a} - \mathbf{b}\right)$
- Βασική αναδρομική σχέση: $\mathbf{a}(k+1) = \mathbf{a}(k) + \eta(k)\mathbf{Y}^t\left(\mathbf{b} - \mathbf{Y}\mathbf{a}(k)\right)$
- Χρησιμοποιώντας ένα δείγμα σε κάθε βήμα, προκύπτει ο αλγόριθμος Widrow-Hoff (LMS):

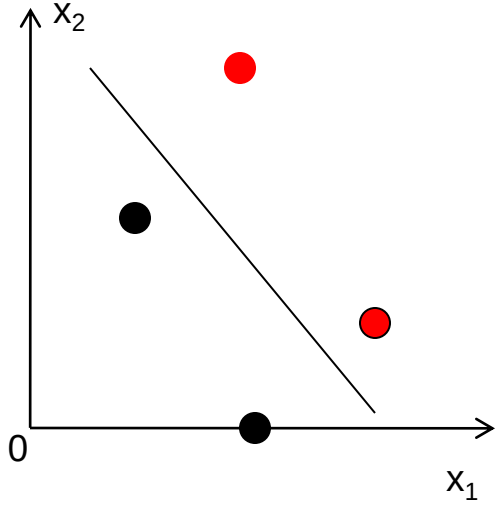
Αλγόριθμος 8. Widrow-Hoff (LMS)

1 **begin initialize** a, b, κριτήριο θ, η(), k=0
2 **do** $k \leftarrow (k+1) \bmod n$
3 $\mathbf{a} \leftarrow \mathbf{a} + \eta(k)\left(b_k - \mathbf{a}^t\mathbf{y}^k\right)\mathbf{y}^k$
4 **until** $\left|\eta(k)\left(b_k - \mathbf{a}^t\mathbf{y}^k\right)\mathbf{y}^k\right| < \theta$
5 **return** a
6 **end**



Παράδειγμα

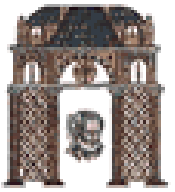
- Υποθέσουμε ότι έχουμε τα παρακάτω σημεία 2 διαστάσεων που ανήκουν σε 2 κατηγορίες:
- ω_1: (1, 2) και (2, 0), και ω_2: (3, 1) και (2, 3), τα πρώτα είναι με μαύρο και τα δεύτερα με κόκκινο χρώμα στο επόμενο σχήμα.
- Ο πίνακας **Y** είναι επομένως
- **Y** = (1 1 2, 1 2 0, −1 −3 −1, −1 −2 −3)
- και μετά από μερικές απλές πράξεις έχουμε ότι ο ψευδοαντίστροφος πίνακας είναι:



$$Y=\begin{pmatrix}1 & 1 & 2\\ 1 & 2 & 0\\ -1 & -3 & -1\\ -1 & -2 & -3\end{pmatrix} \qquad a=\left(Y^{T}Y\right)^{-1}Y^{T}b=\begin{pmatrix}5/4 & 13/12 & 3/4 & 7/12\\ -1/2 & -1/6 & -1/2 & -1/6\\ 0 & -1/3 & 0 & -1/3\end{pmatrix}b$$

Και διαλέγοντας αυθαίρετα $b^T=(1,1,1,1)$, έχουμε **a^T= (11/3, -4/3, -2/3)** άρα

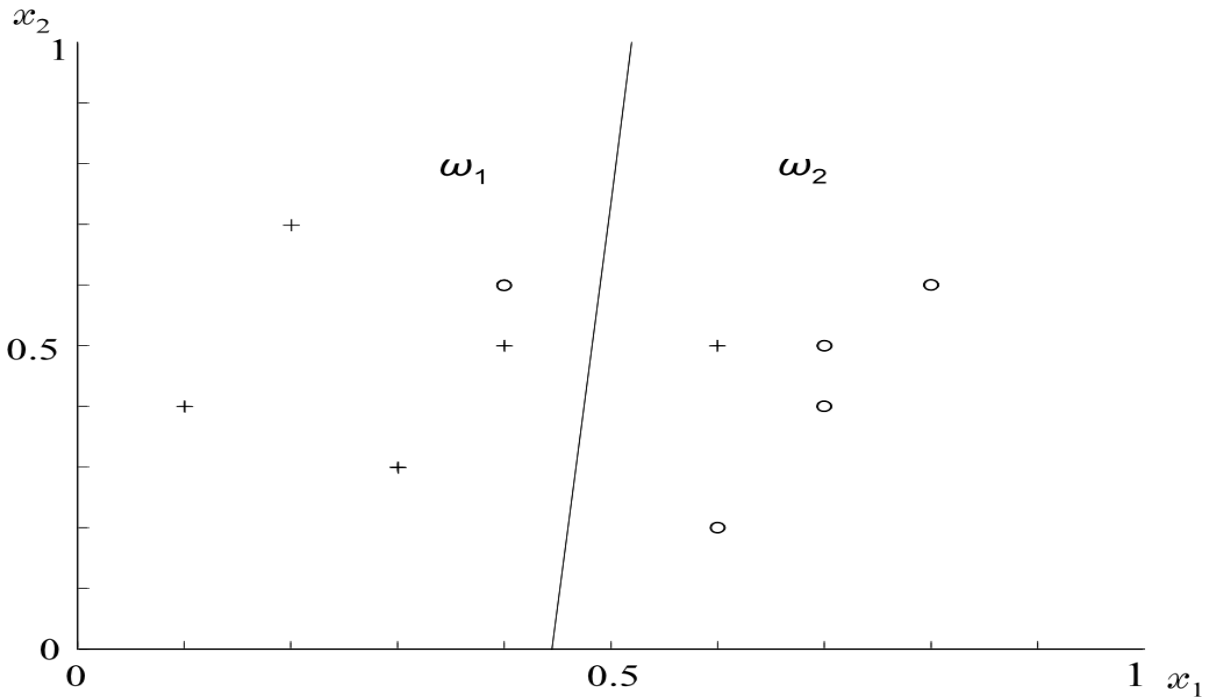
$$g(\mathbf{x})=(11/3)-(4/3)x_1-(2/3)x_2$$



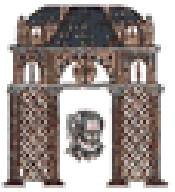
Παράδειγμα 2

$$\omega_1 : \begin{bmatrix}0.4\\0.5\end{bmatrix}, \begin{bmatrix}0.6\\0.5\end{bmatrix}, \begin{bmatrix}0.1\\0.4\end{bmatrix}, \begin{bmatrix}0.2\\0.7\end{bmatrix}, \begin{bmatrix}0.3\\0.3\end{bmatrix}$$

$$\omega_2 : \begin{bmatrix}0.4\\0.6\end{bmatrix}, \begin{bmatrix}0.6\\0.2\end{bmatrix}, \begin{bmatrix}0.7\\0.4\end{bmatrix}, \begin{bmatrix}0.8\\0.6\end{bmatrix}, \begin{bmatrix}0.7\\0.5\end{bmatrix}$$



$$Y = \begin{bmatrix} 1 & 0.4 & 0.5 \\ 1 & 0.6 & 0.5 \\ 1 & 0.1 & 0.4 \\ 1 & 0.2 & 0.7 \\ 1 & 0.3 & 0.3 \\ -1 & -0.4 & -0.6 \\ -1 & -0.6 & -0.2 \\ -1 & -0.7 & -0.4 \\ -1 & -0.8 & -0.6 \\ -1 & -0.7 & -0.5 \end{bmatrix}^{\mathrm{T}}$$



$$b = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$$

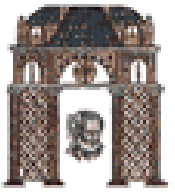
$$Y^T Y = \begin{bmatrix} 10 & 4.8 & 4.7 \\ 4.8 & 2.8 & 2.24 \\ 4.7 & 2.24 & 2.41 \end{bmatrix}, Y^T \underline{b} = \begin{bmatrix} 0.0 \\ -1.6 \\ 0.1 \end{bmatrix}$$

$$\underline{a} = (Y^T Y)^{-1} Y^T \underline{b} = \begin{bmatrix} 1.43 \\ -3.21 \\ 0.24 \end{bmatrix}$$

$$g(\mathbf{x}) = ((Y^T Y)^{-1} Y^T \underline{b}) \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = [1.43 \quad -3.21 \quad 0.24] \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 1.434 - 3.21 x_1 + 0.24 x_2$$

Σημείωση: Στο MSE μπορούμε αντί να αλλάξουμε το πρόσημο στα Y για την κλάση ω_2 να αλλάξουμε το πρόσημο στο b για τα δεδομένα που ανήκουν στη κλάση ω_2 , δηλαδή

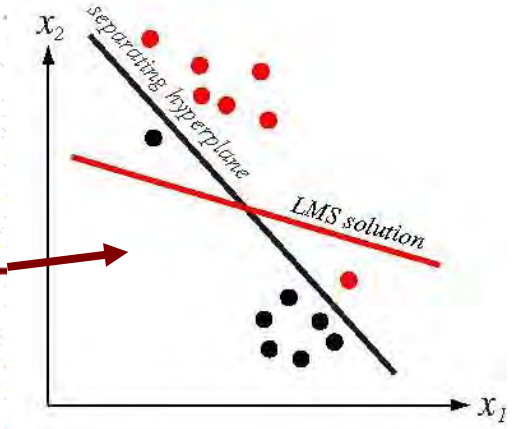
$$\mathbf{Y} = \begin{bmatrix} 1 & 0.4 & 0.5 \\ 1 & 0.6 & 0.5 \\ 1 & 0.1 & 0.4 \\ 1 & 0.2 & 0.7 \\ 1 & 0.3 & 0.3 \\ 1 & 0.4 & 0.6 \\ 1 & 0.6 & 0.2 \\ 1 & 0.7 & 0.4 \\ 1 & 0.8 & 0.6 \\ 1 & 0.7 & 0.5 \end{bmatrix}^T \quad \mathbf{b}^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$



Μέθοδος Ho-Kashyap

- Το perceptron και οι τεχνικές χαλάρωσης βρίσκουν διαχωριστικά διανύσματα βαρών αν τα δείγματα είναι γραμμικά διαχωρίσιμα, αλλά δεν συγκλίνουν για μη διαχωρίσιμες κλάσεις.

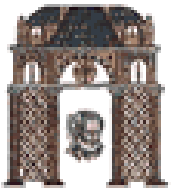
- Οι τεχνικές ελαχίστων τετραγώνων δίνουν πάντα ένα διάνυσμα λύσης (αυτό που ελαχιστοποιεί το $\|\mathbf{Ya}-\mathbf{b}\|^2$) το οποίο όμως δεν είναι απαραίτητα διαχωριστικό στην διαχωρίσιμη περίπτωση.



- Ο αλγόριθμος Ho-Kashyap λύνει αναδρομικά το εξής πρόβλημα ελαχιστοποίησης:

$$\min_{\mathbf{a},\mathbf{b}} J_s(\mathbf{a},\mathbf{b}) = \|\mathbf{Ya}-\mathbf{b}\|^2 \quad s.t. \quad \mathbf{b} > \mathbf{0}$$

- Είναι ένας αλγόριθμος που φροντίζει ώστε το $\boldsymbol{b}$ να μην συγκλίνει στο $\boldsymbol{0}$, θέτοντας όλες τις θετικές συνιστώσες του διανύσματος κλίσης $\nabla_b J_s$ ίσες με το μηδέν.

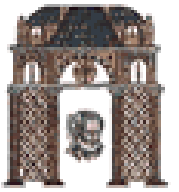


Αλγόριθμος Ho-Kashyap

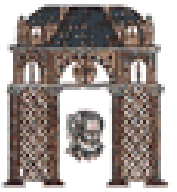
Αλγόριθμος 9. Ho-Kashyap

```
1  begin initialize a, b, η()<1, threshold b_min, k_max
2     do k ← (k+1) mod n
3        e ← Ya-b
4        e+ ← (e+|e|)/2
5        b ← b + 2η(k)e+
6        a ← (YtY)-1Yb
7        if Abs(e)< b_min then return a, b and exit
8     until k= k_max
9     print "No solution found"
10 end
```

Βασικά βρίσκουμε πρώτα το gradient descent ως προς b και μετά εφαρμόζουμε ελάχιστα τετράγωνα



- Έωο ηώξα είδακε ην πξόβι εκ α κόλν γηα2 θαηεγνξί εο. Τη κπνξνύ κε λα θάλνπκε εάλ έρνπκε πνιιέο θαηεγνξίεο πξνηύπωλ?
- Θα δνύκε ην πξόβιεκα π πνπζέ ηην ληαοόηεηνγηα θαηάζεηο είλαηγξαθηθή δηαρωξίζηκεο. Βαζηθόκελνηζηηογξαθηθέο ζπλαξηήζεωλ δηαθξίλεη ε, ζέιν πκε λα βξνύκ ε έλα βξνύκ ε έλα ζύλνιν $g_i(x)$ γξαθηθώλ ζπλαξηήζεωλ δηαθξίλεη ε έηζη ώζ ηε εάλ $x \in \omega_i$, ηόηε $g_i(x) > g_j(x)$ γηα θάζε $j \neq i$.
- Δηαηηπώλνπκε ηώξα ην πξόβιεκα ζ ηην λ ρώξν ηωλ **y**, (ην ζπλαξηήζεωλ δηαθξίλεη ε ε είλαηηπεξί πεδα πνπ πεξλνύλ από ηελ αξρή ηωλ αμόλωλ.



Ταξινόμηση Πολλαπλών Κλάσεων με MSE

- Έστω $\mathbf{y_1, \ldots, y_n}$, πρότυπα από c>2 κλάσεις. Θέλουμε να βρούμε ένα ταξινομητή που να αποτελείται από γραμμικές συναρτήσεις διάκρισης $g_i(\mathbf{x}) = w_i\mathbf{x} + w_{i0}$ (μία για κάθε κλάση), ούτως ώστε εάν x ανήκει στην i κλάση τότε $g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \forall j \neq i$. Συνεπώς στον επαυξημένο χώρο ψάχνουμε για τα βάρη $\mathbf{a_i}$ ώστε εάν y_k ανήκει στην i κλάση τότε

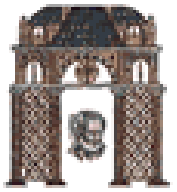
$$\mathbf{a_i} : \mathbf{a_i^t y}_\kappa > \mathbf{a_j^t y}_\kappa \quad \forall j \neq i$$

- Για την επίλυση του προβλήματος συνεπώς υπολογίζουμε το βέλτιστο $\mathbf{a_i}$ ώστε να διαχωρίζει την κλάση i από όλες τις υπόλοιπες κλάσεις j≠i .

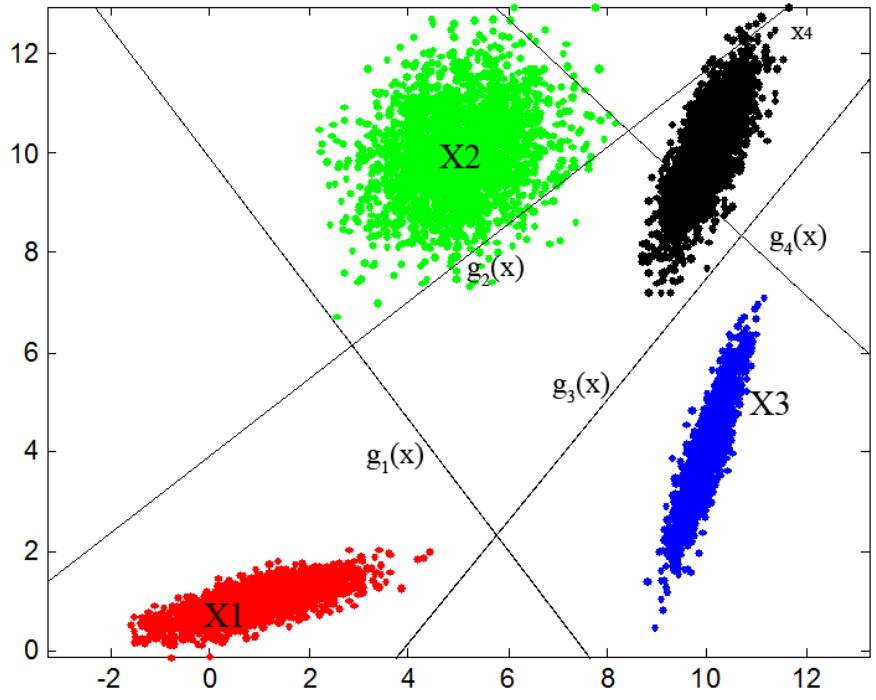
$$\mathbf{a}_\iota = \left(\mathbf{Y^T Y}\right)^{-1} \mathbf{b}_\iota$$

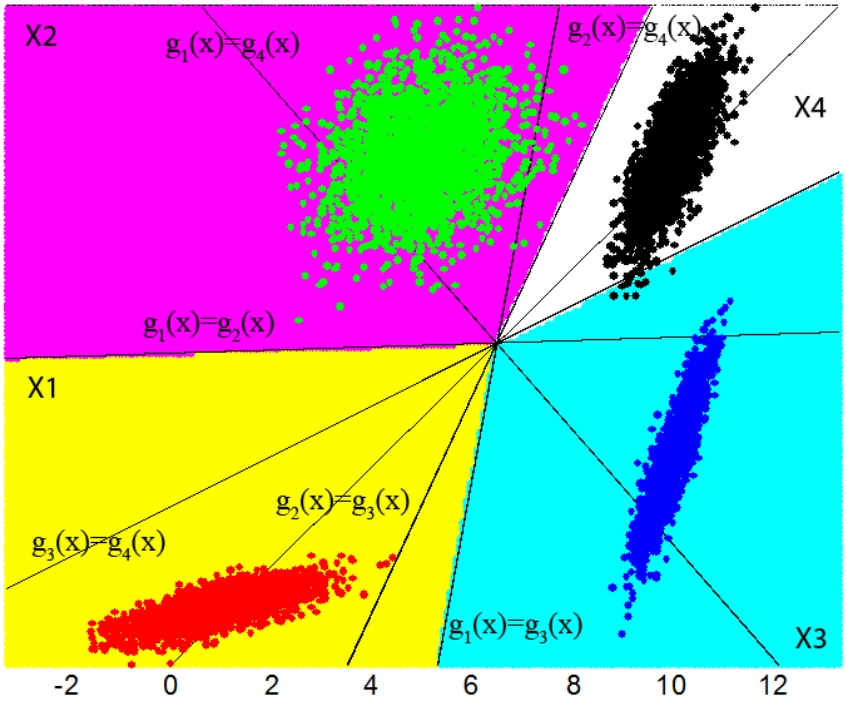
Y= [$\mathbf{y_1, \ldots, y_n}$] και $\mathbf{b_i = [b_{jk}]}$ k=1,…,n και b_{jk} =1 εάν $\mathbf{y_j}$ ανήκει στην κλάση ω_i και b_{jk} =-1 εάν $\mathbf{y_j}$ **δεν** ανήκει στην κλάση ω_i

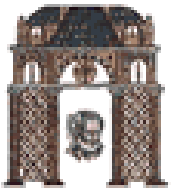
- Σημειώνεται ότι δεν χρειάζεται να αλλάξουμε το πρόσημο στα $\mathbf{y_j}$ αλλά μπορούμε να το αλλάξουμε στο $\mathbf{b_i}$



MSE ταξινόμηση







Πολλαπλές Κλάσεις
Δομή του Kesler

- Στόχος: Γραμμικός διαχωρισμός πολλαπλών κλάσεων:

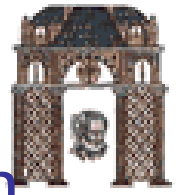
  Αν $y \sim \omega_1$, τότε $\boldsymbol{a}_1^t y - \boldsymbol{a}_j^t y > 0$ για κάθε $j=2,\ldots,c$.

- Αυτό το σύστημα των c-1 ανισοτήτων μπορεί να περιγραφεί ως εξής:
  Το $c\hat{d}-D$ διάνυσμα βαρών $\hat{\boldsymbol{\alpha}}$ ταξινομεί ορθά όλα τα c-1 $c\hat{d}-D$ πρότυπα $\boldsymbol{\eta}_{12}, \boldsymbol{\eta}_{13},\ldots, \boldsymbol{\eta}_{1c}$: $\hat{\boldsymbol{\alpha}}^t \boldsymbol{\eta}_{1j} > 0 \quad \forall \quad j = 2,\ldots,c$ όπου:

$$\hat{\boldsymbol{\alpha}}_{c\hat{d}\times 1} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_c \end{bmatrix} \quad \boldsymbol{\eta}_{12} = \begin{bmatrix} \mathbf{y} \\ -\mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad \boldsymbol{\eta}_{13} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ -\mathbf{y} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad ,\cdots, \quad \boldsymbol{\eta}_{1c} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\mathbf{y} \end{bmatrix}$$

Δομή Kesler

- Γενικότερα: $\hat{\boldsymbol{\alpha}}^t \boldsymbol{\eta}_{ij} > 0 \quad \forall \quad j \neq i,$ όπου $\boldsymbol{\eta}_{ij} = \begin{bmatrix} \vdots \\ \mathbf{y} \leftarrow i \\ \vdots \\ -\mathbf{y} \leftarrow j \\ \vdots \end{bmatrix}$



Ταξινόμηση Πολλαπλών Κλάσεων Perceptron

- Έστω $y_1,\ldots, y_n$ πρότυπα από c κλάσεις, γραμμικά διαχωρίσιμα. Έστω L_k μία γραμμική μηχανή $a_1(k),\ldots, a_c(k)$. Θέλουμε να κατασκευάσουμε μία ακολουθία γρ. μηχ. $L_1,\ldots,L_k,\ldots$ που να συγκλίνει σε μία διαχωριστική μηχανή L.

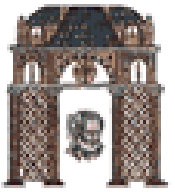
- Έστω y^k το k-στό δείγμα που ζητά διόρθωση (σωστή ταξινόμηση). Αν $y^k \sim \omega_i$, σημαίνει ότι υπάρχει τουλάχιστον ένα $j \neq i$ για το οποίο $a_i^t(k)\, y^k < a_j^t(k)\, y^k$.

- Ο κανόνας διόρθωσης του L_k (perceptron με σταθερό μοναδιαίο βήμα) λέει:

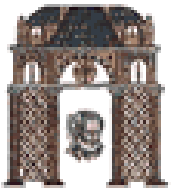
$$\boldsymbol{\alpha}(k+1)=\boldsymbol{\alpha}(k)+\boldsymbol{\eta}_{ij}^k \quad \text{όπου} \quad \boldsymbol{\alpha}^t(k)\boldsymbol{\eta}_{ij}^k \le 0 \quad \text{με} \quad \boldsymbol{\alpha}(k)=\begin{bmatrix}\mathbf{a}_1(k)\\ \vdots \\ \mathbf{a}_c(k)\end{bmatrix} \quad \text{και} \quad \boldsymbol{\eta}_{ij}^k=\begin{bmatrix}\vdots \\ \mathbf{y}^k \\ \vdots \\ -\mathbf{y}^k \\ \vdots\end{bmatrix} \begin{matrix}\\ \leftarrow i \\ \\ \leftarrow j \\ \\ \end{matrix}$$

Δηλαδή:

$$\begin{aligned}\mathbf{a}_i(k+1)&=\mathbf{a}_i(k)+\mathbf{y}^k\\ \mathbf{a}_j(k+1)&=\mathbf{a}_j(k)-\mathbf{y}^k\\ \mathbf{a}_l(k+1)&=\mathbf{a}_l(k) \quad l\neq i \quad \text{και} \quad l\neq j\end{aligned}$$



- ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ

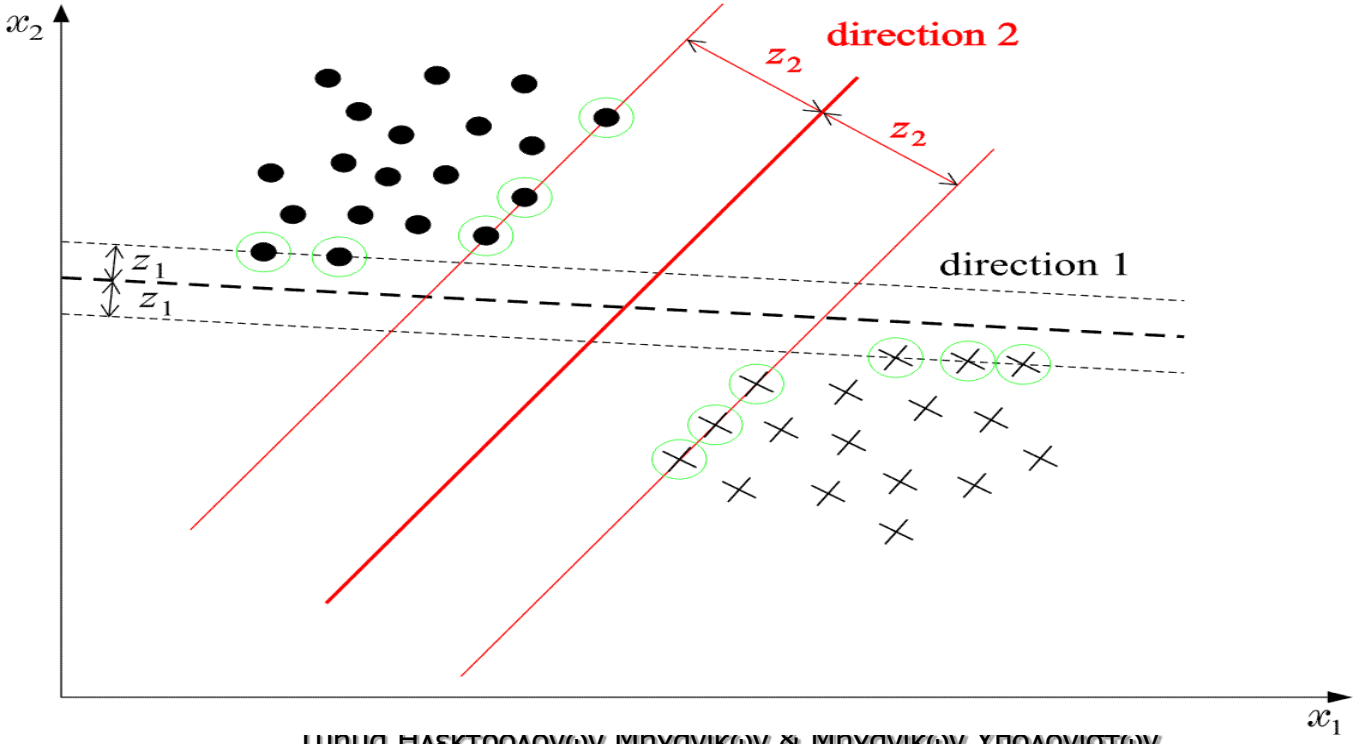


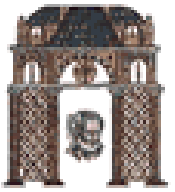
SUPPORT VECTOR MACHINES

- Στοχος: Εάν έχουμε, δύο κατηγορίες που είναι γραμμικά διαχωρίσιμες , να βρείς την συνάρτηση διάκρισης

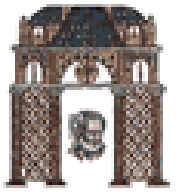
$$g(\underline{x}) = \underline{w}^T \underline{x} + w_0 = 0$$

- Η οποία αφήνει την μέγιστη απόσταση, περιθώριο, (maximum margin) και από τις δύο κατηγορίες from both classes





- Margin: Each hyperplane is characterized by
  - Its direction in space, i.e., $\underline{w}$
  - Its position in space, i.e., w_0
  - For EACH direction, $\underline{w}$ choose the hyperplane that leaves the SAME distance from the nearest points from each class. The margin is twice this distance.



- The distance of a point $\hat{\underline{x}}$ from a hyperplane is given by

$$z_{\hat{x}} = \frac{g(\hat{\underline{x}})}{\|\underline{w}\|}$$

- Scale, $\underline{w}, \underline{w}_0$, so that at the nearest points from each class the discriminant function is ±1:

$$|g(\underline{x})| = 1 \ \{g(\underline{x}) = +1 \text{ for } \omega_1 \text{ and } g(\underline{x}) = -1 \text{ for } \omega_2\}$$

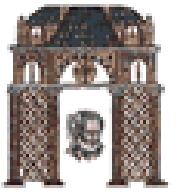
- Thus the margin is given by

$$\frac{1}{\|\underline{w}\|} + \frac{1}{\|\underline{w}\|} = \frac{2}{\|\underline{w}\|}$$

- Also, the following is valid

$$\underline{w}^T \underline{x} + w_0 \geq 1 \ \ \forall \underline{x} \in \omega_1$$

$$\underline{w}^T \underline{x} + w_0 \leq -1 \ \ \forall \underline{x} \in \omega_2$$



- SVM (linear) classifier

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

- Minimize

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

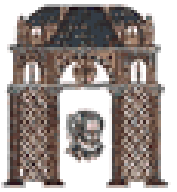
- Subject to

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

$$y_i = 1, \text{ for } \underline{x}_i \in \omega_i,$$

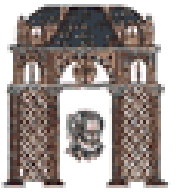
$$y_i = -1, \text{ for } \underline{x}_i \in \omega_2$$

- This is because minimizing $\|\underline{w}\|$ the margin $\frac{2}{\|\underline{w}\|}$ is maximised

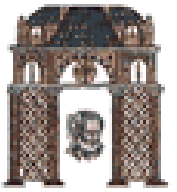


- The above is a quadratic optimization task subject to a set of linear inequality constraints. The Karush-Kuhh-Tucker conditions, state that the minimizer satisfies:
  - (1) $$\frac{\partial}{\partial \underline{w}} \mathrm{L}(\underline{w}, w_0, \underline{\lambda}) = 0$$
  - (2) $$\frac{\partial}{\partial w_0} L(\underline{w}, w_0, \underline{\lambda}) = 0$$
  - (3) $$\lambda_i \geq 0, i = 1,2,\ldots,N$$
  - (4) $$\lambda_i \left[y_i (\underline{w}^T \underline{x}_i + w_0) - 1 \right] = 0, i = 1,2,\ldots,N$$
  - Where $L(.,.,.)$ is the Lagrangian

$$L(\underline{w}, w_0, \underline{\lambda}) \equiv \frac{1}{2} \underline{w}^T \underline{w} - \sum_{i=1}^{N} \lambda_i [y_i (\underline{w}^T \underline{x}_i + w_0)]$$



- The solution: from the above, it turns out that
  - $$\underline{w} = \sum_{i=1}^{N} \lambda_i y_i \underline{x}_i$$
  - $$\sum_{i=1}^{N} \lambda_i y_i = 0$$



Remarks:

- The Lagrange multipliers can be either zero or positive. Thus,

$$\underline{w} = \sum_{i=1}^{N_s} \lambda_i y_i \underline{x}_i$$

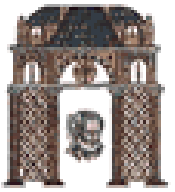
where $N_s \leq N_0$ corresponding to positive Lagrange multipliers

- From constraint (4) above, i.e.,

$$\lambda_i [y_i (\underline{w}^T \underline{x}_i + w_0) - 1] = 0, \quad i = 1,2,\ldots,N$$

- the vectors contributing to $\underline{w}$ satisfy

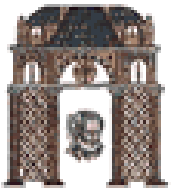
$$\underline{w}^T \underline{x}_i + w_0 = \pm 1$$



- These vectors are known as SUPPORT VECTORS and are the closest vectors, from each class, to the classifier.

- Once $\underline{w}$ is computed, w_0 is determined from conditions (4).

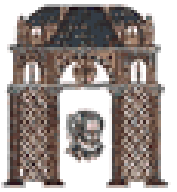
- The optimal hyperplane classifier of a support vector machine is UNIQUE.



Dual Problem Formulation

- The SVM formulation is a convex programming problem, with
  - Convex cost function
  - Convex region of feasible solutions
- Thus, its solution can be achieved by its dual problem, i.e.,
  - maximize $\underline{\lambda}$ $L(\underline{w}, w_0, \underline{\lambda})$
  - subject to

$$\underline{w} = \sum_{i=1}^{N} \lambda_i y_i \underline{x}_i \qquad \sum_{i=1}^{N} \lambda_i y_i = 0 \qquad \underline{\lambda} \geq \underline{0}$$



- Combine the above to obtain

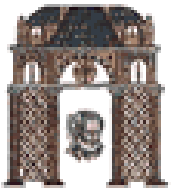
  - maximize $\underline{\lambda}$

$$(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{ij}\lambda_i\lambda_j y_i y_j \underline{x}_i^T \underline{x}_j)$$

  - subject to

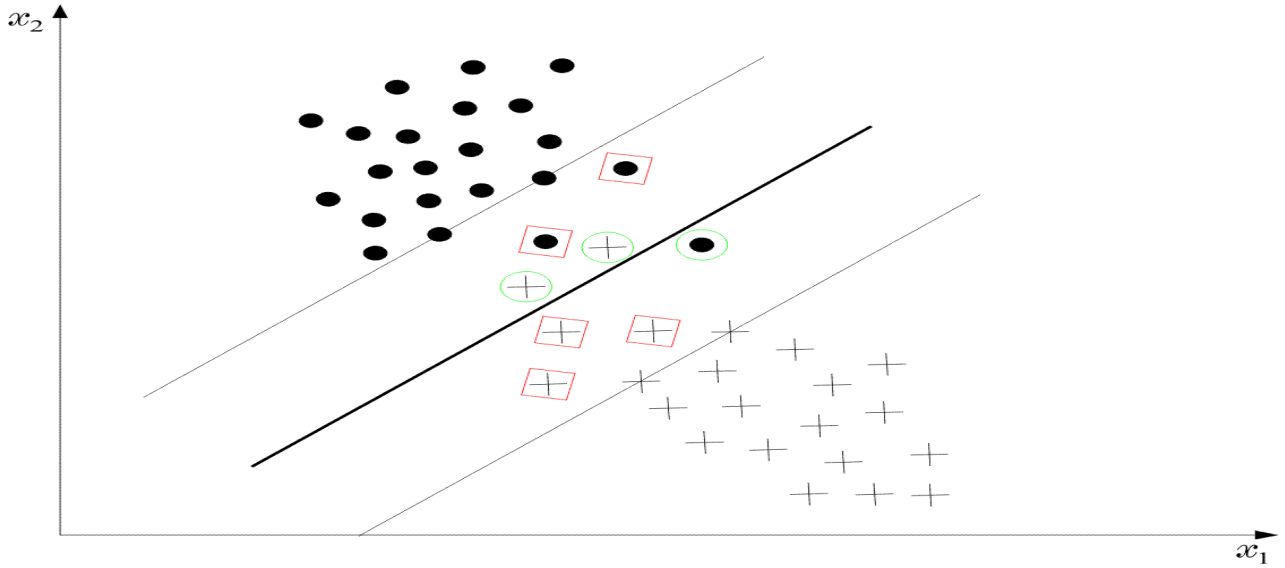
$$\sum_{i=1}^{N}\lambda_i y_i = 0$$

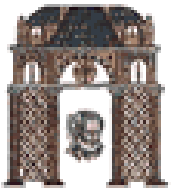
$$\underline{\lambda} \geq \underline{0}$$



- Remarks:
  - Support vectors enter via inner products
  - Although the solution, w, is unique, the Lagrange multipliers ARE NOT.

- Non-Separable classes





- In this case there is no hyperplane, such that

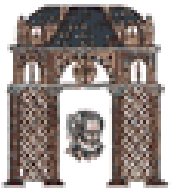
$$\underline{w}^T \underline{x} + w_0 (>=<) 1, \quad \forall \underline{x}$$

- Recall that the margin is defined as twice the distance between the following two hyperplanes

$$\underline{w}^T \underline{x} + w_0 = 1$$

$$\text{and}$$

$$\underline{w}^T \underline{x} + w_0 = -1$$



- The training vectors belong to one of three possible categories

  - A. Vectors outside the band which are correctly classified, i.e.,

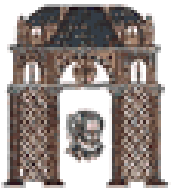
$$y_i(\underline{w}^T \underline{x} + w_0) > 1$$

  - B. Vectors inside the band, and correctly classified, i.e.,

$$0 \le y_i(\underline{w}^T \underline{x} + w_0) < 1$$

  - C. Vectors misclassified, i.e.

$$y_i(\underline{w}^T \underline{x} + w_0) < 0$$



- All three cases above can be represented as

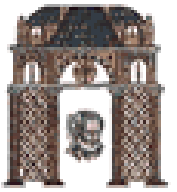
$$y_i(\underline{w}^T \underline{x} + w_0) \geq 1 - \xi_i$$

A. $\rightarrow \xi_i = 0$

B. $\rightarrow 0 < \xi_i \leq 1$

C. $\rightarrow 1 < \xi_i$

ξ_i are known as slack variables



The goal of the optimization is now two-fold

- Maximize margin
- Minimize the number of patterns with $\xi_i > 0$,

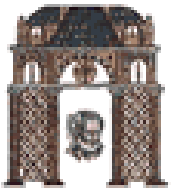
  That is

$$J(\underline{w}, w_0, \underline{\xi}) = \frac{1}{2}\|\underline{w}\|^2 + C\sum_{i=1}^{N} I(\xi_i)$$

  where C a constant and

$$I(\xi_i) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi_i = 0 \end{cases}$$

- I(.) not differentiable. In practice, we use an approximation
- $$J(\underline{w}, w_0, \underline{\xi}) = \frac{1}{2}\|\underline{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$
- Following similar procedure as before we obtain



KKT conditions

$$(1)\ \ \underline{w} = \sum_{i=1}^{N} \lambda_i y_i \underline{x}_i$$

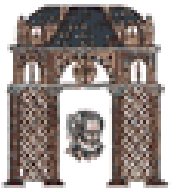
$$(2)\ \ \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$(3)\ \ C - \mu_i - \lambda_i = 0, i = 1,2,\ldots, N$$

$$(4)\ \ \lambda_i [y_i (\underline{w}^T \underline{x}_i + w_0) - 1 + \xi_i] = 0, \quad i = 1,2,\ldots, N$$

$$(5)\ \ \mu_i \xi_i = 0, \quad i = 1,2,\ldots, N$$

$$(6)\ \ \mu_i, \lambda_i \geq 0, \quad i = 1,2,\ldots, N$$



- The associated dual problem

Maximize

$$\underline{\lambda}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \underline{x}_i^T \underline{x}_j\right)$$

subject to

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

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

- Remarks:

  The only difference with the separable class case is the existence of C in the constraints