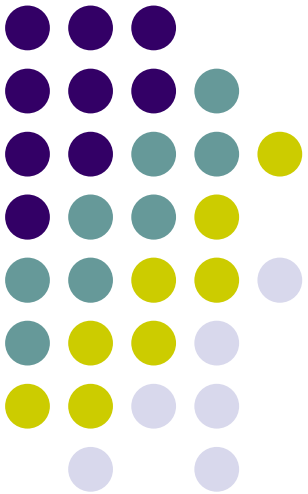
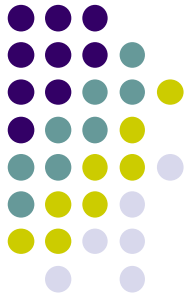
Εισαγωγή στη Μηχανή Διανυσμάτων Υποστήριξης

(Support Vector Machine - SVM)



Ανασκόπηση: Τι μάθαμε μέχρι τώρα



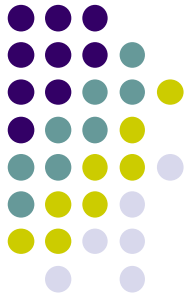
- Θεωρία Αποφάσεων του Bayes
- Maximum-Likelihood & Bayesian Εκτίμηση Παραμέτρων
- Μη Παραμετρική Εκτίμηση Πυκνότητας:
  - Parzen-Windows,
  - k_n-Nearest-Neighbor

Support Vector Machine (SVM)

- Είναι ένα ταξινομητής που έχει προκύψει από τη στατιστική θεωρία μάθησης (από τον Vapnik, et al. το 1992).
- Το SVM έγινε γνωστό όταν, χρησιμοποιώντας εικόνες σαν είσοδο, έδωσε αποτελέσματα συγκρίσιμα με τα Νευρωνικά Δίκτυα με χαρακτηριστικά σχεδιασμένα με το χέρι σε ένα έργο αναγνώρισης χειρογράφων.
- Σήμερα, το SVM χρησιμοποιείται ευρέως σε ανίχνευση αντικειμένων και αναγνώριση, σε ανάκτηση εικόνας βασιζόμενη σε περιεχόμενο, αναγνώριση κειμένου, στη βιοπληροφορικη, αναγνώριση ομιλίας, κλπ.

V. Vapnik

Διακρίνουσα Συνάρτηση (Discriminant Function)



- Ένας ταξινομητής αναθέτει ένα διάνυσμα χαρακτηριστικών *x* σε μια κλάση ω_i αν:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \text{for all} \quad j \neq i$$

- Για την περίπτωση 2 κατηγοριών ισχύει: $g(\mathbf{x}) \equiv g_1(\mathbf{x}) - g_2(\mathbf{x})$

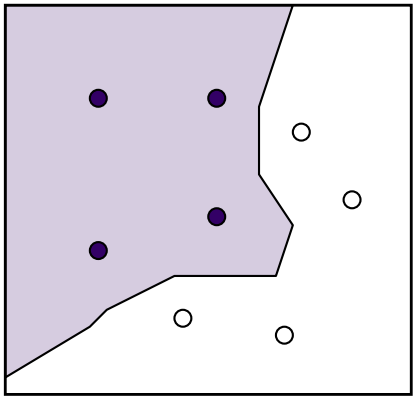
$$\text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0; \text{ otherwise decide } \omega_2$$

- Ένα παράδειγμα που έχουμε μάθει είναι:
  - Minimum-Error-Rate Classifier

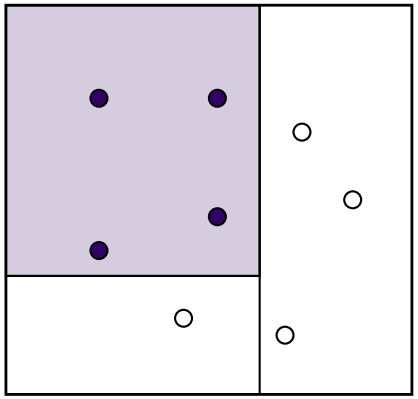
$$g(\mathbf{x}) \equiv p(\omega_1 \mid \mathbf{x}) - p(\omega_2 \mid \mathbf{x})$$

Διακρίνουσα Συνάρτηση (Discriminant Function)

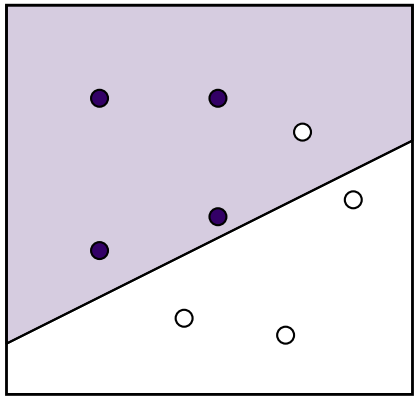
- Μπορεί να είναι μια κατάλληλη συνάρτηση του *x*, τέτοια ώστε:



Nearest Neighbor

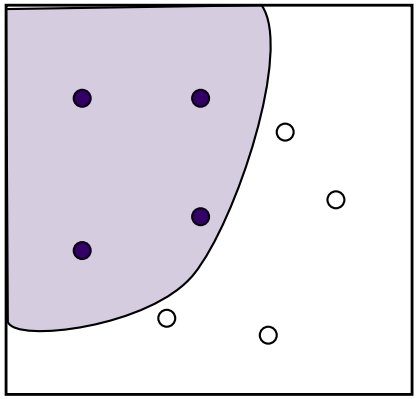


Decision Tree



Linear Functions

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



Nonlinear Functions

Γραμμική Discriminant Function

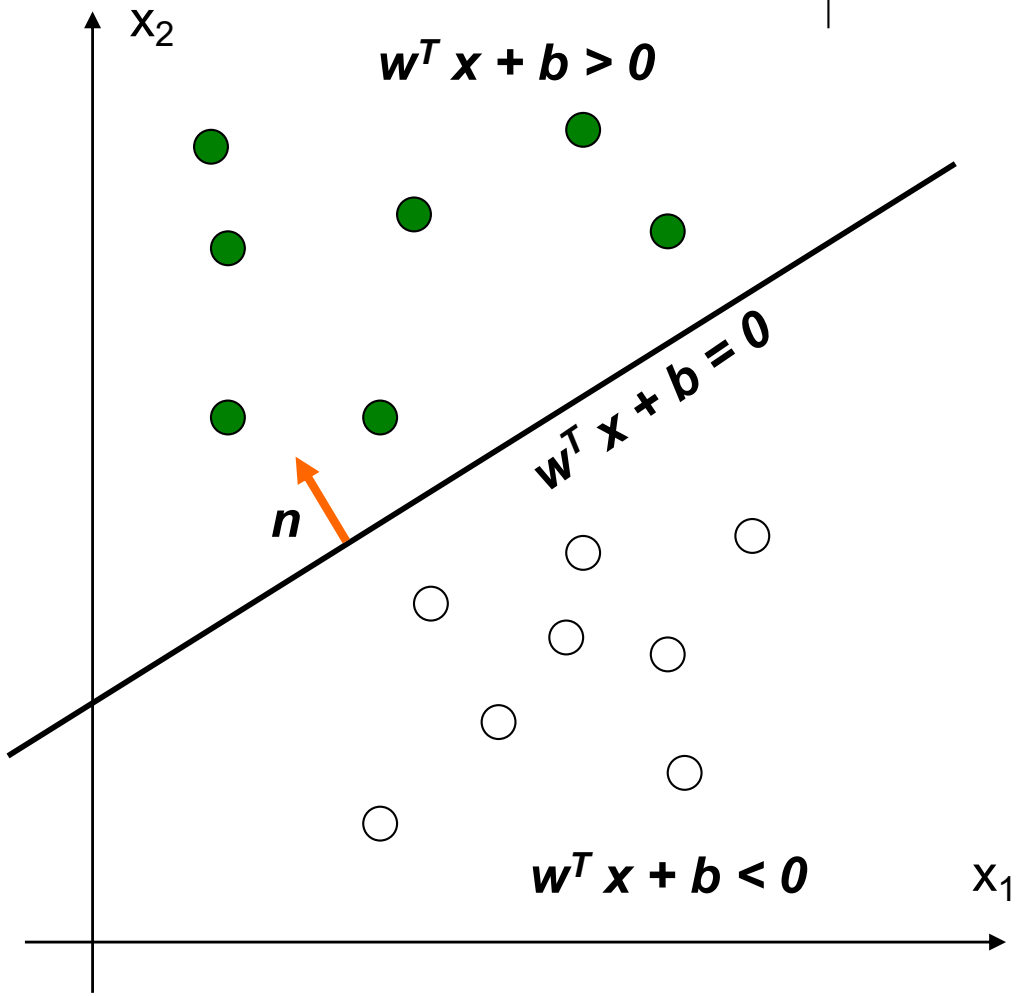
- Η g(x) είναι μια γραμμική συνάρτηση:

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

- Ένα υπερεπίπεδο στο χώρο χαρακτηριστικών

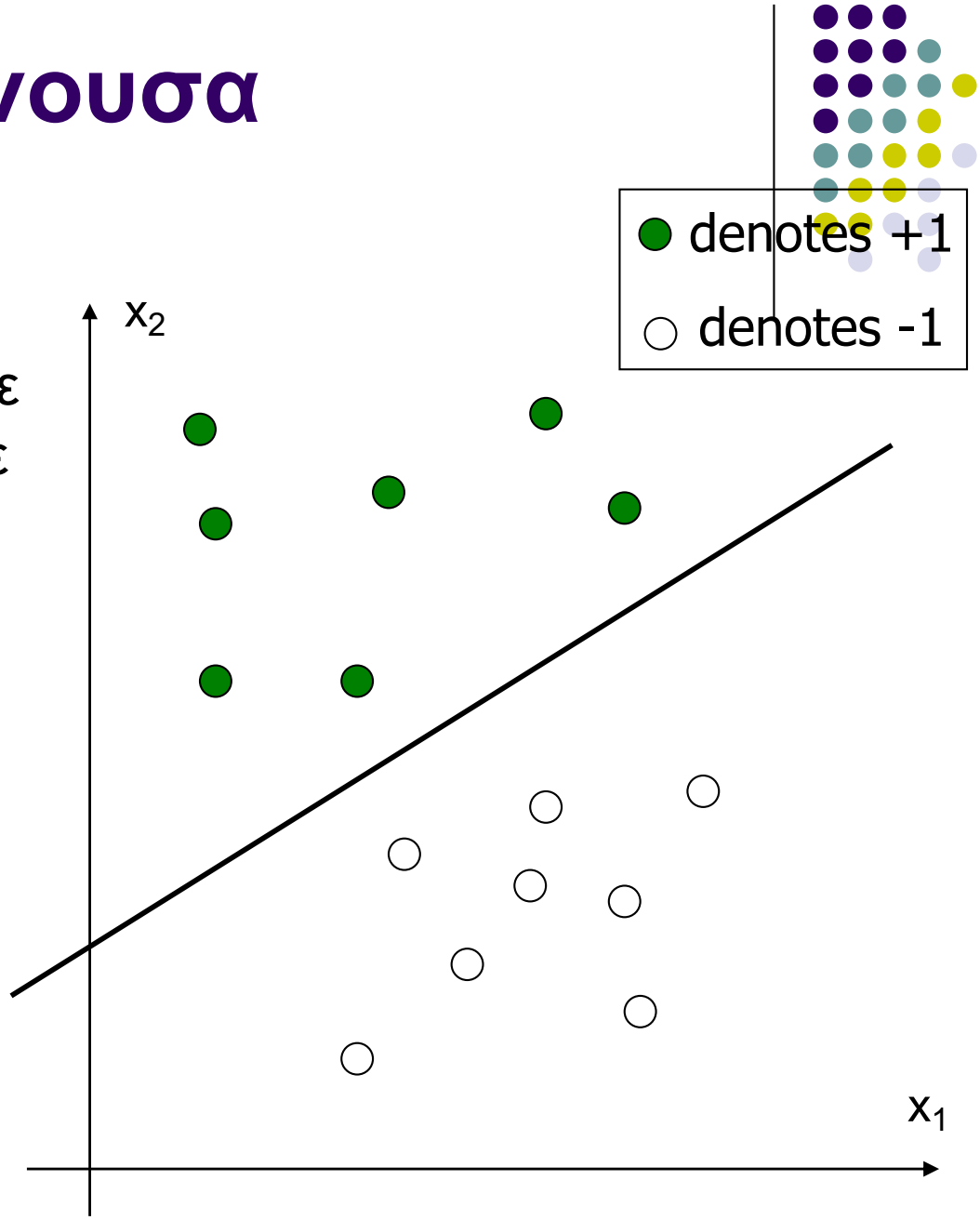
- Ένα (μοναδιαίου μήκους) κανονικοποιημένο διάνυσμα του υπερεπιπέδου:

$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$



Γραμμική διακρίνουσα συνάρτηση

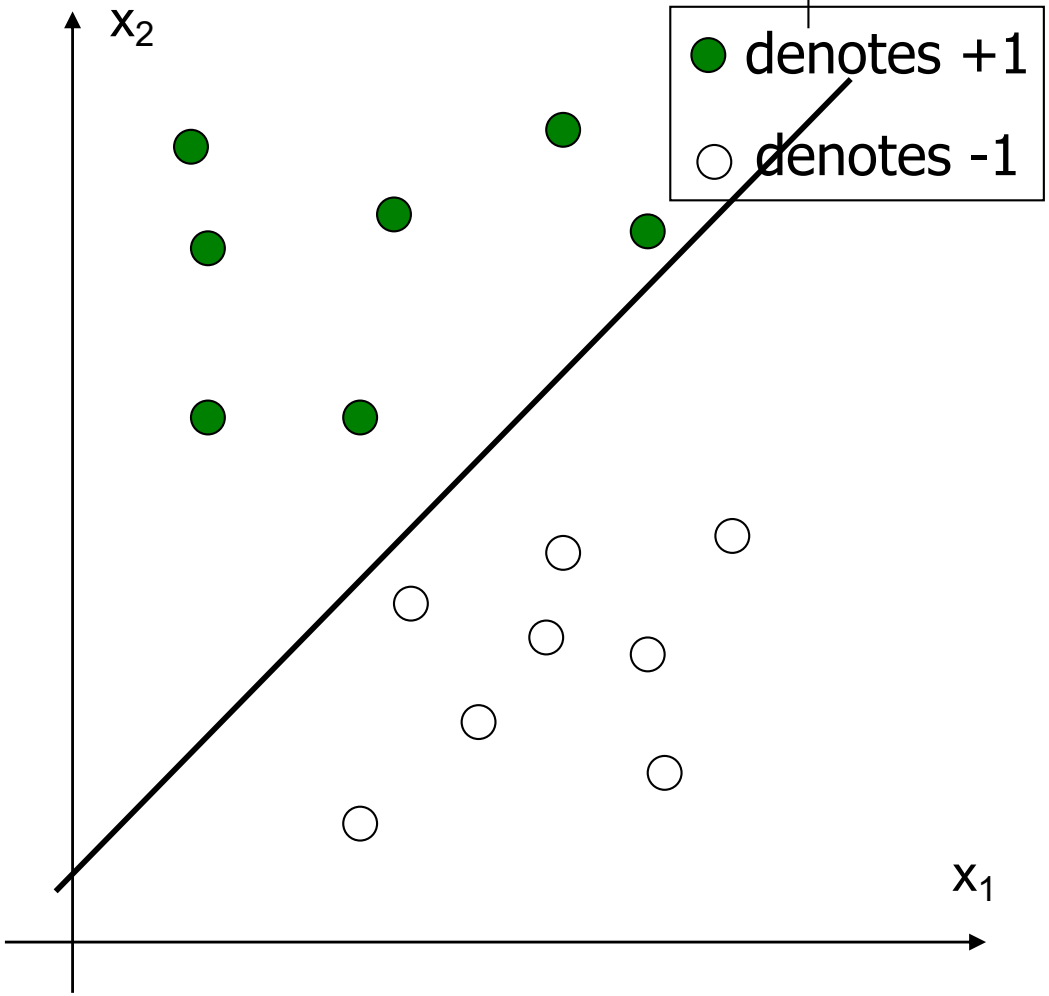
- Πως ορίζατε μια γραμμική διακρίνουσα συνάρτηση, με στόχο να ελαχιστοποιήσετε το σφάλμα;



- Υπάρχει ένας άπειρος αριθμός απαντήσεων !

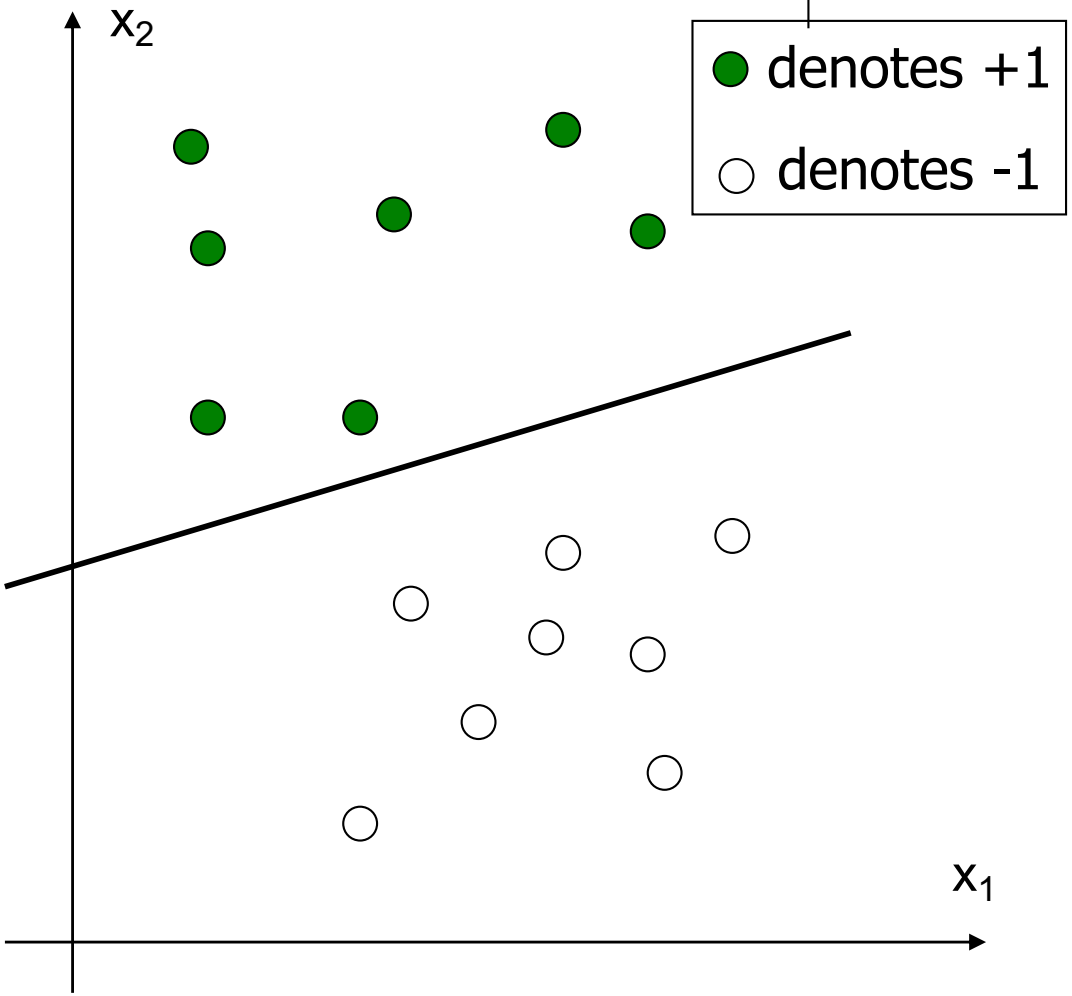
Linear Discriminant Function

- Μια 2η λύση



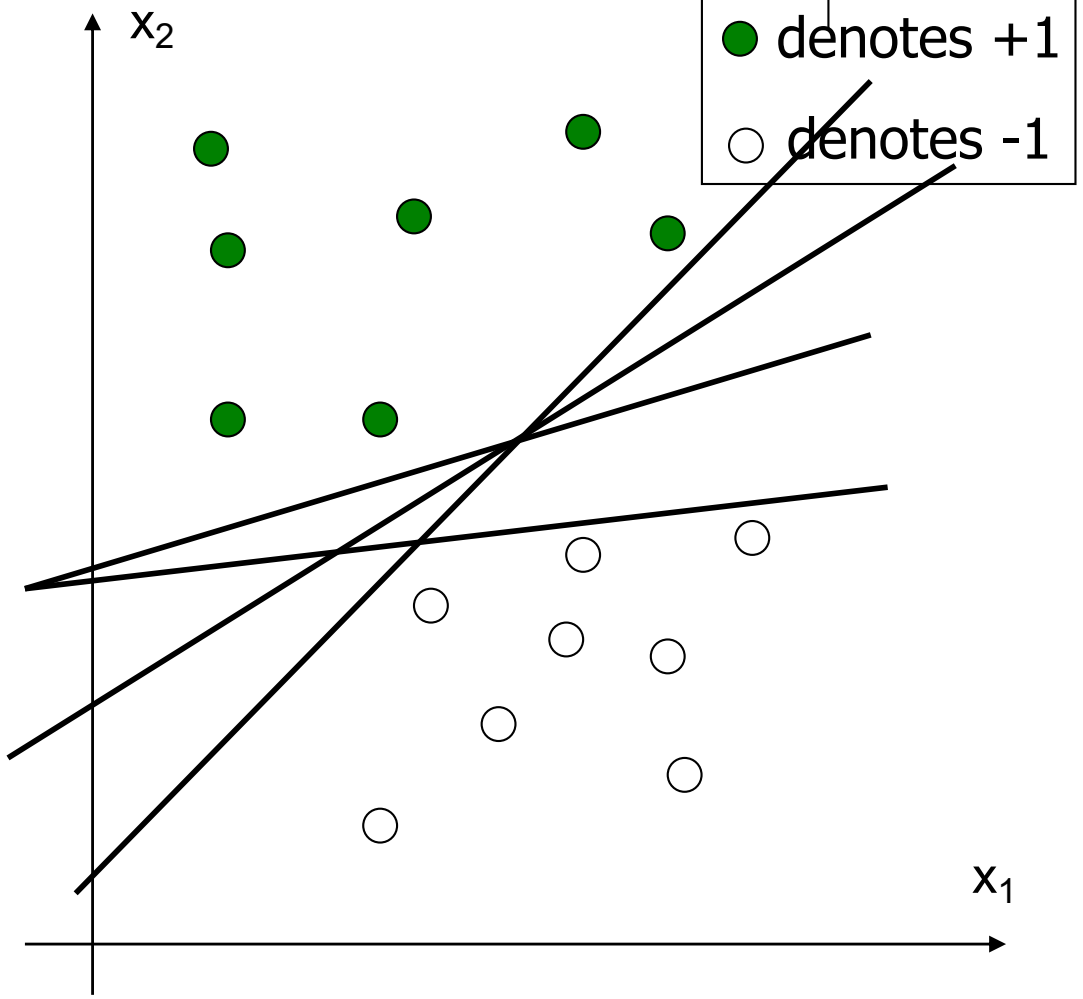
Linear Discriminant Function

- Μια 3η λύση



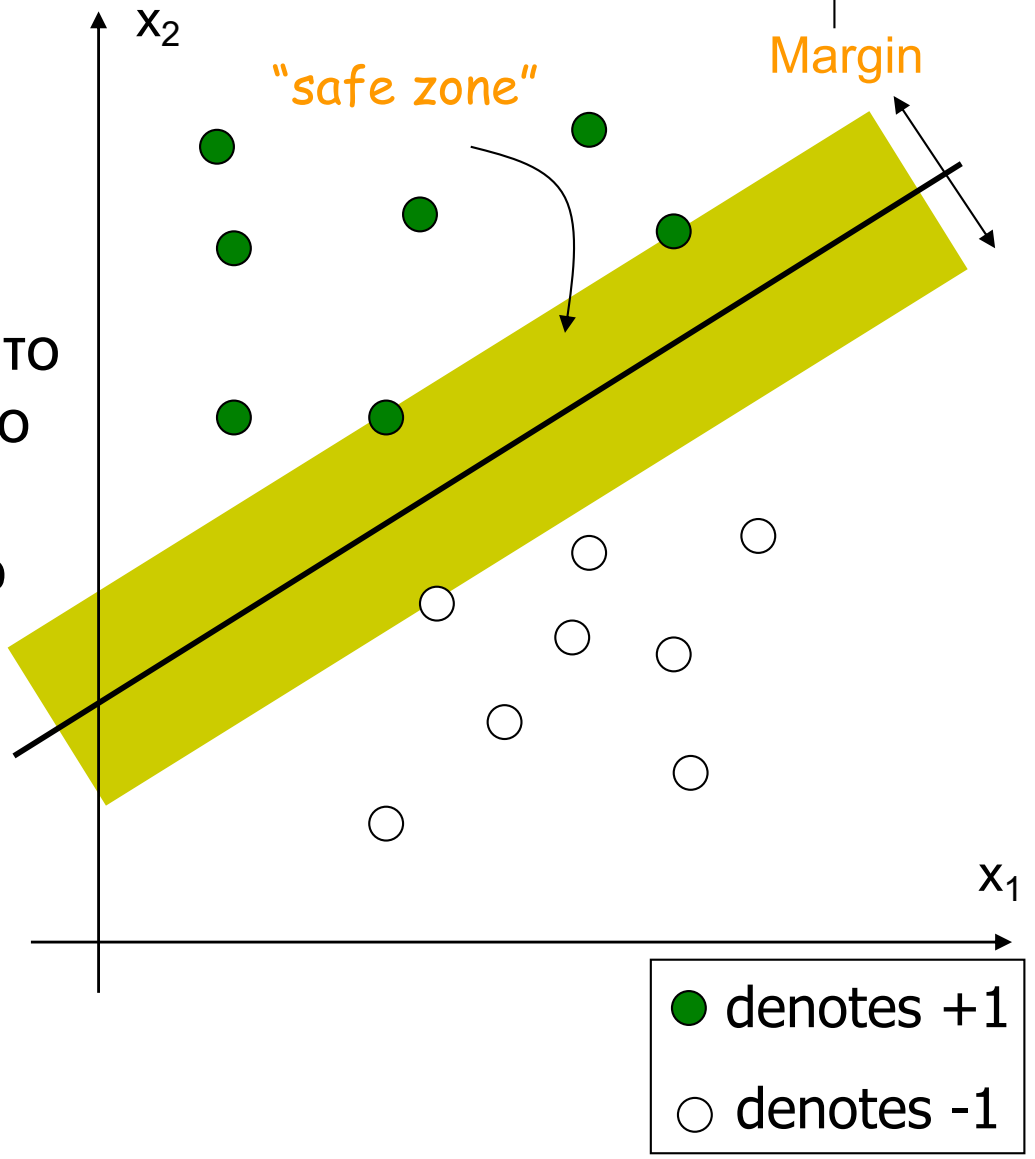
Linear Discriminant Function

- Ποια από τις λύσεις του σχήματος είναι η καλύτερη;



Γραμμικός Ταξινομητής Μεγάλου Περιθωρίου

- Ο γραμμικός ταξινομητής διακρίνουσας συνάρτησης με το μέγιστο περιθώριο είναι ο καλύτερος
- Το περιθώριο ορίζεται σαν το εύρος κατά το οποίο το όριο θα μπορούσε να αυξηθεί πριν “χτυπήσει” ένα σημείο δεδομένων
- Γιατί είναι το καλύτερο;
  - Εύρωστο σε ακραίες τιμές και έτσι έχει ισχυρή δυνατότητα γενίκευσης



Γραμμικός Ταξινομητής Μεγάλου Περιθωρίου

- Έστω ένα σύνολο δεδομένων:

$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n,$ όπου:

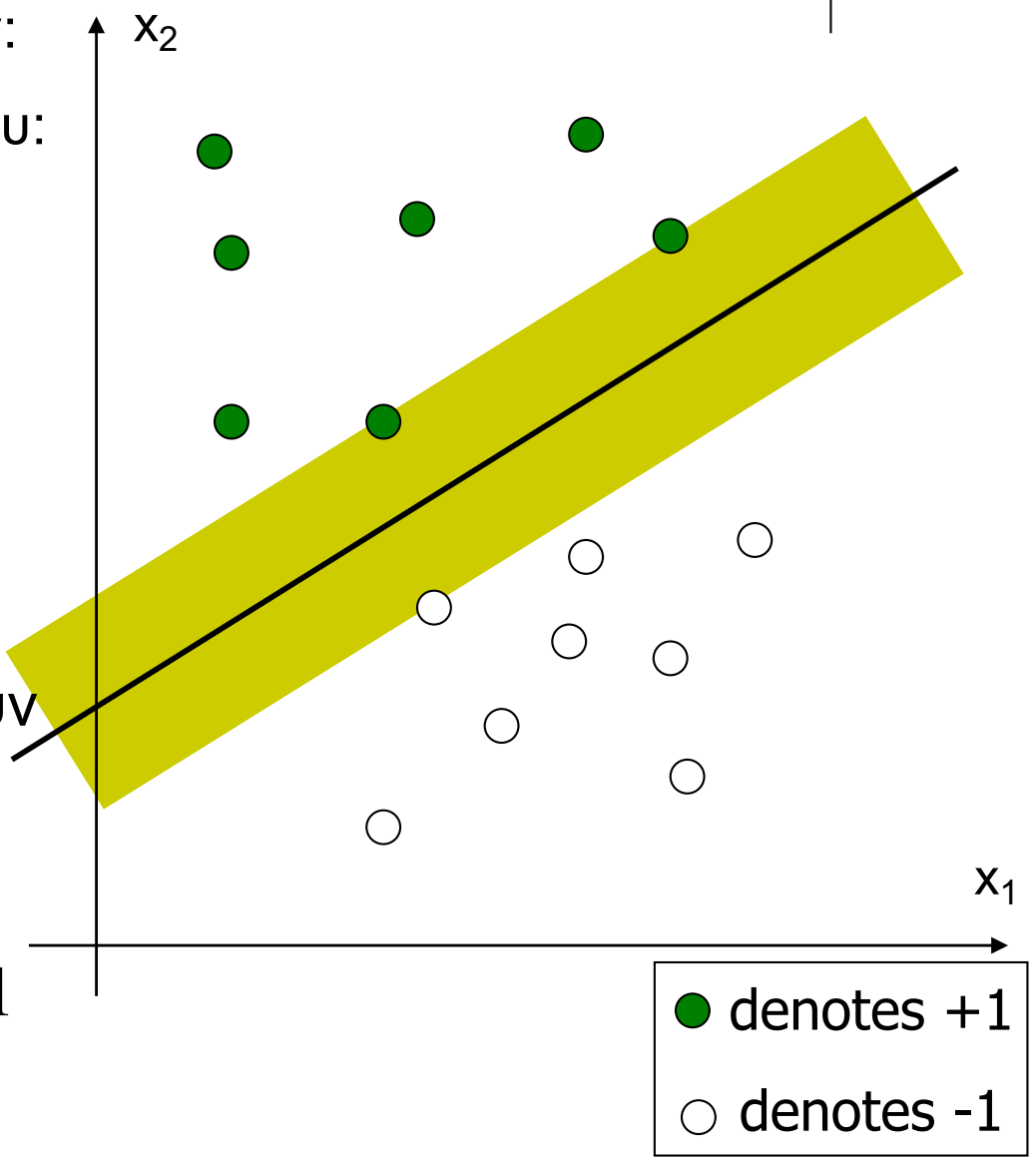
$$\text{For } y_i = +1,\ \ \mathbf{w}^T \mathbf{x}_i + b > 0$$

$$\text{For } y_i = -1,\ \ \mathbf{w}^T \mathbf{x}_i + b < 0$$

- Θεωρώντας περιθώριο οι παραπάνω εξισώσεις μπορούν να τροποποιηθούν σε:

$$\text{For } y_i = +1,\ \ \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1,\ \ \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



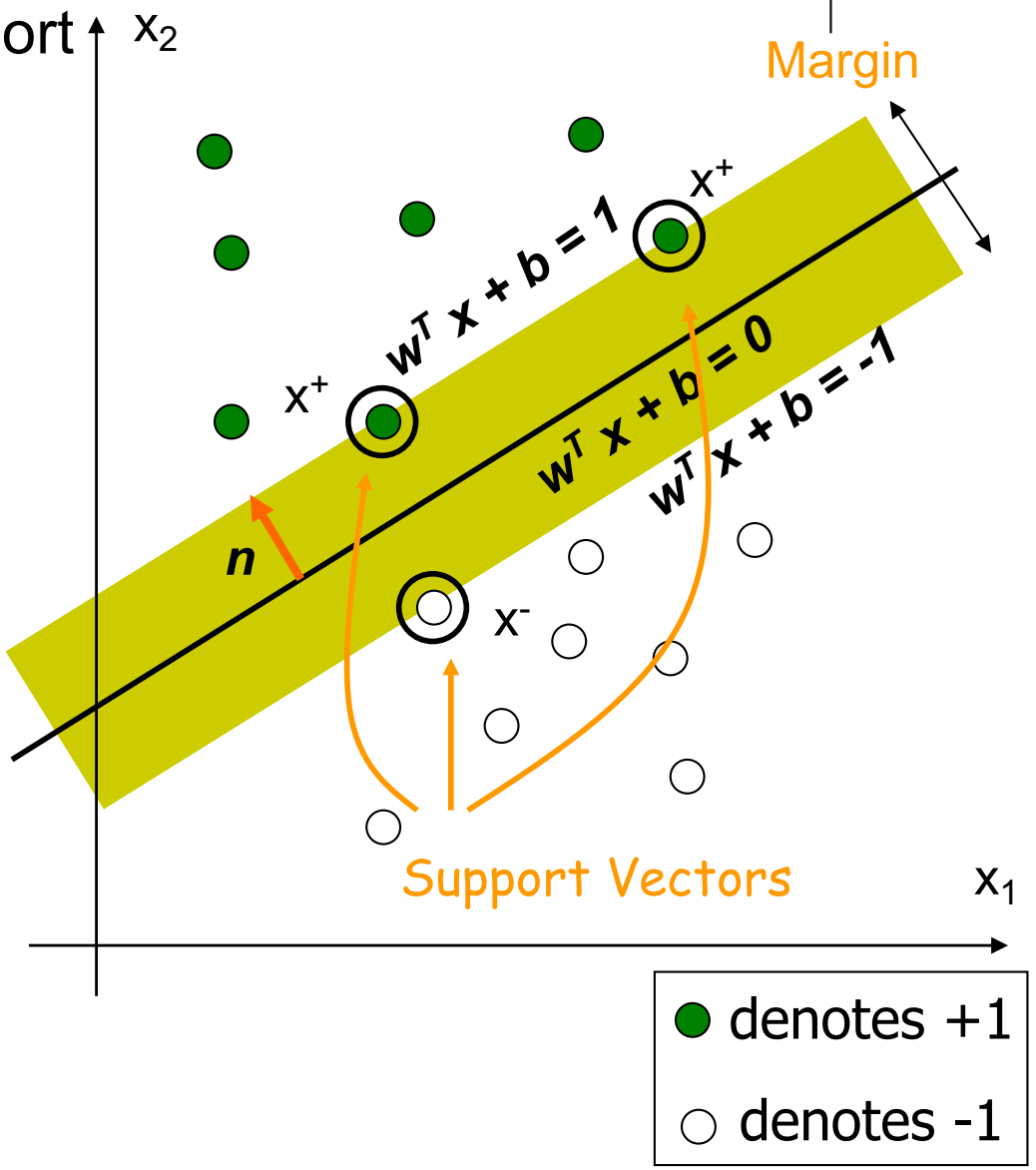
Γραμμικός Ταξινομητής Μεγάλου Περιθωρίου

- Γνωρίζουμε ότι για τα support vectors:

$$\mathbf{w}^T\mathbf{x}^+ + b = 1$$
$$\mathbf{w}^T\mathbf{x}^- + b = -1 \qquad (1)$$

- Το εύρος του περιθωρίου θα είναι:

$$M = (\mathbf{x}^+ - \mathbf{x}^-)^T \cdot \mathbf{n}$$
$$= (\mathbf{x}^+ - \mathbf{x}^-)^T \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$



Γραμμικός Ταξινομητής Μεγάλου Περιθωρίου

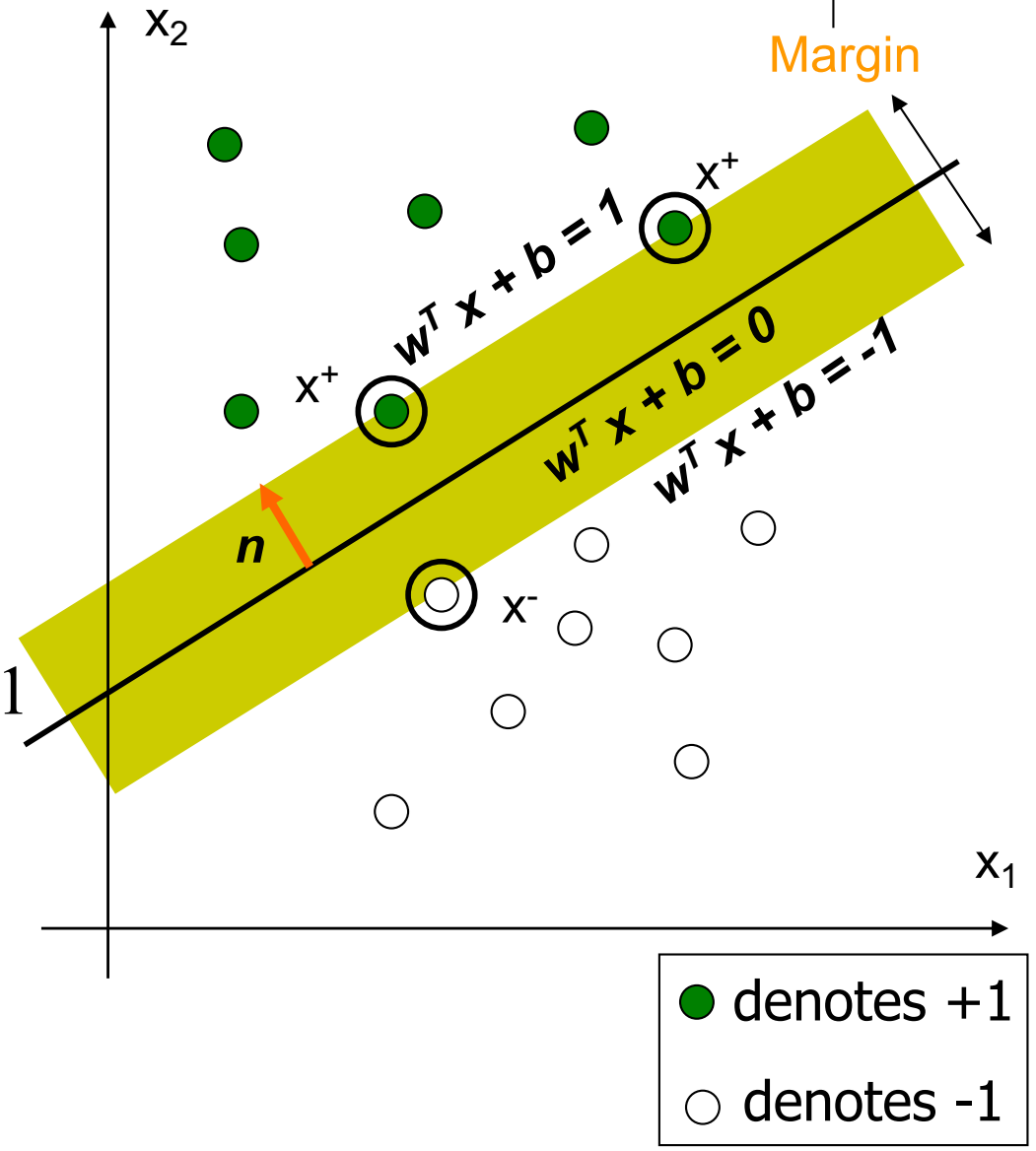
- Διατύπωση:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

Έτσι ώστε:

$$\text{For } y_i = +1, \quad \mathbf{w}^T \mathbf{x}_i + b \geq 1$$

$$\text{For } y_i = -1, \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1$$



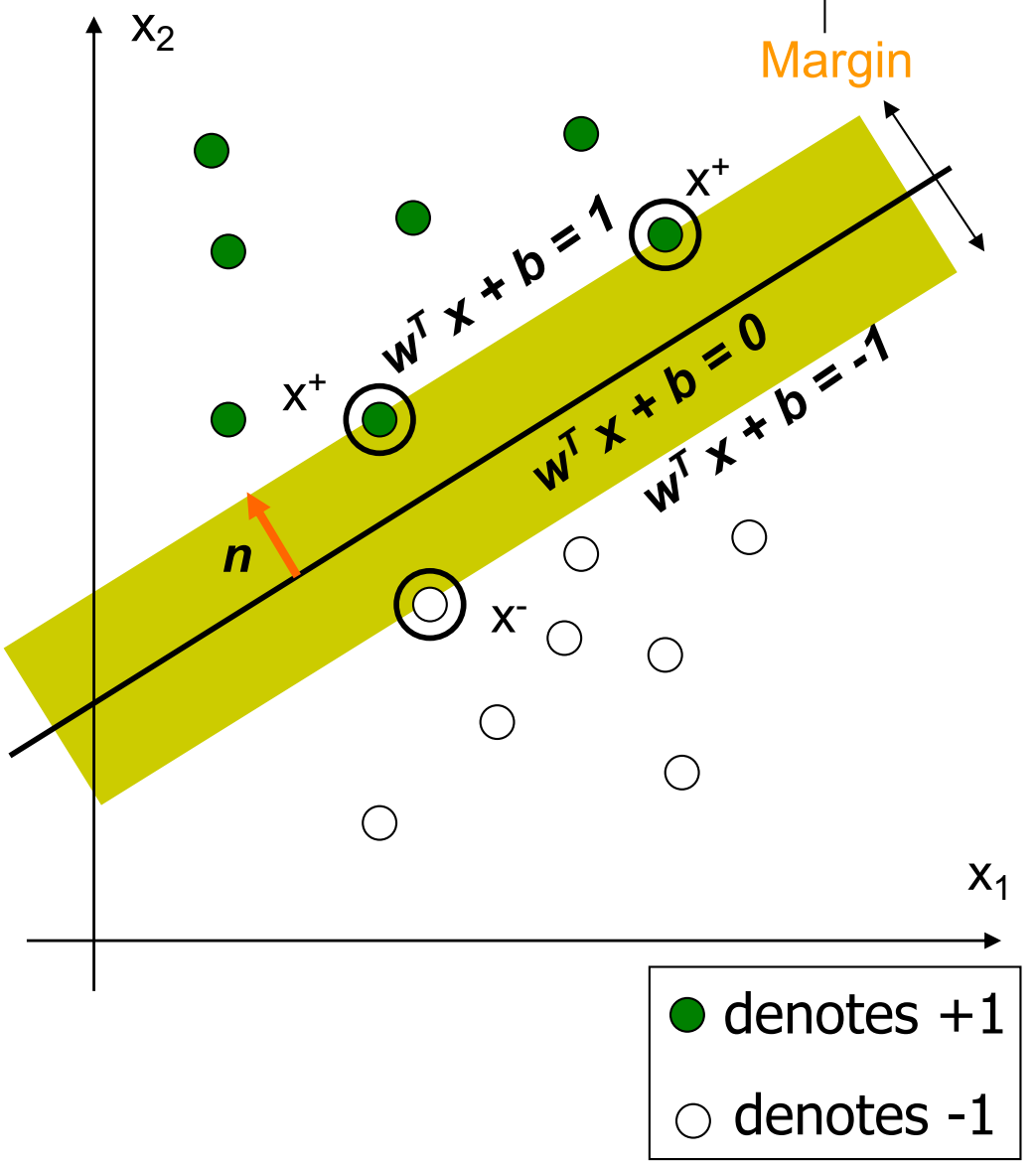
Γραμμικός Ταξινομητής Μεγάλου Περιθωρίου

- Διατύπωση:

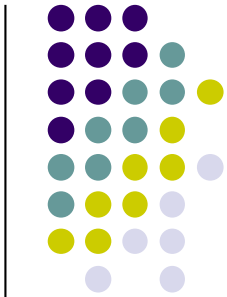
$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

Έτσι ώστε:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$



Επίλυση Προβλήματος Βελτιστοποίησης



Τετραγωνικός προγραμματισμός με γραμμικούς περιορισμούς

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

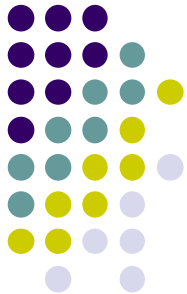
$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Lagrangian συνάρτηση

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right)$$

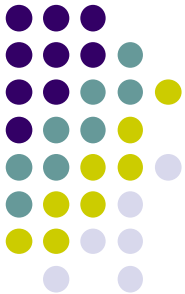
$$\text{s.t.} \quad \alpha_i \geq 0$$

Πολλαπλασιαστές Lagnrange



- Πρόβλημα βελτιστοποίησης με περιορισμούς
- Μετασχηματισμός σε πρόβλημα βελτιστοποίησης χωρίς περιορισμούς

Επίλυση Προβλήματος Βελτιστοποίησης



$$\text{minimize } L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

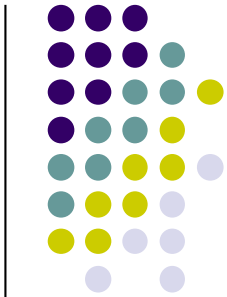
$$\text{s.t.} \quad \alpha_i \geq 0$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Γραμμικός συνδυασμός δειγμάτων

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

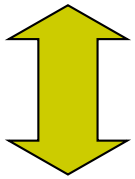
Solving the Optimization Problem



$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

Δυαδικό Lagrangian πρόβλημα



$$\text{maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

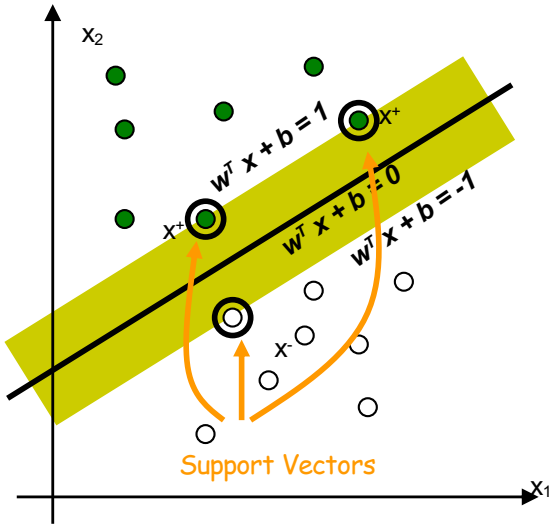
$$\text{s.t.} \quad \alpha_i \geq 0 \text{ , and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Επίλυση Προβλήματος Βελτιστοποίησης

- Από τη συνθήκη KKT *(Karush-Kuhn-Tucker)*, γνωρίζουμε ότι:

$$\alpha_i \left(y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right) = 0$$

- Έτσι, μόνο για τα διανύσματα υποστήριξης ισχύει: $\alpha_i \neq 0$



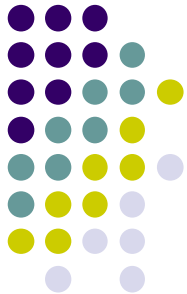
- Η λύση έχει τη μορφή:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i$$

$$\text{get } b \text{ from } y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0,$$

$$\text{where } \mathbf{x}_i \text{ is support vector}$$

Επίλυση Προβλήματος Βελτιστοποίησης



- Η γραμμική διακρίνουσα συνάρτηση είναι:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i \in \mathrm{SV}} \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$

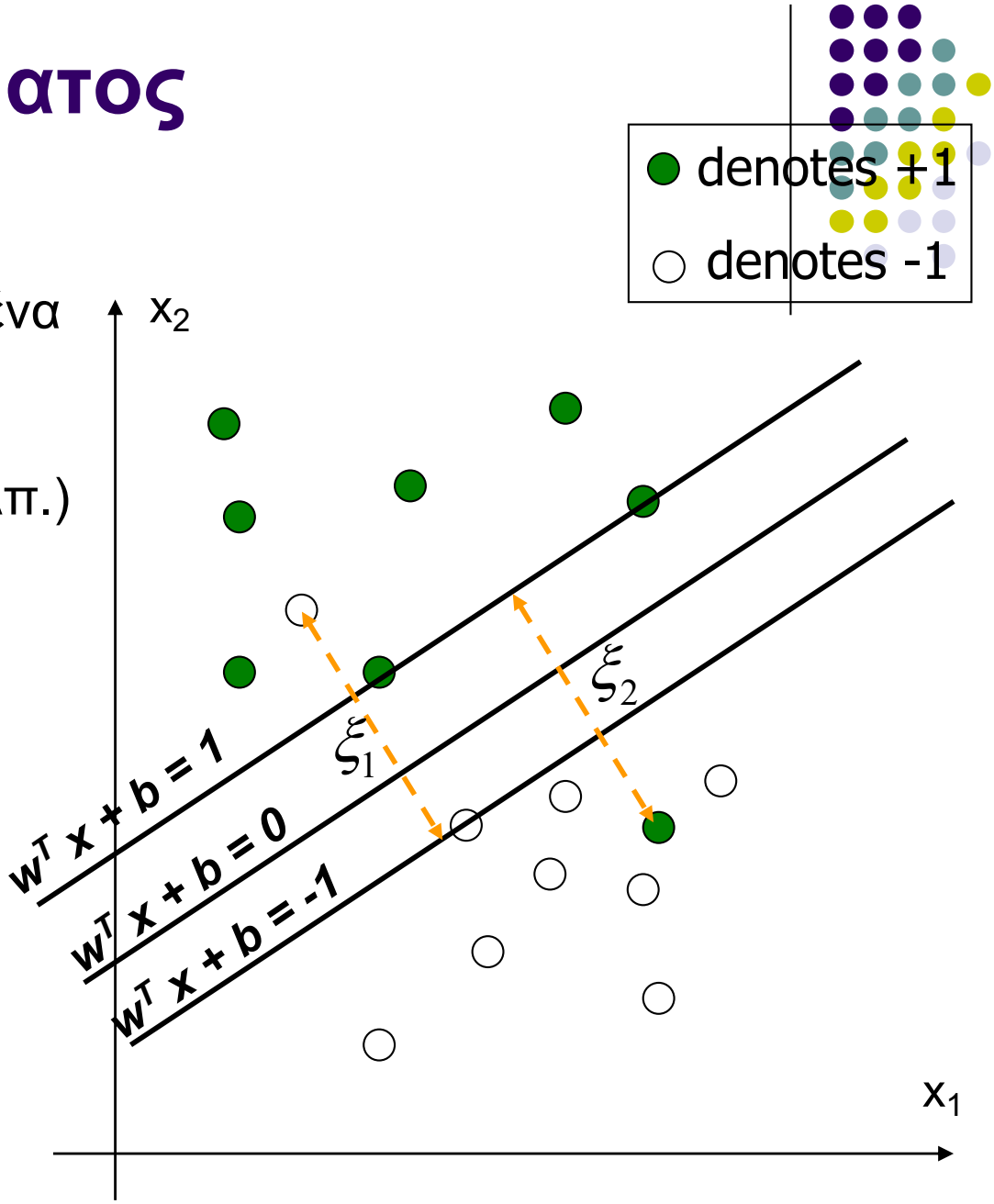
- Σημειώστε ότι βασίζεται σε ένα *dot product* μεταξύ του δείγματος ***x*** και των διανυσμάτων υποστήριξης $\boldsymbol{x_i}$.
- Είναι βαθμωτό μέγεθος ανεξάρτητα από τη διάσταση του ***x***

- Επίσης, να θυμάστε ότι η επίλυση του Π.Β. περιλαμβάνει υπολογισμό των dot products $\boldsymbol{x_i^T x_j}$ μεταξύ όλων των ζευγών των σημείων εκπαίδευσης

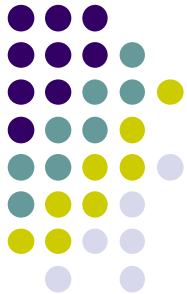
Επίλυση Προβλήματος Βελτιστοποίησης

- Τι γίνεται όμως αν τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα; (θορυβώδη δεδομένα, ακραία σημεία, κλπ.)

- *Μπορούν να προστεθούν μεταβλητές χαλάρωσης* ξ_i για να επιτρέπουν τη μη-κατηγοριοποίηση δύσκολων ή θορυβωδών σημείων δεδομένων



Επίλυση Προβλήματος Βελτιστοποίησης



- Διατύπωση:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

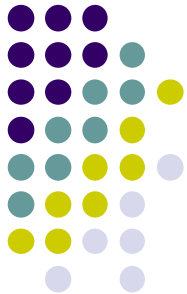
Έτσι ώστε:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- Η παράμετρος *C* μπορεί να ειδωθεί σαν ένα τρόπος να ελέγξουμε το υπερ-ταίριασμα (υπερεκπαίδευση).

Επίλυση Προβλήματος Βελτιστοποίησης



- Διατύπωση: (Lagrangian Dual Problem)

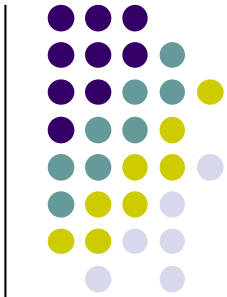
$$\text{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

Έτσι ώστε:

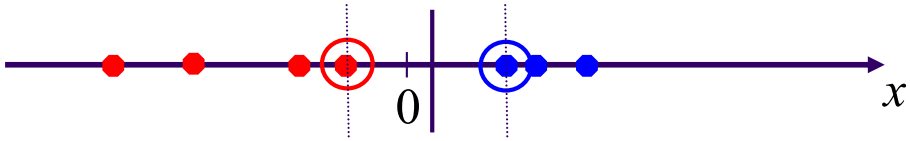
$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

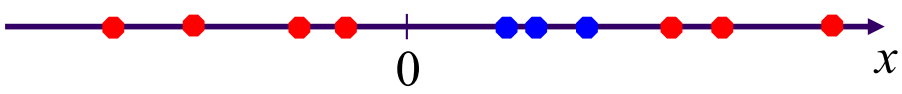
Μη - Γραμμικά SVMs



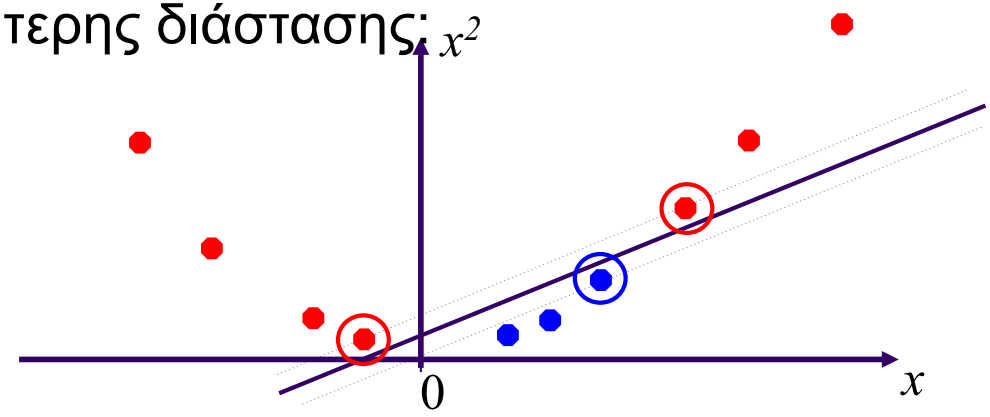
- Σύνολα δεδομένων τα οποία είναι γραμμικά διαχωρίσιμα, δουλεύουν καλά:



- Αλλά τι μπορούμε να κάνουμε αν δεν είναι;



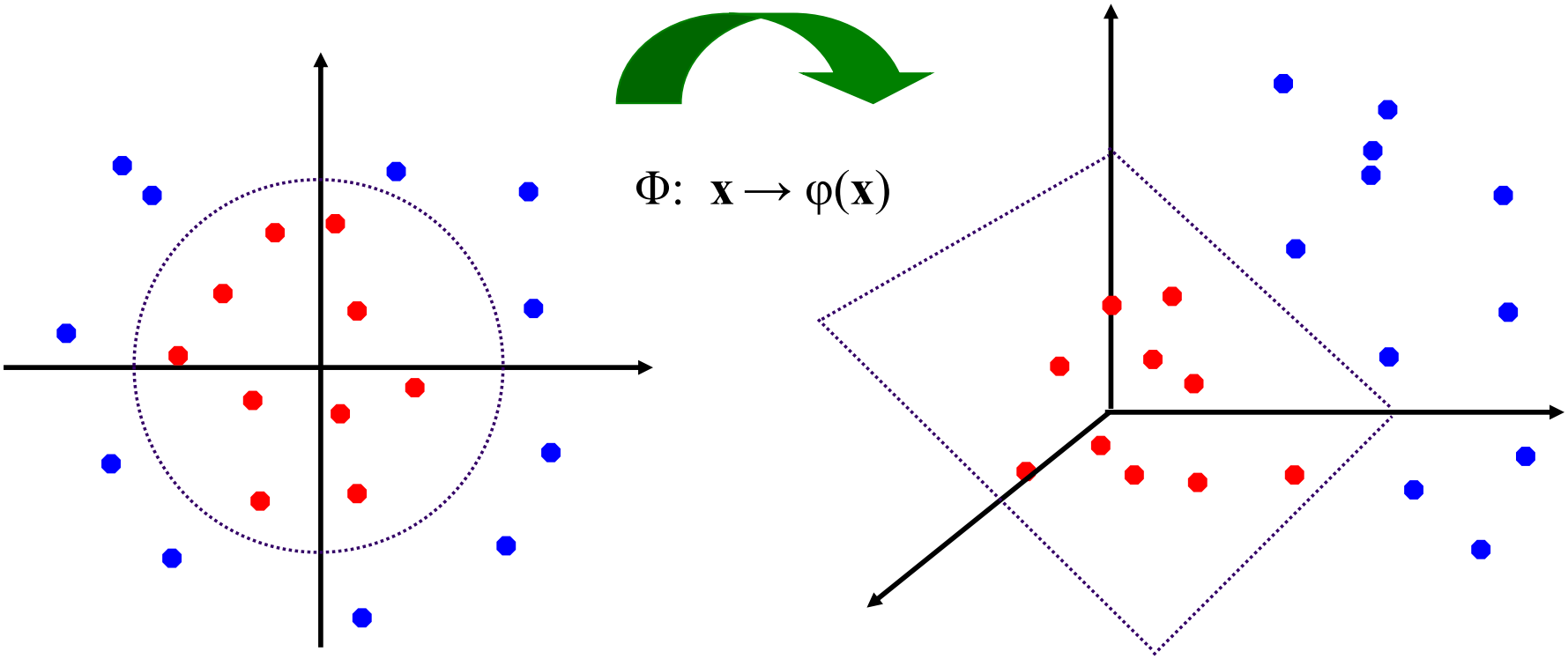
- Ίσως έλυνε το πρόβλημα… η απεικόνιση των δεδομένων σε ένα χώρο μεγαλύτερης διάστασης;



This slide is courtesy of *www.iro.umontreal.ca/~pift6080/documents/papers/**svm**_tutorial.**ppt***

Μη-Γραμμικά SVMs: Χώρος Χαρακτηριστικών

- Γενική ιδέα: ο αρχικός χώρος εισόδου μπορεί να απεικονιστεί σε κάποιο μεγαλύτερης διάστασης χώρο χαρακτηριστικών, όπου το σύνολο εκπαίδευσης είναι διαχωρίσιμο:



This slide is courtesy of *www.iro.umontreal.ca/~pift6080/documents/papers/**svm**_tutorial.**ppt***

Μη-Γραμμικά SVMs: Το "κόλπο" του πυρήνα

- Με αυτή την απεικόνιση, η διακρίνουσα συνάρτηση γίνεται:

$$g(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i \in \mathrm{SV}} \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b$$

- Δεν είναι ανάγκη να ξέρουμε αυτή την απεικόνιση φ με ακρίβεια, γιατί χρησιμοποιούμε μόνο το dot product των διανυσμάτων χαρακτ/κών, τόσο στην εκπαίδευση όσο και στον έλεγχο.

- Μια συνάρτηση πυρήνα (*kernel function)* ορίζεται σαν η συνάρτηση που αντιστοιχεί στο dot product δύο διανυσμάτων χαρακτ/κών, σε κάποιο επαυξημένο χώρο χαρακτ/κών:

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Μη-Γραμμικά SVMs: Το "κόλπο" του πυρήνα

- Ένα παράδειγμα:

2-διάστατα διανύσματα $\mathbf{x}=[x_1 \; x_2]$;

έστω: $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

Πρέπει να δείξουμε ότι: $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2,$$
$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$
$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T \, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$
$$= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j), \quad \text{όπου } \varphi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

This slide is courtesy of *www.iro.umontreal.ca/~pift6080/documents/papers/**svm**_tutorial.**ppt***

Μη-Γραμμικά SVMs: Το "κόλπο" του πυρήνα

- Παραδείγματα από κοινά χρησιμοποιούμενες συναρτήσεις πυρήνα:
  - Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
  - Gaussian (Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$

  - Sigmoid:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$$

Μη-Γραμμικά SVM: Βελτιστοποίηση

- Διατύπωση:Langrangian Dual Problem)

$$\text{maximize} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)$$

Έτσι ώστε:

$$0 \le \alpha_i \le C$$

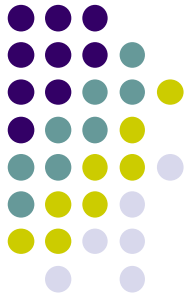
$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

- Η λύση της διακρίνουσας συνάρτησης είναι:

$$g(\mathbf{x}) = \sum_{i\in \mathrm{SV}}\alpha_i K(\mathbf{x}_i,\mathbf{x}) + b$$

- Η τεχνική βελτιστοποίησης είναι η ίδια.

Support Vector Machine: Ο Αλγόριθμος



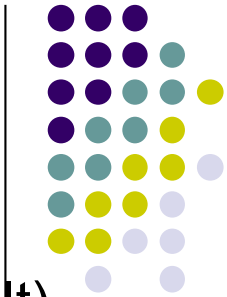
1. Επίλεξε μια συνάρτηση πυρήνα.

2. Επίλεξε μια τιμή για το *C.*

3. Επίλυσε το πρόβλημα τετραγωνικού προγραμματισμού (υπάρχουν διαθέσιμα πολλά πακέτα λογισμικού).

4. Κατασκεύασε τη διακρίνουσα συνάρτηση από τα διανύσματα υποστήριξης.

Μερικά Θέματα

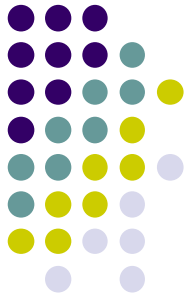


- Επιλογή Πυρήνα:
  - Gaussian ή πολυωνυμικός πυρήνας είναι η προεπιλογή (default).
  - Αν είναι αναποτελεσματικοί, τότε χρειάζονται πιο σύνθετοι πυρήνες.
  - Οι έμπειροι του πεδίου μπορούν να βοηθήσουν στο σχηματισμό κατάλληλων μέτρων ομοιότητας.

- Επιλογή των παραμέτρων των πυρήνων:
  - π.χ. του σ στον Gaussian πυρήνα.
  - το σ είναι η απόσταση μεταξύ των πλησιέστερων σημείων με διαφορετική κατηγοριοποίηση.
  - Όταν λείπουν αξιόπιστα κριτήρια, οι εφαρμογές βασίζονται στη χρήση ενός συνόλου επικύρωσης ή σε διασταυρωμένη επικύρωση για τον καθορισμό αυτών των παραμέτρων.

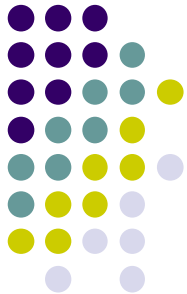
- Κριτήριο Βελτιστοποίησης – Hard margin v.s. Soft margin
  - μια σειρά εκτεταμμένων πειραμάτων στα οποία ελέγχονται διάφορες τιμές των παραμέτρων.

Συμπεράσματα: Support Vector Machine



- 1. Ταξινομητής Μεγάλου Margin:
  - Δυνατότητα καλύτερης γενίκευσης και λιγότερο υπέρ-ταίριασμα (over-fitting).
- 2. Το “Κόλπο” του Πυρήνα:
  - Απεικονίζει τα δεδομένα σε χώρο μεγαλύτερης διάστασης, με στόχο να τα κάνει γραμμικά διαχωρίσιμα.
  - Αφού χρησιμοποιείται μόνο εσωτερικό γινόμενο, δεν χρειάζεται να αναπαραστήσουμε την απεικόνιση με ακρίβεια.

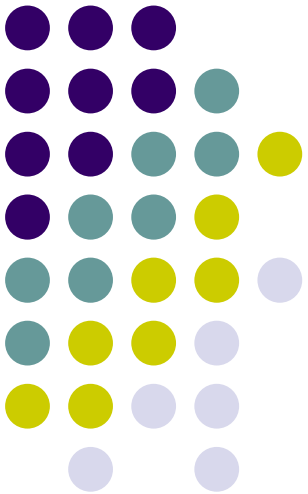
Πρόσθετες Πηγές

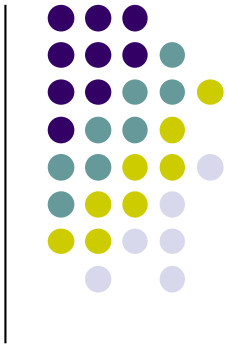


- http://www.kernel-machines.org/

Demo of LibSVM

https://cs.stanford.edu/~karpathy/svmjs/demo/





Lagrangian Duality in brief

The Primal Problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \le 0, \quad i = 1,\ldots,k \\ & h_i(w) = 0, \quad i = 1,\ldots,l \end{aligned}$$

The generalized Lagrangian:

$$L(w,\alpha,\beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

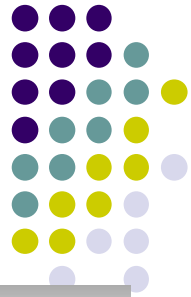
the α's ($\alpha_i \ge 0$) and β's are called the Lagrange multipliers

Lemma:

$$\max_{\alpha,\beta,\alpha_i \ge 0} L(w,\alpha,\beta) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise} \end{cases}$$

A re-written Primal:

$$\min_w \max_{\alpha,\beta,\alpha_i \ge 0} L(w,\alpha,\beta)$$



Lagrangian Duality, cont.

The Primal Problem: $p^* = \min_w \max_{\alpha,\beta,\alpha_i \geq 0} \mathrm{L}(w,\alpha,\beta)$

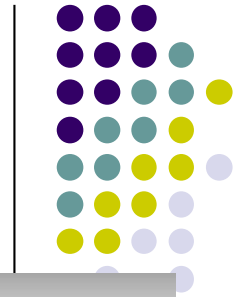
The Dual Problem: $d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_w \mathrm{L}(w,\alpha,\beta)$

Theorem (weak duality):

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_w \mathrm{L}(w,\alpha,\beta) \leq \min_w \max_{\alpha,\beta,\alpha_i \geq 0} \mathrm{L}(w,\alpha,\beta) = p^*$$

Theorem (strong duality):

Iff there exist a saddle point of $L(w,\alpha,\beta)$, we have $d^* = p^*$



The KKT conditions

If there exists some saddle point of *L*, then it satisfies the following "Karush-Kuhn-Tucker" (KKT) conditions:

$$\frac{\partial}{\partial w_i} \mathrm{L}(w, \alpha, \beta) = 0, \quad i = 1, \ldots, k$$

$$\frac{\partial}{\partial \beta_i} \mathrm{L}(w, \alpha, \beta) = 0, \quad i = 1, \ldots, l$$

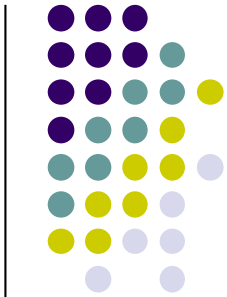
$$\alpha_i g_i(w) = 0, \quad i = 1, \ldots, m$$

$$g_i(w) \leq 0, \quad i = 1, \ldots, m$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, m$$

Theorem: If *w**, *a** and *b** satisfy the KKT condition, then it is also a solution to the primal and the dual problems.

Τα SVM στην python



```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC()
```

Hide prompts and outputs

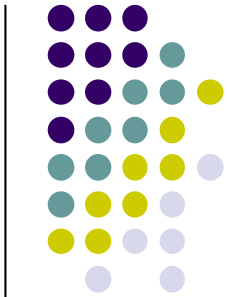
After being fitted, the model can then be used to predict new values:

```
>>> clf.predict([[2., 2.]])
array([1])
```

SVMs decision function (detailed in the Mathematical formulation) depends on some subset of the training data, called the support vectors. Some properties of these support vectors can be found in attributes `support_vectors_`, `support_` and `n_support_`:

```
>>> # get support vectors
>>> clf.support_vectors_
array([[0., 0.],
       [1., 1.]])
```

Τα SVM στην python



```
>>> linear_svc = svm.SVC(kernel='linear')
>>> linear_svc.kernel
'linear'
>>> rbf_svc = svm.SVC(kernel='rbf')
>>> rbf_svc.kernel
'rbf'
```

Περισσότερα:

https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8