Αναγνώριση Προτύπων - Μηχανική Μάθηση

Μηχανές Διανυσμάτων Υποστήριξης - Support vector machines

Γιάννης Παναγάκης

Το πρόβλημα της ταξινόμησης

— Στόχος είναι δοθέντος ενός συνόλου παρατηρήσεων (σημάτων, δεδομένων) να μάθουμε μια συνάρτηση η οποία εκχωρεί σε κάθε δεδομένο X μια κατηγορία (category or class) Y.

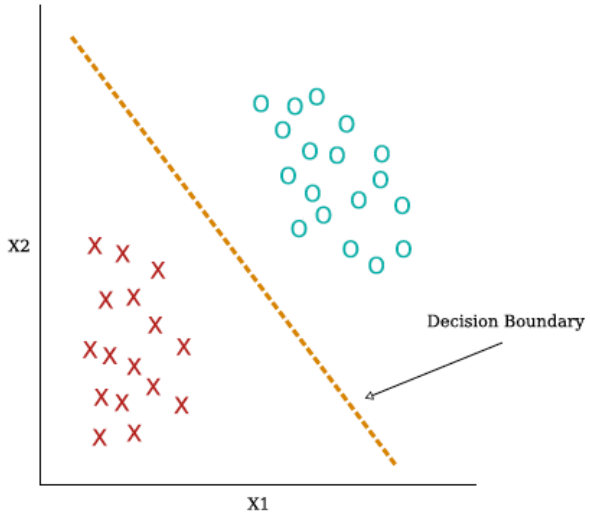
Σύνολο εκπαίδευσης: $\mathscr{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \sim p$

Δεδομένα εισόδου		**Μεταβλητές στόχου**
Χαρακτηριστικά (features)	f	Labels
$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}, \mathbf{x}_i \in X \subseteq \mathbb{R}^D$	→	$\{y_1, y_2, \cdots, y_N\}, y_i \in Y \subseteq \mathbb{Z}$

— Σε αυτό το πλαίσιο, η συνάρτηση f(.) ονομάζεται ταξινομητής (classifier)

Γραμμικός ταξινομητής: 2 κλάσεις

— Εάν τα δεδομένα X διαχωρίζονται γραμμικά σε 2 κλάσεις τότε ο ταξινομητής αποτελεί ένα υπερεπίπεδο που ορίζει το σύνορο απόφασης (decision boundary).



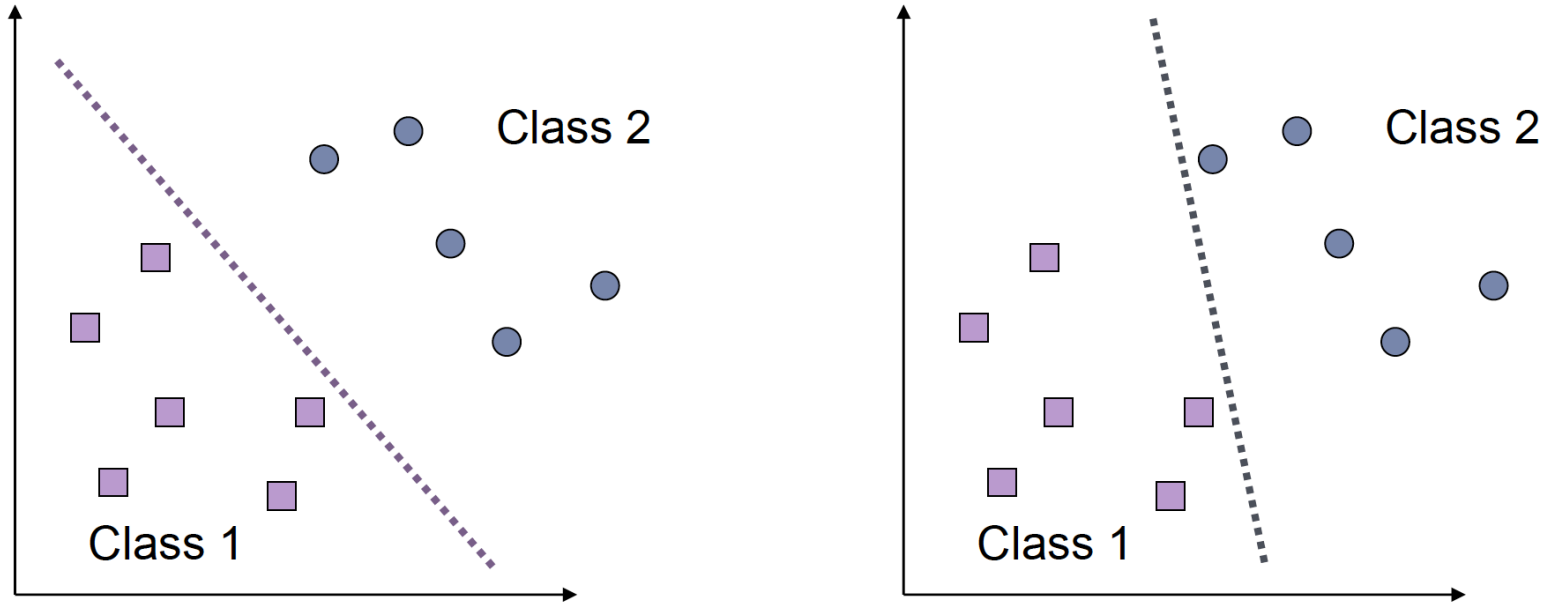
$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$$

—f(x) > 0 για τα δεδομένα που ανήκουν στη κατηγορία “κύκλος”

—f(x) < 0 για τα δεδομένα που ανήκουν στη κατηγορία “τετράγωνο”

Κίνητρο

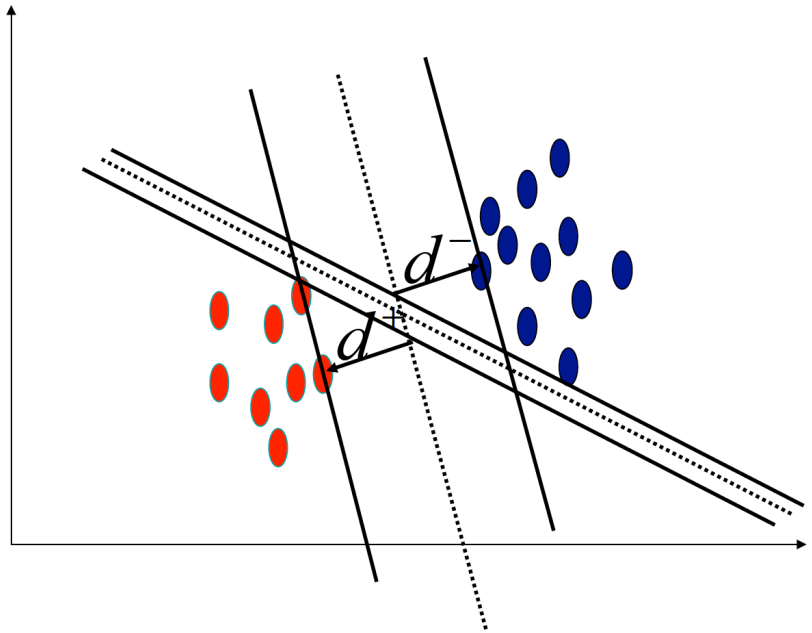
— Στο πρόβλημα ταξινόμησης με 2 κατηγορίες (κλάσεις), οι οποίες είναι γραμμικά διαχωρίσιμες μπορούμε να έχουμε πολλούς γραμμικούς ταξινομητές που να χωρίζουν τα δύο σύνολα



— Ποιος είναι ο καλύτερος ταξινομητής;

Κεντρική ιδέα

— Το σύνορο (decision boundary) που διαχωρίζει τις κλασεις (κατηγορίες), πρέπει να είναι όσο το δυνατόν πιο μακρυά από τα δεδομένα εκπαίδευσης.



— Πώς θα βρούμε το υπερεπίπεδο το οποίο διαχωρίζει τις κλάσεις (κατηγορίες) και είναι όσο το δυνατόν πιο μακρυά από τα δεδομένα;

Γραμμικός ταξινομητής: υπενθύμιση

— Ένας γραμμικός ταξινομητής ορίζεται ως υπερεπίπεδο στις d διαστάσεις.

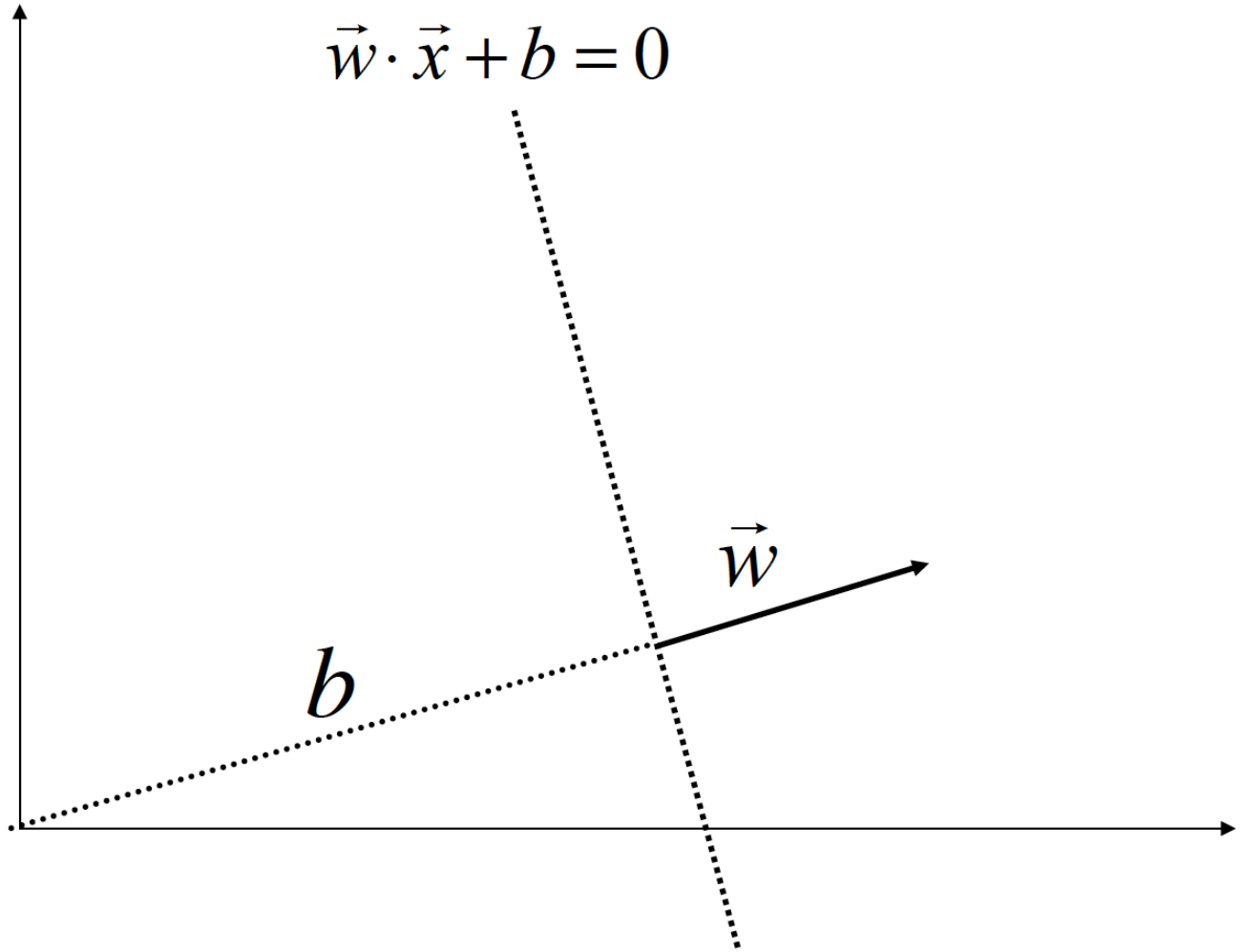
— Οποιοδήποτε υπερεπίπεδο εκφράζεται ως το σύνολο των σημείων x τα οποία ικανοποιούν την παρακάτω εξίσωση

$$< \mathbf{w}, \mathbf{x} > + b = \mathbf{w}^T \mathbf{x} + b = 0$$

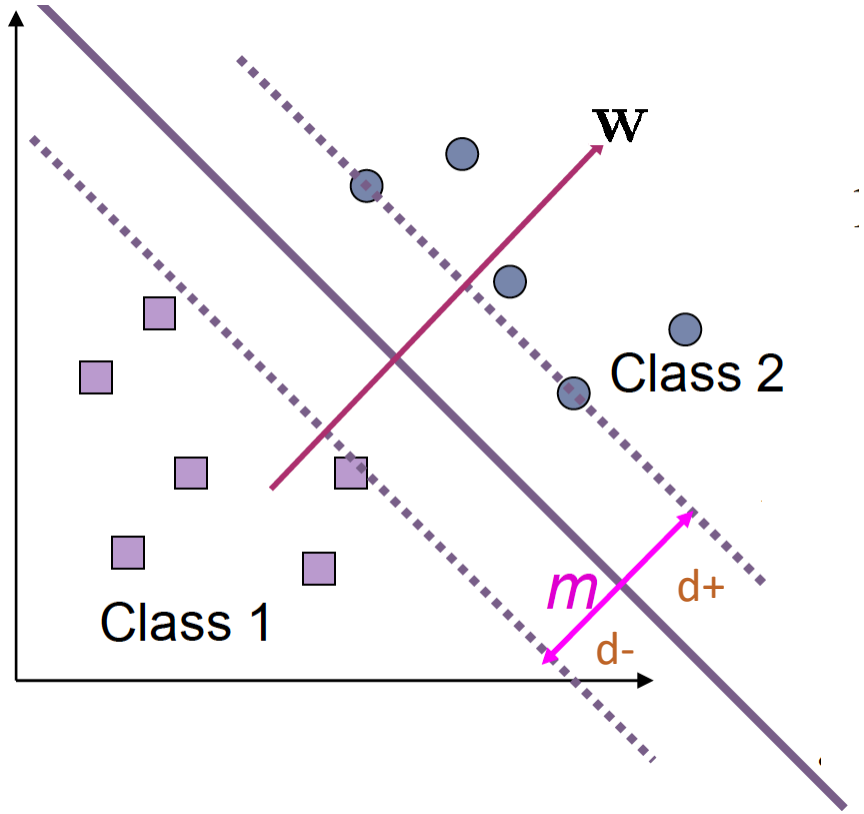
— Το διάνυσμα **w** είναι κανονικό διάνυσμα (normal vector), δηλαδή είναι κάθετο στο υπερεπιπέδο. Η παράμετρος b καθορίζει τη μετατόπιση του υπερεπιπέδου από την αρχή των αξόνων κατά μήκος του διανύσματος w.

$$\text{dist. to origin } = \frac{|b|}{\|w\|_2}$$

Παράδειγμα στις 2 διαστάσεις



Η έννοια του περιθωρίου



— Έστω d+ η απόσταση του πλησιέστερου στο υπερεπίπεδο δεδομένου που ανήκει στη κατηγορία “κύκλος”.

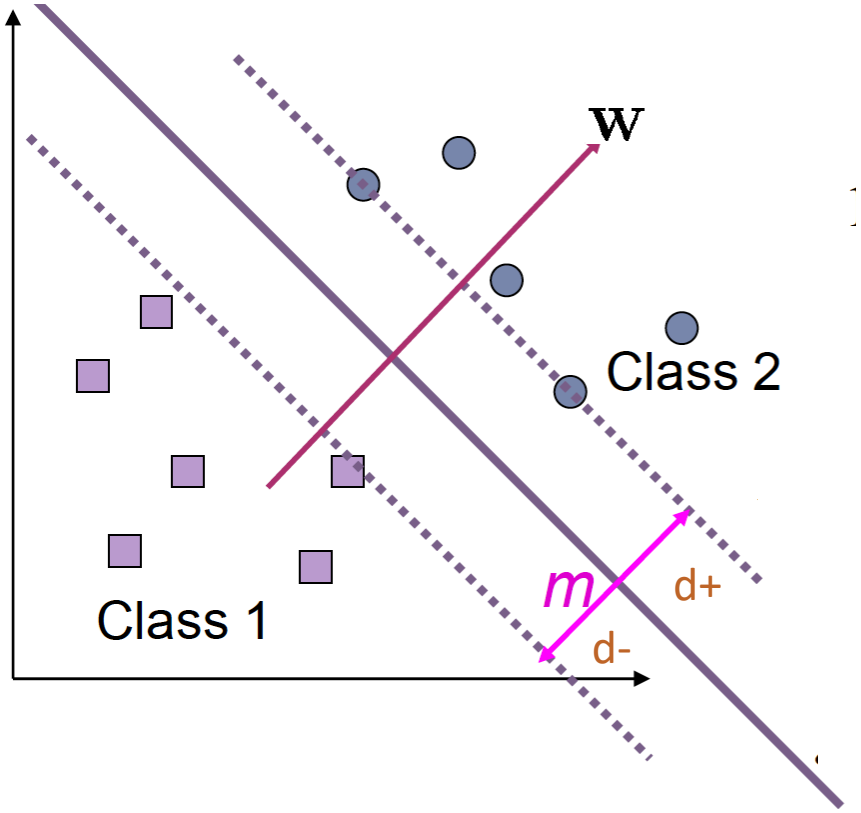
— Έστω d- η απόσταση του πλησιέστερου στο υπερεπίπεδο δεδομένου που ανήκει στη κατηγορία “τετράγωνο”.

— Ως περιθώριο (margin) ορίζεται η απόσταση

$$m = d^+ + d^-$$

— Στόχος μας είναι να εκπαιδεύσουμε τον ταξινομητή f(x;w) για τον οποίο το margin είναι μέγιστο.

Η έννοια των διανυσμάτων στήριξης (support vectors)



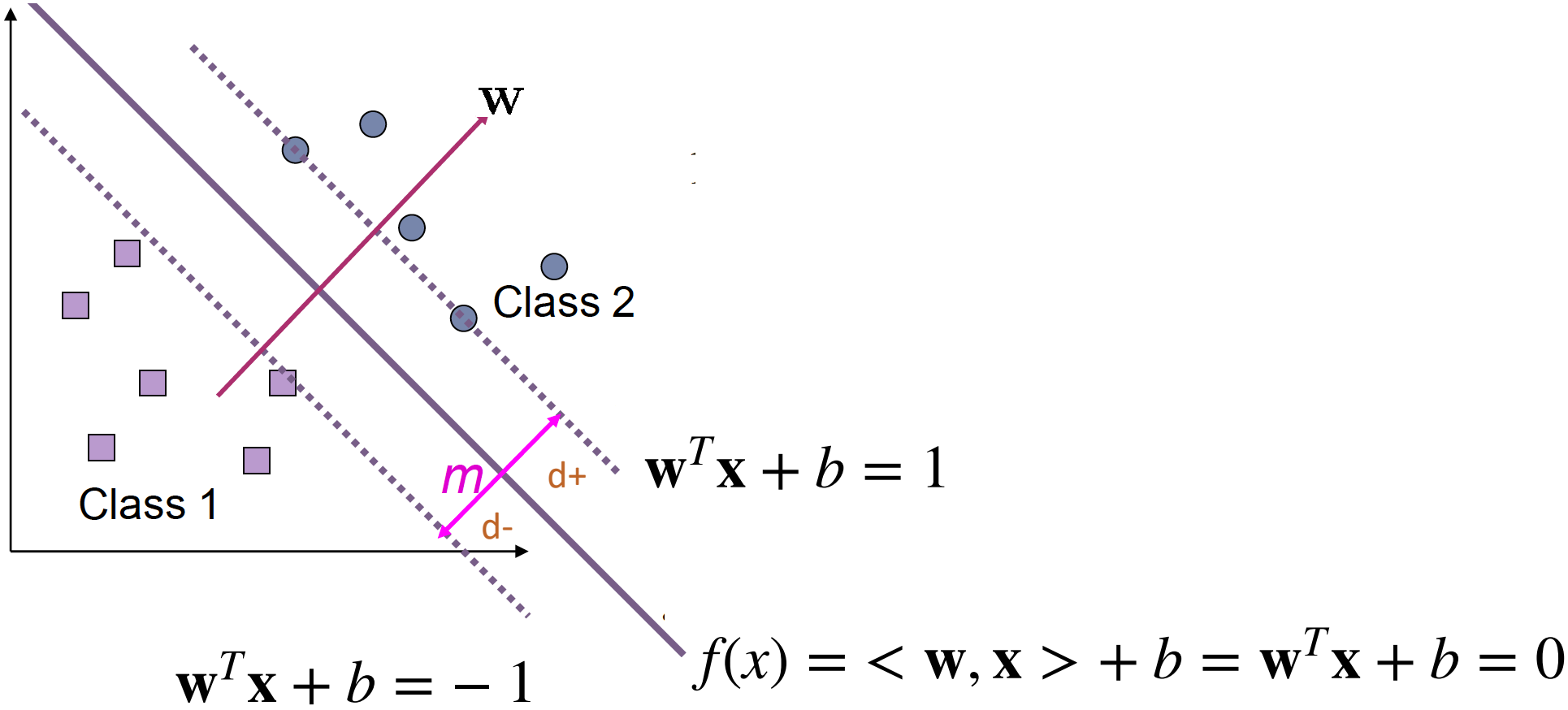
— Υπάρχουν w και b για τα οποία ισχύει

$$d^+ = d^- = \frac{1}{\|w\|_2}$$

— Τα δεδομένα που βρίσκονται σε απόσταση $1/||w||_2$ από την επιφάνεια απόφασης ονομάζονται διανύσματα στήριξης (support vectors).

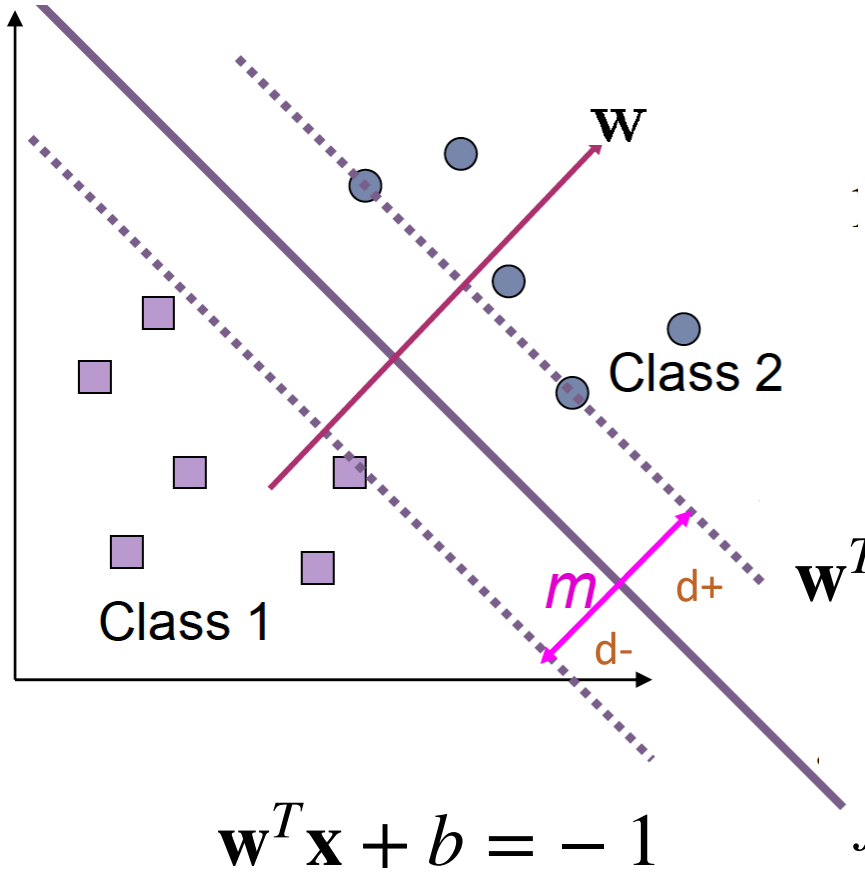
Η έννοια των διανυσμάτων στήριξης (support vectors)

—Τα διανύσματα στήριξης ορίζουν δύο επίπεδα παράλληλα στον υπερεπίπεδο του ταξινομητή f(x;w) = 0.



Η έννοια των διανυσμάτων στήριξης (support vectors)

—Τα διανύσματα στήριξης ορίζουν δύο επίπεδα παράλληλα στον υπερεπίπεδο του ταξινομητή f(x;w) = 0.



— Επιπλέον ισχύει:

$$m = \frac{2}{\|w\|_2}$$

$$\mathbf{w}^{\mathbf{T}}\mathbf{x} + b \geq 1 \quad \forall \mathbf{x} \in \omega_1$$

$$\mathbf{w}^{\mathbf{T}}\mathbf{x} + b \leq -1 \quad \forall \mathbf{x} \in \omega_2$$

$$\mathbf{w}^T\mathbf{x} + b = 1$$

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

$$f(x) = < \mathbf{w}, \mathbf{x} > + b = \mathbf{w}^T\mathbf{x} + b = 0$$

Μεγιστοποίηση περιθωρίου

— Επομένως ο στόχος μας είναι να μάθουμε τις w και b του γραμμικού ταξινομητή f(x) ο οποίος μεγιστοποιεί το περιθώριο m. Αυτός ο ταξινομητής ονομάζεται support vector machine.

— Οι άγνωστες παράμετροι προκύπτουν ως λύση του παρακάτω προβλήματος βελτιστοποίησης:

$$\max_{w,b} \frac{1}{\|w\|_2} \text{ s.t. } y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq 1, \forall\left\{x_i, y_i\right\} \in D$$

— Το παραπάνω πρόβλημα αποτελεί τετραγωνικό πρόβλημα βελτιστοποίησης (quadratic problem) για το οποίο υπάρχουν αρκετοί γνωστοί αλγόριθμοι.

Ισοδύναμο πρόβλημα πρόβλημα βελτιστοποίησης

— Η μεγιστοποίηση του περιθωρίου ισοδυναμεί με το πρόβλημα ελαχιστοποίησης

$$\min_{w,b} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i \left(\mathbf{x}_i^T \cdot w + b\right) + \sum_{i=1}^{N} \alpha_i$$

όπου α_i οι πολλαπλασιαστές Lagrnage για κάθε δεδομένο του συνόλου εκπάιδευσης.

— Το ενδιαφέρον είναι ότι οι άγνωστοι παράμετροι του SVM υπολογίζονται χρησιμοποιώντας μόνο τα S σε πλήθος support vectors!

$$\mathbf{w} = \sum_{j=1}^{S} \alpha_{t_j} y_{t_j} \mathbf{x}_j$$

Ταξινόμηση με SVM

— Όταν έρθει ένα νέο δείγμα **z** (εκτός του συνόλου εκπαίδευσης) αυτό ταξινομείται στην κατηγορία 1 εάν f(z)>0 και στην κατηγορία -1 εάν f(z)<0.

— Η συνάρτηση απόφασης είναι ένα linear discriminant

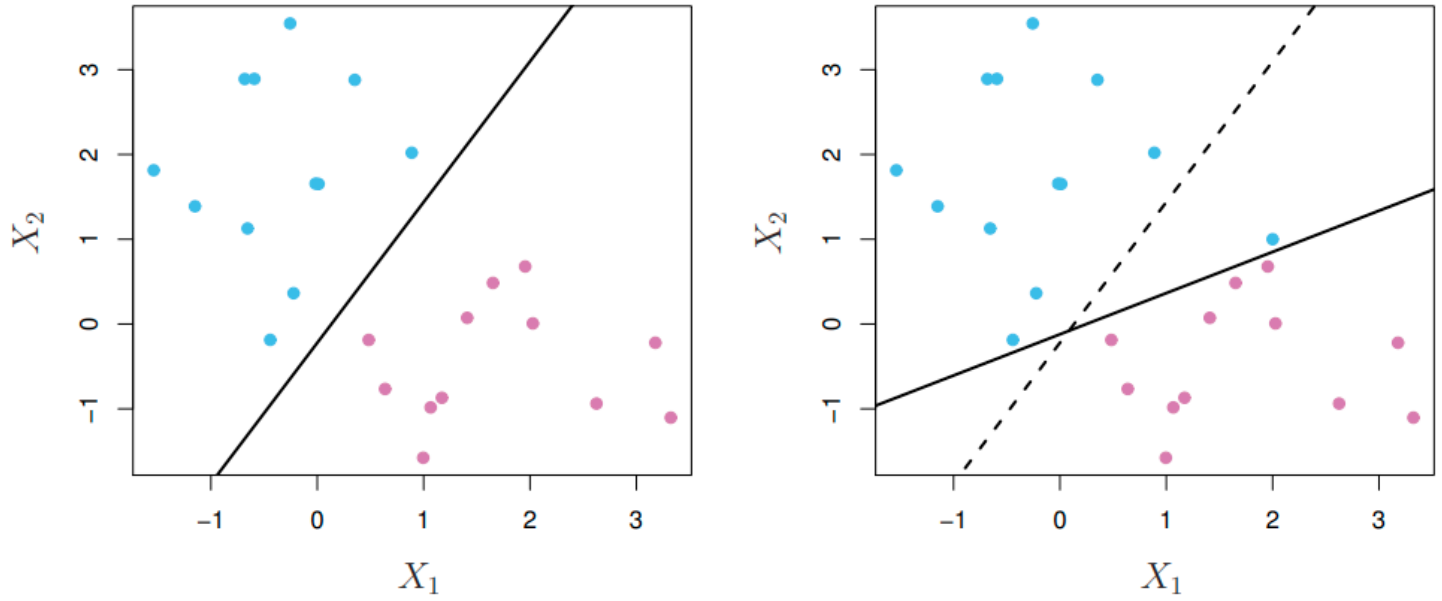
$$f = \mathbf{w}^T \mathbf{z} + b$$

— Το οποίο υπολογίζεται χρησιμοποιώντας μόνο τα support vectors

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

SVMs και δεδομένα με θόρυβο

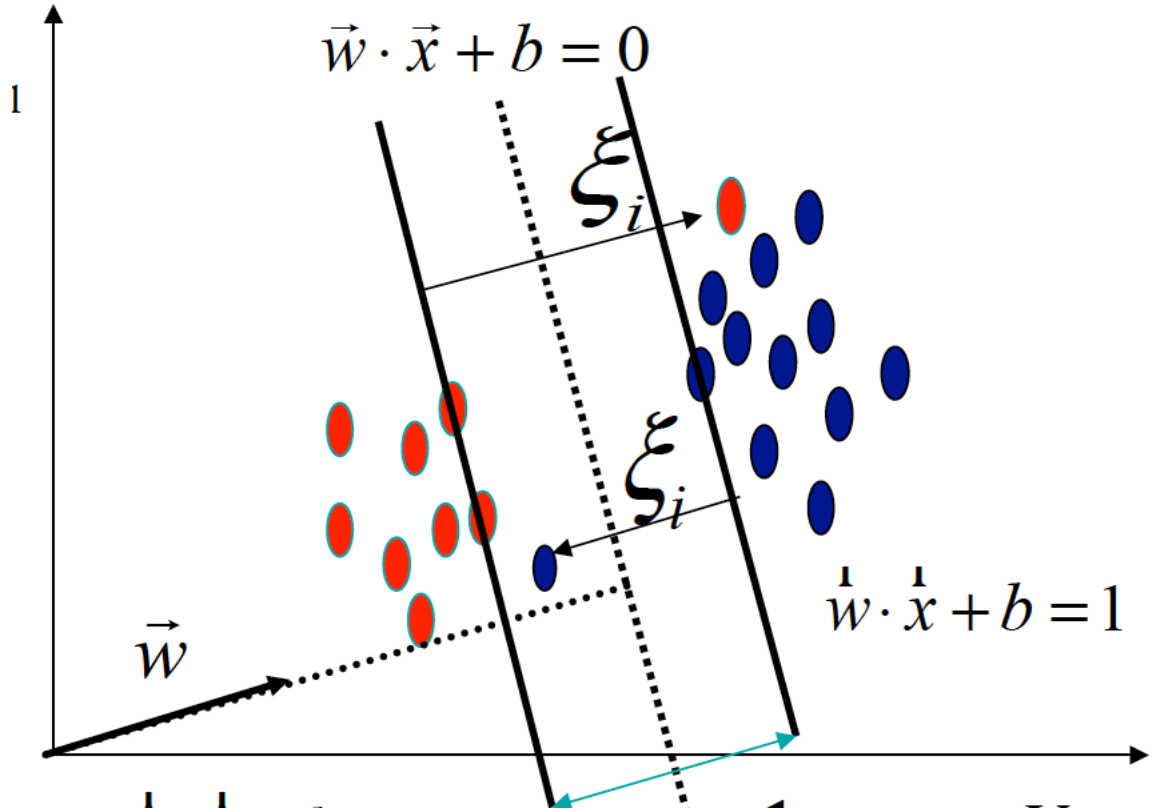
— Τα δεδομένα συχνά περιέχουν θόρυβο ή/και ακραίες τιμές (outliers), γεγονός που μπορεί να οδηγήσει σε μη-ακριβή εκτίμηση των παραμέτρων (βλ. δεξί διάγραμμα)



— Για να αντιμετωπιστεί αυτό το πρόβλημα μεγιστοποιούμε το λεγόμενο soft-margin.

SVMs και δεδομένα με θόρυβο

—Ιδέα: Επιτρέψτε μερικά δεδομένα να πέσουν μέσα στο περιθώριο, αλλά “τιμωρήσετε τα”. Αυτό επιτυγχάνεται με την εισαγωγή μεταβλητών χαλάρωσης (slack variables), μία για κάθε δεδομένο του συνόλου εκπαίδευσης.



— Οι slack variables (ξ) εκφράζουν την απόσταση των δεδομένων τα οποία δεν ταξινομούνται σωστά από το υπερεπίπεδο μέγιστου περιθωρίου, το οποίο είναι γνωστό ως hard margin.

Πρόβλημα βελτιστοποίησης

— Θελουμε να μεγιστοποιήσουμε το περιθώριο και ταυτόχρονα να ελαχιστοποιήσουμε το σφάλμα ταξινόμησης (δηλαδή όσο το δυνατόν λιγότερα δεδομένα εκπαίδευσης να ταξινομούνται εσφαλμένα)

— Με άλλα λόγια, θελουμε να μεγιστοποιήσουμε το περιθώριο και ταυτόχρονα να ελαχιστοποιήσουμε τις αποστάσεις ξ_i

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad s.t. \quad \begin{array}{l} y_i\left(w \cdot x_i + b\right) \geq 1 - \xi_i, \forall x_i \\ \xi_i \geq 0 \end{array}$$

— Το υπερεπίπεδο που προκύπτει ως λύση του παραπάνω προβλήματος βελτιστοποίσης ονομάζεται soft margin.

— Το C αποτελεί υπερπαράμετρο. Όταν το C τείνει στο άπειρο το soft margin τείνει στο hard margin.

Πρόβλημα βελτιστοποίησης

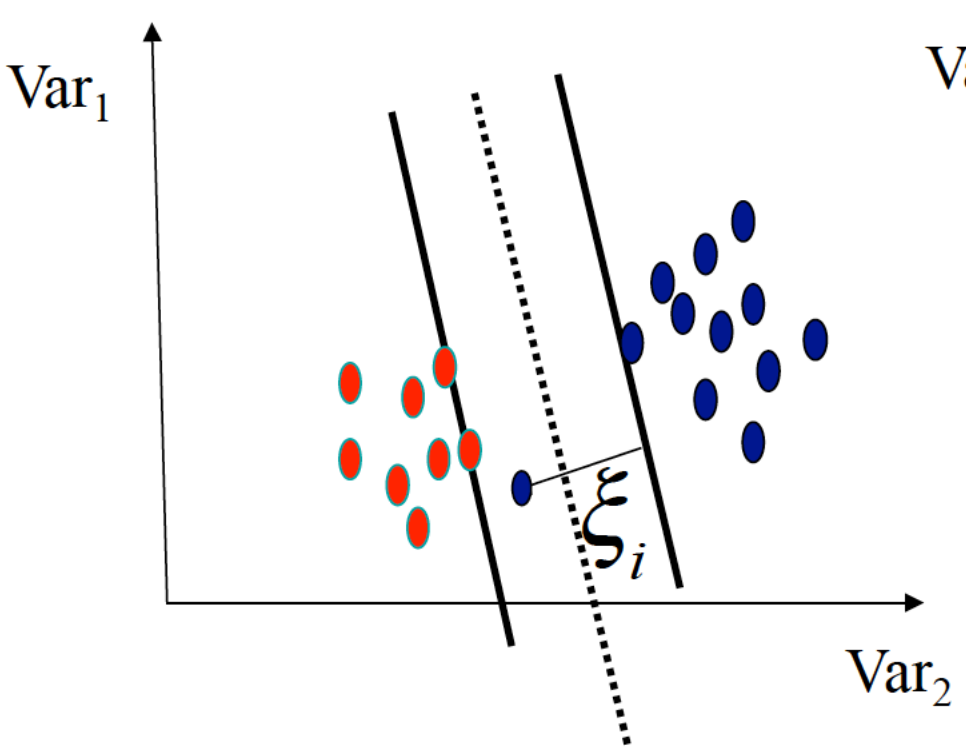
— Συχνά στη πράξη βελτιστοποιούμε το dual προβλημα

$$\max_{a} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_\lambda \mathbf{x}_i^T \mathbf{x}_j$$

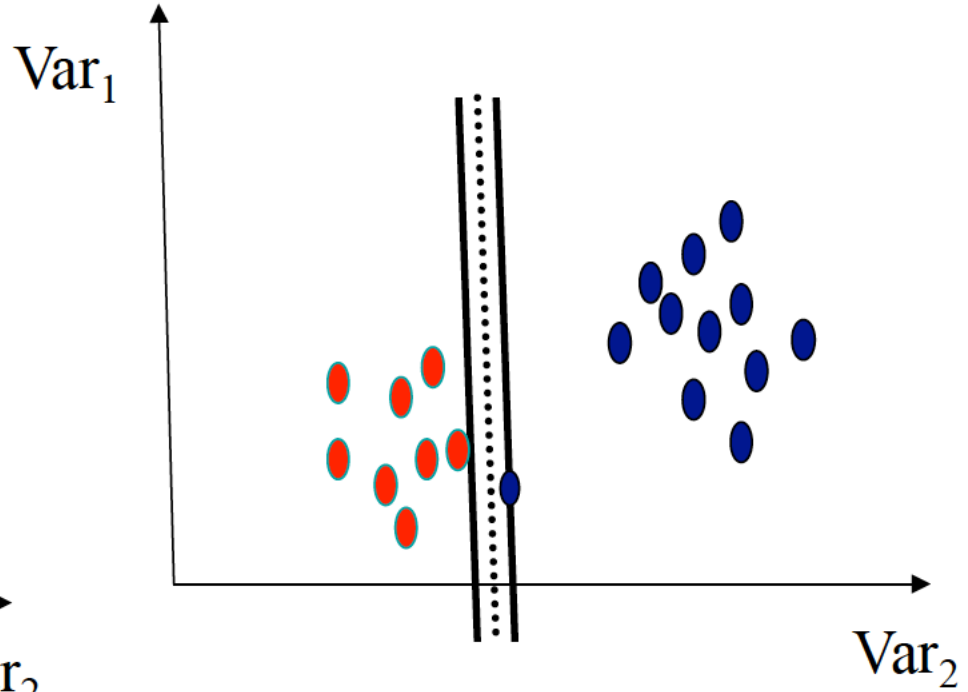
$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0$$

— Το ενδιαφέρον στη παραπάνω διατύπωση είναι ότι εμφανίζονται εσωτερικά γινόμενα των δεδομένων γεγονός χρήσιμο για την μη-γραμμική επέκταση των SVMs.

Soft vs Hard Margin



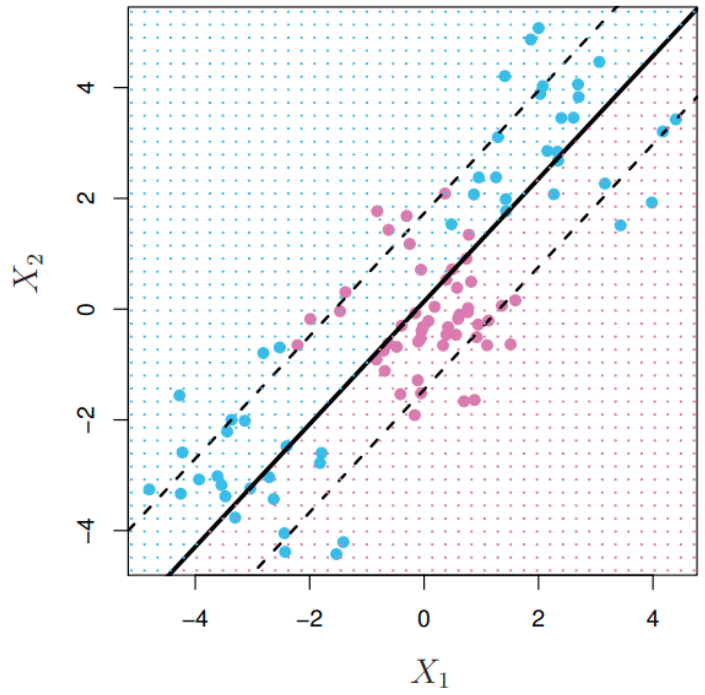
Soft Margin SVM



Hard Margin SVM

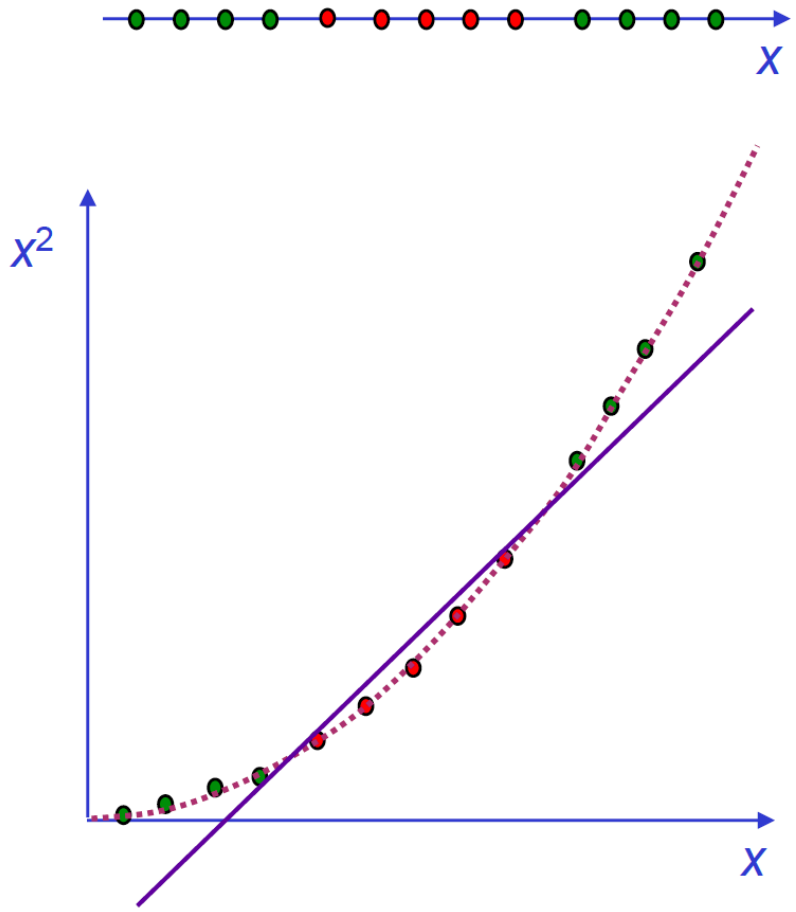
Μη-γραμμικά διαχωρίσιμες κλάσεις

— Συχνά στη πράξη τα δεδομένα δεν διαχωρίζονται από γραμμικά υπερεπίπεδα.



— Πώς θα αντιμετωπίσουμε αυτό το πρόβλημα στα SVMs;

Μη-γραμμικά SVMs



2 κατηγορίες, όπως φαίνεται στο σχήμα. Προφανώς απαιτείται μη γραμμικός διαχωρισμός. Το γνωστό μας γραμμικό SVM αδυνατεί να δώσει λύση.

Ιδέα: Ας μετασχηματίσουμε τα δεδομένα ώστε να κάνουμε το πρόβλημα γραμμικά διαχωρίσιμο. Επεκτείνουμε τα πρότυπα εισάγοντας ένα δεύτερο χαρακτηριστικό στα πρότυπα ίσο με το τετράγωνο του αρχικού χαρακτηριστικού και ταξινομούμε εύκολα με γραμμικό SVM στο επίπεδο (x,x^2).

Η διαχωριστική γραμμή είναι της μορφής

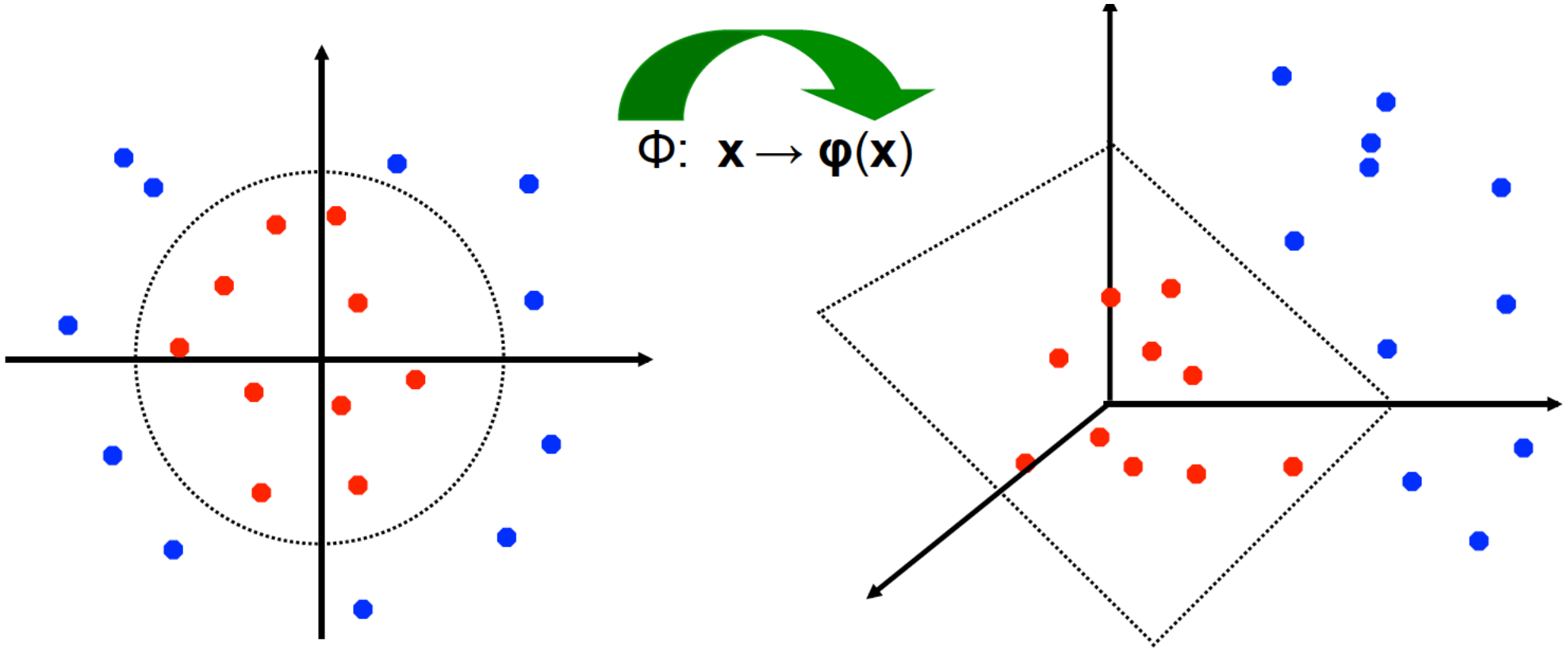
$$w_1 x + w_2 x^2 + w_0 = 0$$

χρειάστηκε δηλαδή να χρησιμοποιήσουμε ένα βάρος παραπάνω από το γραμμικό SVM.

Τα νέα πρότυπα «ζουν» μεν σ' ένα διδιάστατο χώρο, αλλά ανήκουν σε μια μονοδιάστατη καμπύλη του χώρου.

Μη-γραμμικά SVMs: χώροι χαρακτηριστικών

— Γενική ιδέα: Μπορούμε να απεικονίσουμε τα δεδομένα σε ένα χώρο χαρακτηριστικών (feature space) μεγαλύτερης διάστασης όπου τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα.



Πρόβλημα βελτιστοποίησης

— Το dual προβλημα για γραμμικά SVMs

$$\max_{a} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_\lambda \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0$$

— Εάν μετασχηματίσουμε μη-γραμμικά τα δεδομένα σε νέο χώρο χαρακτηριστικών το παραπάνω πρόβλημα τροποποιείται ελάχιστα ως

$$\max_{a} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_\lambda \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Kernel functions

— Μια συνάρτηση πυρήνα (kernel function) ορίζεται ως

$$K\left(x_1, x_2\right) = \left\langle \Phi\left(x_1\right), \Phi\left(x_2\right) \right\rangle = \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)$$

— Περιγράφουν την ομοιότητα των δεδομένων

— Οι πυρήνες είναι εξαιρετικά χρήσιμοι στη μηχανική μάθηση

This original formula...

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

...becomes this, when we apply a data transformation...

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

...and if we can define a kernel...

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

...it becomes this...

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

- Suppose ϕ is given as follows

$$\phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

- An inner product in the feature space is

$$\langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle = (1 + x_1y_1 + x_2y_2)^2$$

** NOTE: On this slide *y* is not the label, it is a feature vector, just like *x* **

- So, if we define the kernel function as follows, there is no need to carry out f(.) explicitly

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- This use of kernel function to avoid calculating ϕ explicitly is known as the **kernel trick**

Examples of Kernel Functions

** NOTE: On this slide *y* is not the label, it is a feature vector, just like *x* **

- Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel with width s

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

  - Closely related to radial basis function neural networks
  - The feature space is infinite-dimensional

- Sigmoid with parameter k and q

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

  - It does not satisfy the Mercer condition on all k and q

Classifying with a Kernel

- For testing, the new data **z** is classified as class 1 if $f \geq 0$ and as class -1 if $f < 0$

Original

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

With kernel function

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$$

$$f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

NOTE: ***y*** is once again a label from the set {-1, 1} **

You as the SVM user

- You have two main choices to make:
  1) What kernel will you use?

     Polynomial?

     Radial Basis Function?

     Something else?

  2) How much "slack" will you allow?

     Depends on how much you trust data collection and labeling.

Σύνοψη

— Τα SVMs έχουν ως στόχο την εύρεση ενός υπερεπιπέδου (hyperplane) σε έναν χώρο υψηλών διαστάσεων που διαχωρίζει βέλτιστα τα δεδομένα εκπαίδευσης σε διαφορετικές κλάσεις. Το υπερεπίπεδο επιλέγεται με τέτοιο τρόπο ώστε να μεγιστοποιεί το περιθώριο (margin) μεταξύ των κλάσεων, δηλαδή την απόσταση μεταξύ του υπερεπιπέδου και των πλησιέστερων δεδομένων εκπαίδευσης από κάθε κλάση. Αυτό το μέγιστο περιθώριο παρέχει καλύτερη γενίκευση και ανθεκτικότητα στον θόρυβο των δεδομένων.

— Επιπλέον, οι SVMs χρησιμοποιούν πυρήνες (kernels) για τη μετατροπή των δεδομένων σε έναν χώρο υψηλότερων διαστάσεων, όπου τα δεδομένα μπορούν να διαχωριστούν γραμμικά. Αυτό επιτρέπει στις SVMs να χειρίζονται μη γραμμικά διαχωρίσιμα δεδομένα αποτελεσματικά.

SVMs Vs CNNs

— Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVMs) και τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) είναι δύο διαφορετικές προσεγγίσεις στη μηχανική μάθηση, καθεμία με τα δικά της πλεονεκτήματα και μειονεκτήματα.

— **Αρχιτεκτονική:**

- SVMs: Οι SVMs είναι γραμμικοί ταξινομητές που προσπαθούν να βρουν ένα βέλτιστο υπερεπίπεδο για τον διαχωρισμό των κλάσεων στο χώρο χαρακτηριστικών.
- CNNs: Τα CNNs είναι νευρωνικά δίκτυα που αποτελούνται από πολλαπλά επίπεδα, συμπεριλαμβανομένων συνελικτικών επιπέδων, επιπέδων συγκέντρωσης (pooling) και πλήρως συνδεδεμένων επιπέδων.

SVMs Vs CNNs

— **Χειρισμός δεδομένων:**

- SVMs: Οι SVMs λειτουργούν καλά με δεδομένα υψηλών διαστάσεων και μπορούν να χειριστούν μη γραμμικά διαχωρίσιμα δεδομένα χρησιμοποιώντας πυρήνες (kernels).
- CNNs: Τα CNNs είναι ιδανικά για δεδομένα με χωρική δομή, όπως εικόνες, καθώς μπορούν να μάθουν ιεραρχικά χαρακτηριστικά από τα δεδομένα.

SVMs Vs CNNs

— **Απαιτήσεις σε δεδομένα εκπαίδευσης**:

- SVMs: Οι SVMs μπορούν να λειτουργήσουν καλά με μικρότερα σύνολα δεδομένων εκπαίδευσης.
- CNNs: Τα CNNs συνήθως απαιτούν μεγάλες ποσότητες δεδομένων εκπαίδευσης για να επιτύχουν καλές επιδόσεις.

SVMs Vs CNNs

— **Ερμηνεία των αποτελεσμάτων:**

- SVMs: Τα αποτελέσματα των SVMs είναι πιο ερμηνεύσιμα, καθώς βασίζονται σε ένα μοναδικό βέλτιστο υπερεπίπεδο.
- CNNs: Τα CNNs είναι συχνά θεωρούνται ως "μαύρα κουτιά" λόγω της πολύπλοκης αρχιτεκτονικής τους, καθιστώντας πιο δύσκολη την ερμηνεία των αποτελεσμάτων.

SVMs Vs CNNs

— **Υπολογιστικό κόστος:**

- SVMs: Οι SVMs έχουν γενικά χαμηλότερες υπολογιστικές απαιτήσεις σε σύγκριση με τα CNNs.
- CNNs: Τα CNNs μπορεί να είναι υπολογιστικά ακριβά λόγω της πολύπλοκης αρχιτεκτονικής τους και των μεγάλων συνόλων δεδομένων εκπαίδευσης.