Support Vector Machine

Instructor: Junghye Lee

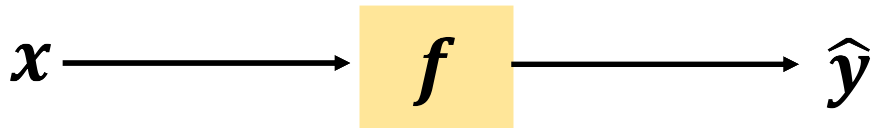
Department of Industrial Engineering
junghyelee@unist.ac.kr

Contents

1

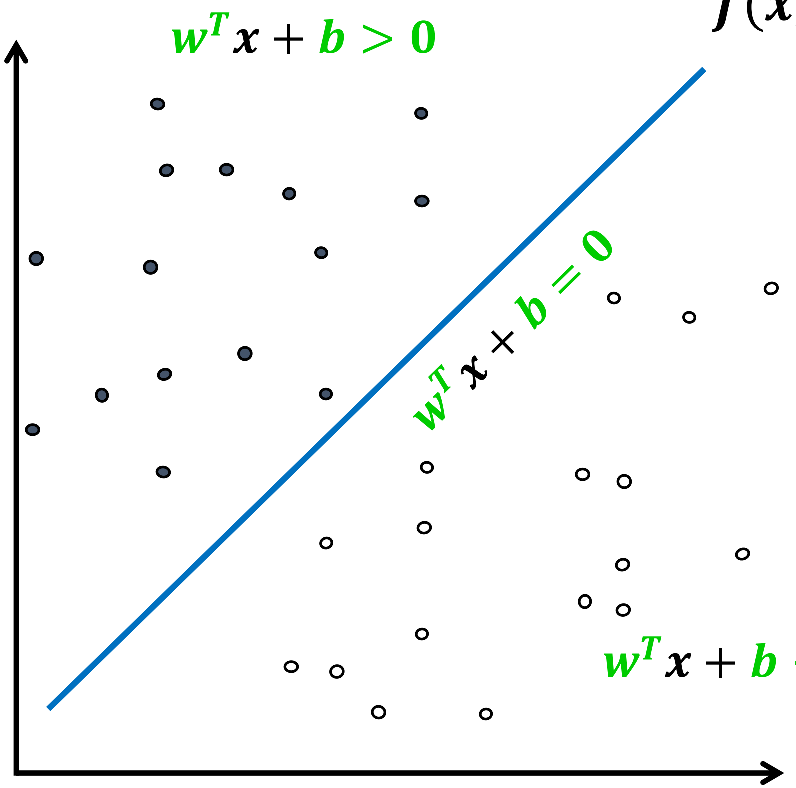
Linear Support Vector Machine
(Separable case)

Linear Classifier



- denotes +1 (class 1)
- denotes -1 (class 2)

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



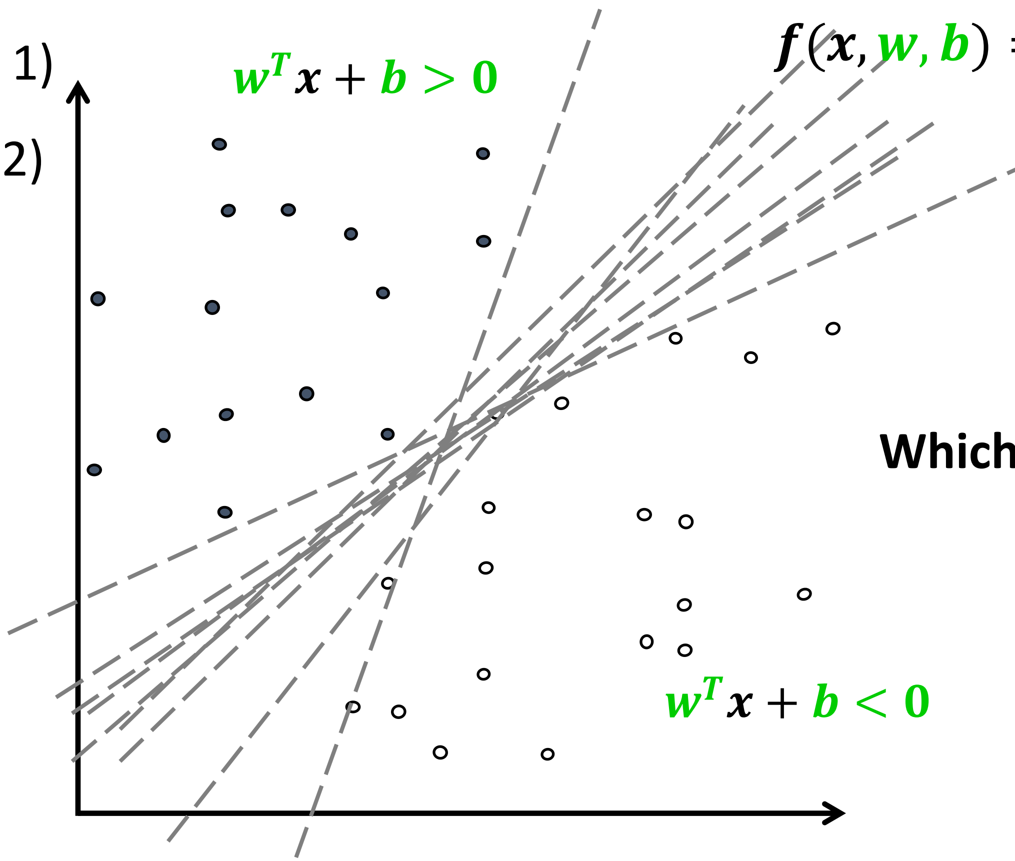
How would you classify this data?

Linear Classifier

$$\boldsymbol{x} \longrightarrow \boxed{\boldsymbol{f}} \longrightarrow \hat{\boldsymbol{y}}$$

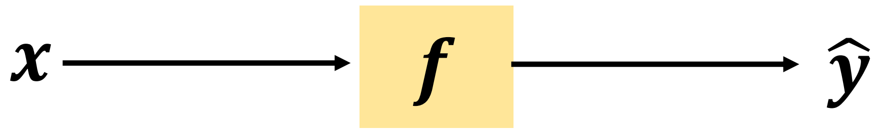
- denotes +1 (class 1)
- denotes -1 (class 2)

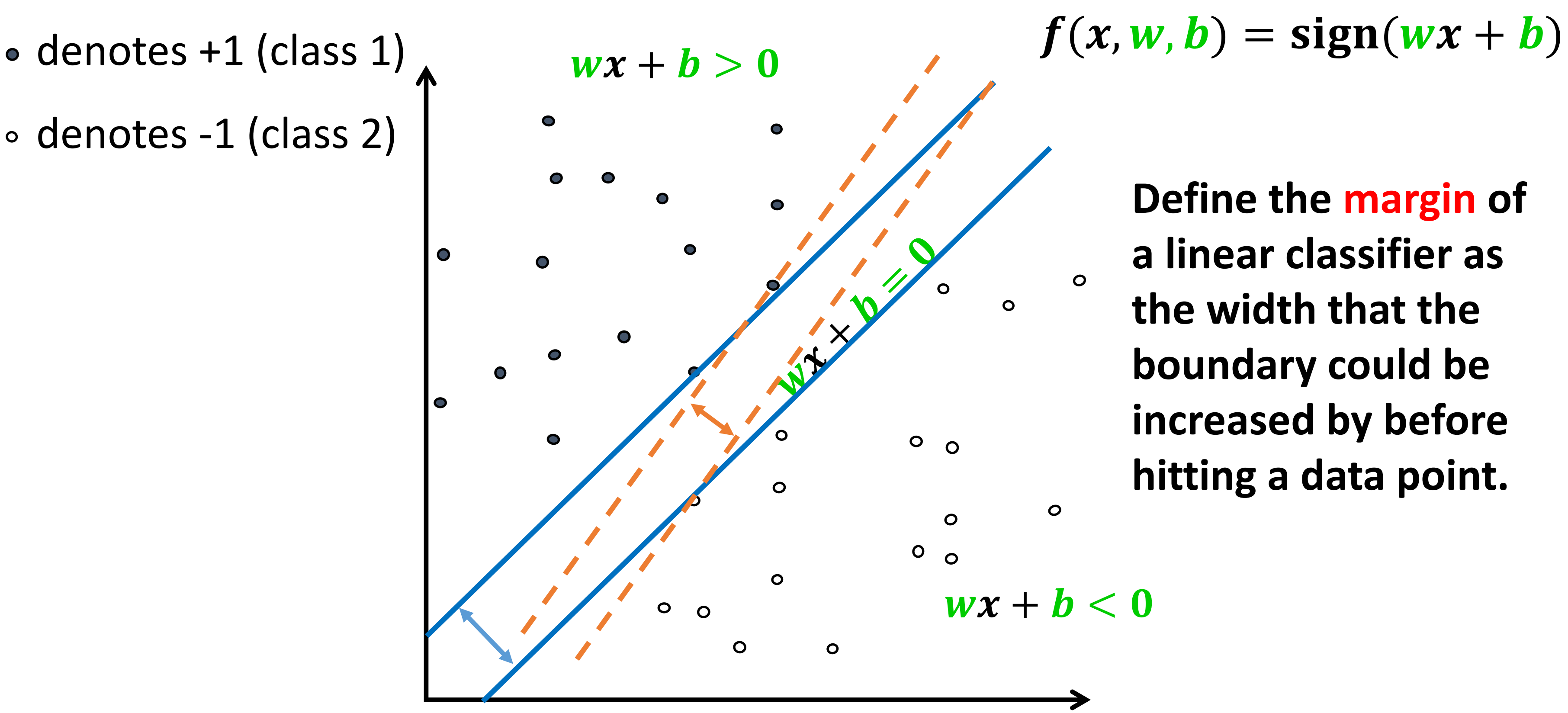
$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{b}) = \mathbf{sign}(\boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{b})$$



Which one is the best?

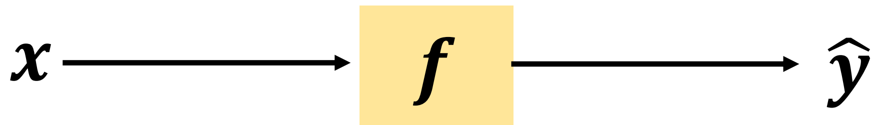
Linear Classifier

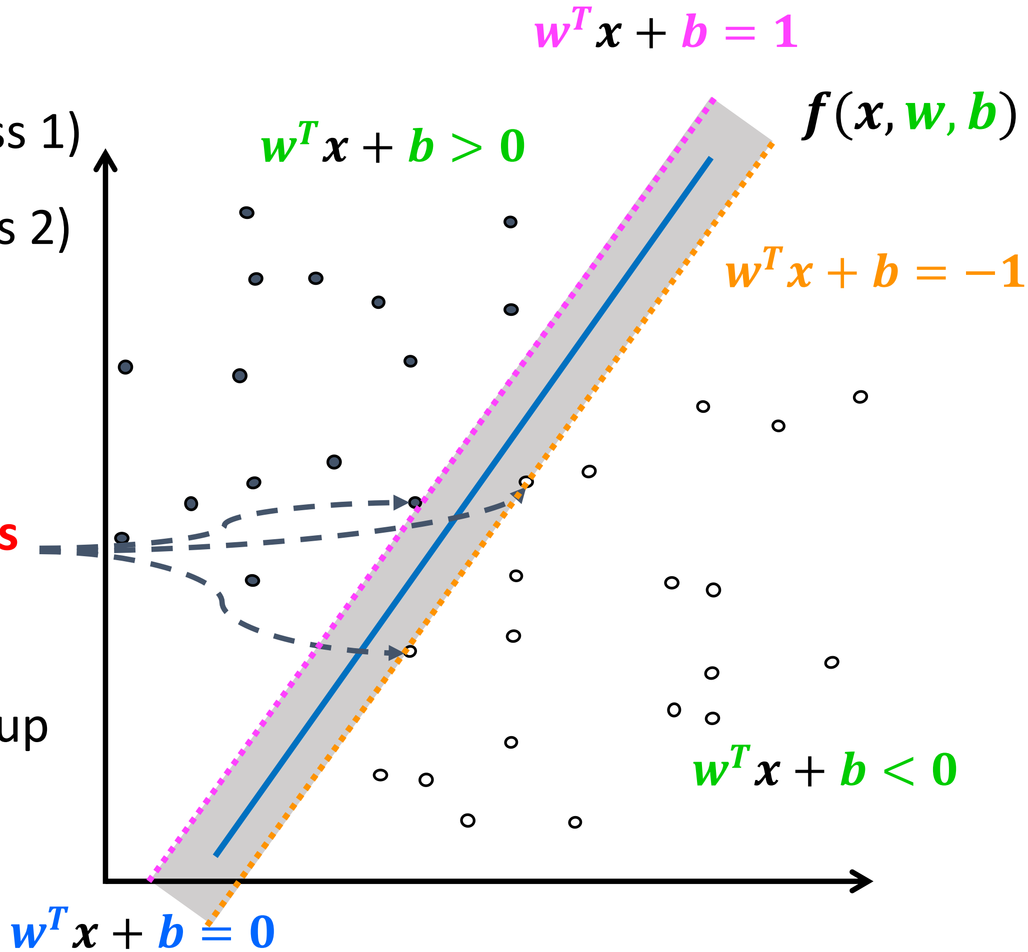




Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a data point.

Linear Classifier





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

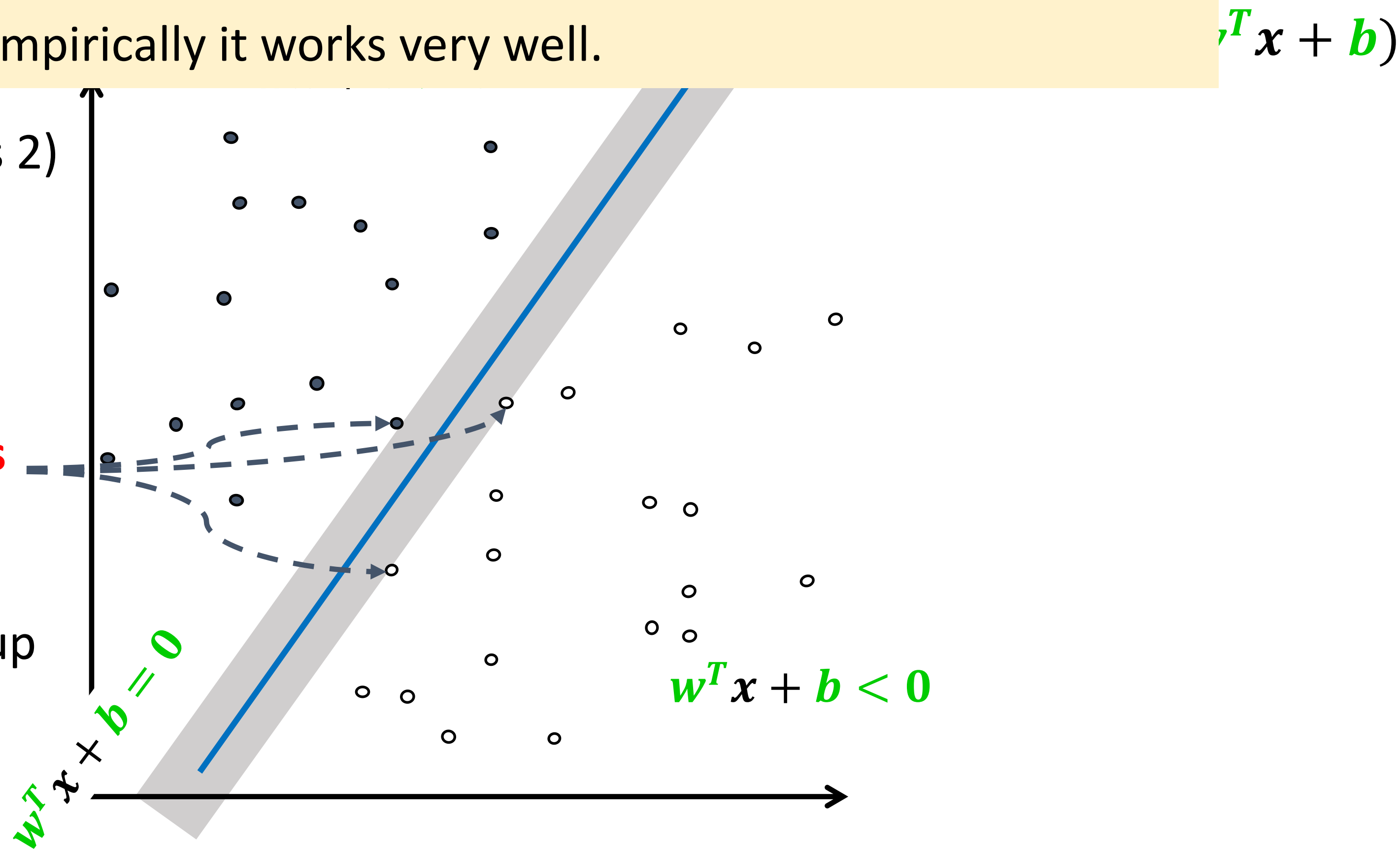
- denotes +1 (class 1)
- denotes -1 (class 2)

Support Vectors are those data points that the margin pushes up against.

Linear Classifier

1. Maximizing the margin is good.
2. Implies that only support vectors are important;
   other training examples are ignorable.
3. Empirically it works very well.

- denotes +
- denotes -1 (class 2)

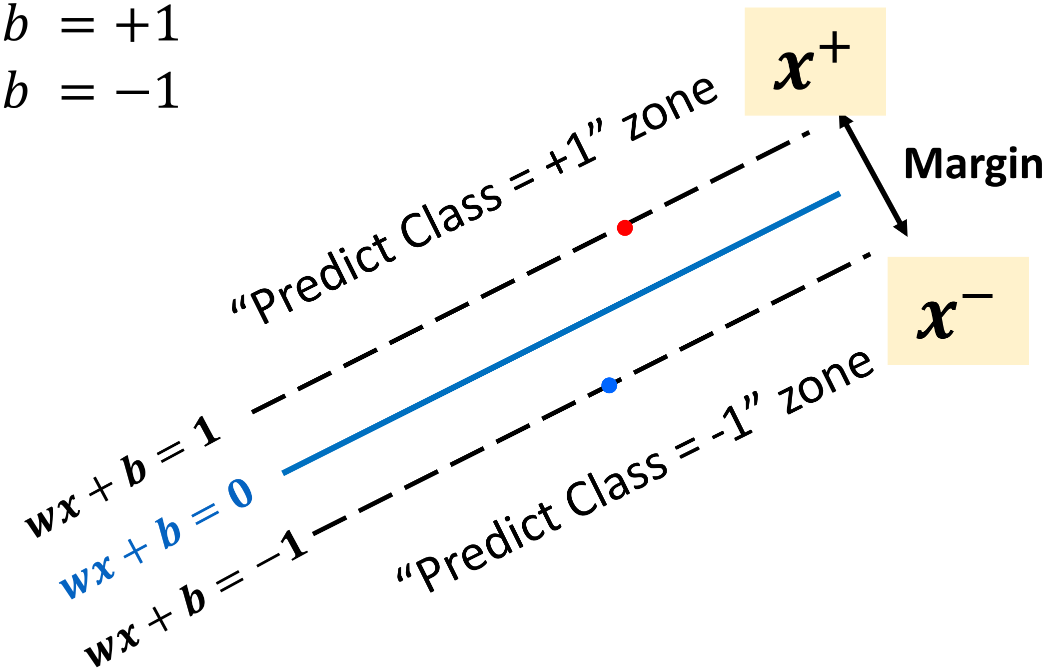


Support Vectors are those data points that the margin pushes up against.

Linear SVM Mathematically

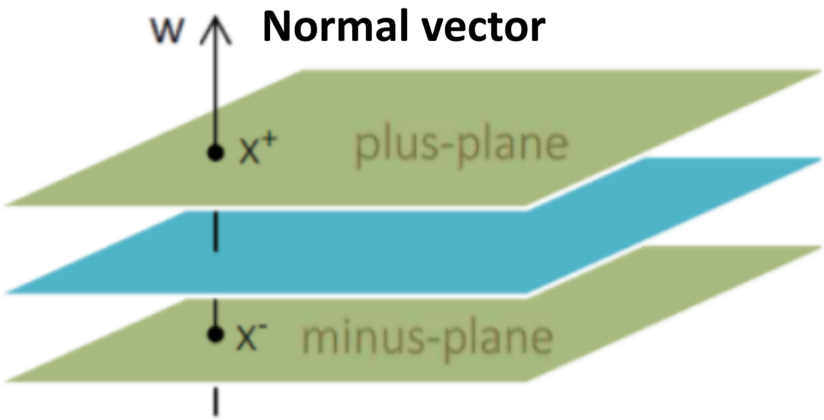
What we know:

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



$$\boldsymbol{M}(\textbf{margin}) = \frac{\mathbf{2}}{\|\boldsymbol{w}\|_2} = \frac{\mathbf{2}}{\sqrt{\boldsymbol{w}^T\boldsymbol{w}}}$$

Linear SVM Mathematically



$$w^T x + b = 1$$

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0$$

$$w^T x + b = -1$$

What we know: $\boldsymbol{x}^+ = \boldsymbol{x}^- + \lambda \boldsymbol{w}$

1st step

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

$$\boldsymbol{w}^T (\boldsymbol{x}^- + \lambda \boldsymbol{w}) + b = 1$$

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

$$-1 + \lambda \boldsymbol{w}^T \boldsymbol{w} = 1$$

$$\therefore \lambda = \frac{2}{\boldsymbol{w}^T \boldsymbol{w}}$$

2nd step

$$\boldsymbol{Margin} = \boldsymbol{distance}(\boldsymbol{x}^+, \boldsymbol{x}^-)$$

$$= \|\boldsymbol{x}^+ - \boldsymbol{x}^-\|_2$$

$$= \|\boldsymbol{x}^- + \lambda \boldsymbol{w} - \boldsymbol{x}^-\|_2$$

$$= \|\lambda \boldsymbol{w}\|_2$$

$$= \lambda \sqrt{\boldsymbol{w}^T \boldsymbol{w}}$$

$$= \frac{2}{\boldsymbol{w}^T \boldsymbol{w}} \sqrt{\boldsymbol{w}^T \boldsymbol{w}}$$

$$\therefore \frac{2}{\sqrt{\boldsymbol{w}^T \boldsymbol{w}}} = \frac{2}{\|\boldsymbol{w}\|_2}$$

https://www.youtube.com/watch?v=GBhH1UNr1VI

Linear SVM Mathematically

1) Correctly classify all training data

$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_i^{+} + b \geq +1 \qquad \text{if } y_i = +1$$

$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_i^{-} + b \leq -1 \qquad \text{if } y_i = -1$$

$$y_i\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) \geq +1 \qquad \text{for all } i$$

2) Maximize the Margin $\dfrac{2}{\|\boldsymbol{w}\|_2}$ same as minimize $\dfrac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w}$

We can formulate a Quadratic Optimization Problem and solve for $\boldsymbol{w}$ and $\boldsymbol{b}$

Minimize $$\Phi(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{w}$$

subject to $$y_i\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}} + b\right) \geq +1$$

Solving the Optimization Problem

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ is minimized;
and for all $\{(\boldsymbol{x_i}, y_i)\}$: $y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1$

- Need to optimize a *quadratic* function subject to ***linear*** constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a ***dual problem*** where a ***Lagrange multiplier*** $\boldsymbol{\alpha_i}$ is associated with every constraint in the primary problem:

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j}$ **is maximized and**
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

  $\boldsymbol{w} = \Sigma_{\boldsymbol{i}} \alpha_i y_i \boldsymbol{x_i}$ $\quad b = y_i - \boldsymbol{w^T x_i}$ for any $\boldsymbol{x_i}$ such that $\alpha_i \neq 0$

- Each non-zero α_i indicates that corresponding $\boldsymbol{x_i}$ is a support vector.
- Then the classifying function will have the form:

$$f(\boldsymbol{x}) = \Sigma \alpha_i y_i \boldsymbol{x_i}^{\boldsymbol{T}} \boldsymbol{x} + b$$

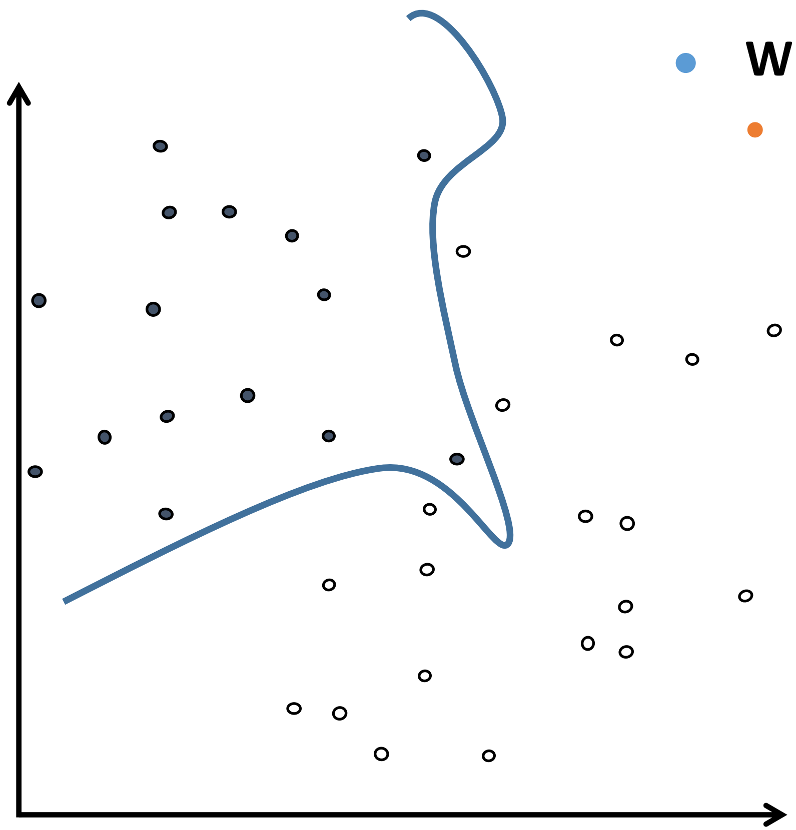
- Notice that it relies on an *inner product* between the test point **x** and the support vectors $\boldsymbol{x_i}$ we will return to this later.
- Also keep in mind that solving the optimization problem involved computing the inner products $\boldsymbol{x_i}^{\boldsymbol{T}} \boldsymbol{x_j}$ between all pairs of training points.

2

Linear Support Vector Machine (Non-separable case)

Dataset with noise

- denotes +1 (class 1)
- denotes -1 (class 2)

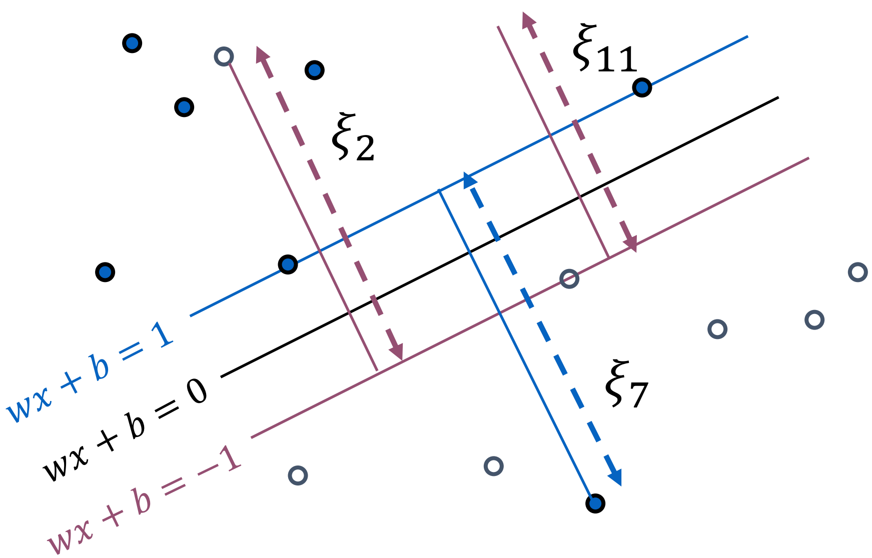


- **Hard Margin: So far we require all data points to be classified correctly**
  - **No training error**
- **What if the training set is noisy?**
  - **Solution 1: use very powerful kernels**

Overfitting!

Soft Margin Classification

Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples.



What should our quadratic optimization criterion be?

Minimize $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \boldsymbol{C}\boldsymbol{\Sigma}_{\boldsymbol{k}=1}^{\boldsymbol{R}}\boldsymbol{\xi}_{\boldsymbol{k}}$

Hard Margin v.s. Soft Margin

- The old formulation:

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ is minimized and for all $\{(\boldsymbol{x_i}, y_i)\}$
$y_i\,(\boldsymbol{w^T x_i} + b) \geq 1$

- The new formulation incorporating slack variables:

Find $\boldsymbol{w}$ and b such that
$\boldsymbol{\Phi}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\Sigma\xi_i$ is minimized and for all $\{(\boldsymbol{x_i}, y_i)\}$
$y_i\,(\boldsymbol{w^T x_i} + b) \geq 1 - \xi_i$ **and** $\xi_i \geq 0$ for all i

- Parameter C can be viewed as a way to control overfitting.

Linear SVM: Overview

- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyp erplane.
- Quadratic optimization algorithms can identify which training points $\boldsymbol{x_i}$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation of the problem and in the solution training po ints appear only inside dot products:

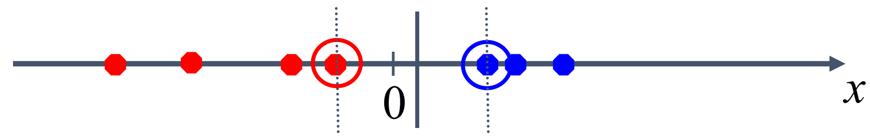
Find $\alpha_1 \dots \alpha_N$ **such that**
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i{}^T x_j$ **is maximized and**
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

$$f(x) = \Sigma\alpha_i y_i x_i{}^T x + b$$

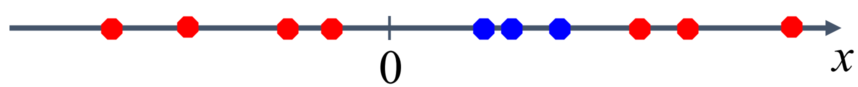
3 Nonlinear Support Vector Machine

Non-linear SVM

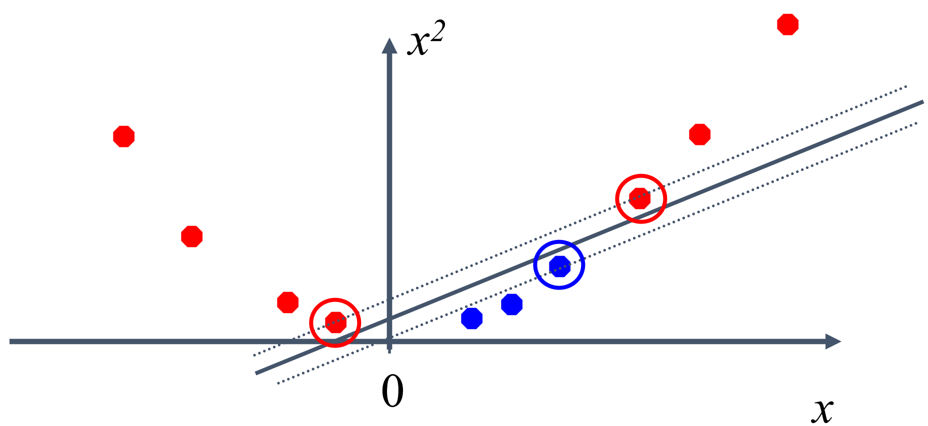
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

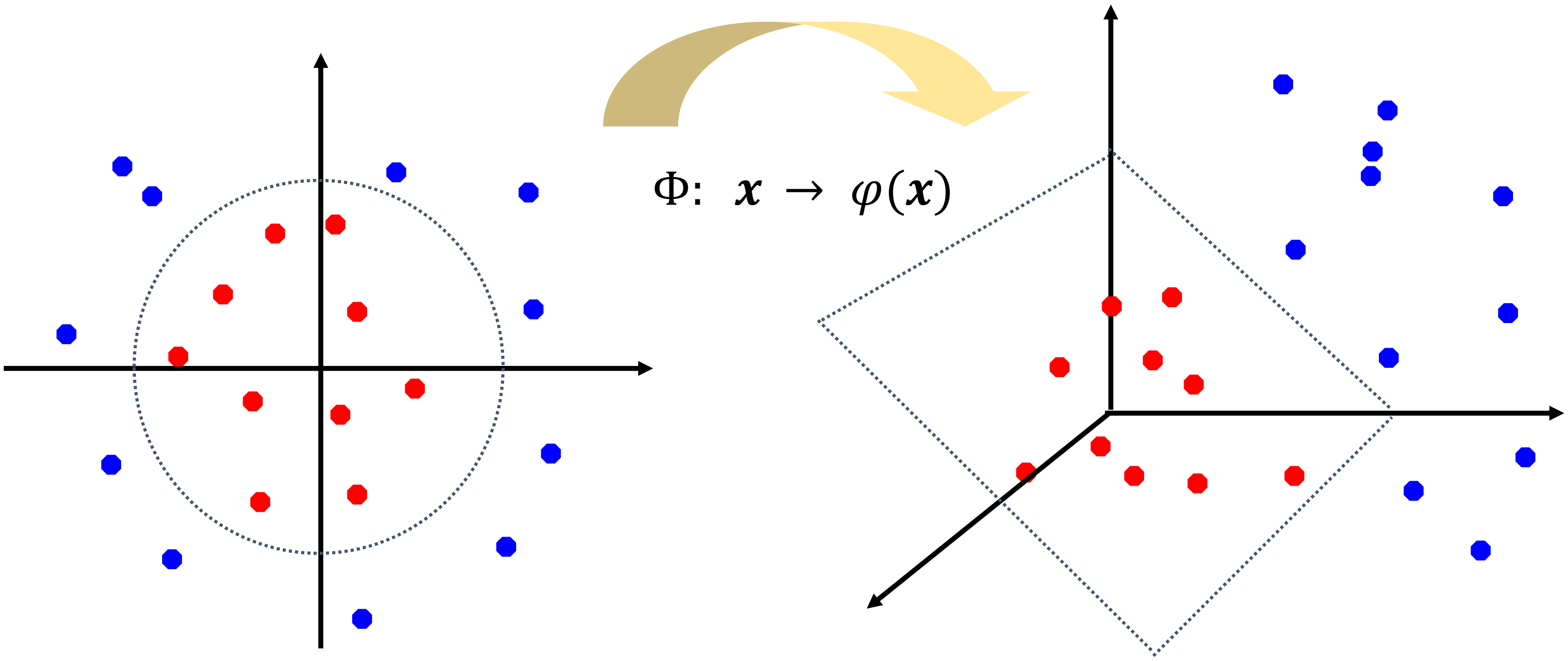


- How about mapping data to a higher-dimensional space:



Non-linear SVM: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Kernel Trick

- The linear classifier relies on dot product between vectors $K(x_i, x_j) = x_i{}^T x_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \varphi(x)$, the dot product becomes:
$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$
- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example: 2-dimensional vectors $x = [x_1\ x_2]$; let $K(x_i, x_j) = \left(1 + x_i^T x_j\right)^2$,

  Need to show that $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$:

  $K\left(x_i, x_j\right)$

  $= \left(1 + x_i^T x_j\right)^2$

  $= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$

  $= \left[1 x_{i1}^2 \sqrt{2} x_{i1} x_{i2} x_{i2}^2 \sqrt{2} x_{i1} \sqrt{2} x_{i2}\right]^T \left[1 x_{j1}^2 \sqrt{2} x_{j1} x_{j2} x_{j2}^2 \sqrt{2} x_{j1} \sqrt{2} x_{j2}\right]$

  $= \varphi(x_i)^T \varphi(x_j)$, where $\varphi(x) = \left[1 x_1^2 \sqrt{2} x_1 x_2 x_2^2 \sqrt{2} x_1 \sqrt{2} x_2\right]$

What Functions are Kernels?

- For some functions $K(x_i, x_j)$ checking that $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ can be cumbersome.
- Mercer's theorem: *Every semi-positive definite symmetric fun ction is a kernel*
- Semi-positive definite symmetric functions correspond to a se mi-positive definite symmetric Gram matrix:

$K =$

$K(x_1, x_1)$	$K(x_1, x_2)$	$K(x_1, x_3)$	…	$K(x_1, x_N)$
$K(x_2, x_1)$	$K(x_2, x_2)$	$K(x_2, x_3)$		$K(x_2, x_N)$
…	…	…	…	…
$K(x_N, x_1)$	$K(x_N, x_2)$	$K(x_N, x_3)$	…	$K(x_N, x_N)$

Semi-positive Definiteness of a Matrix

- In linear algebra, a **symmetric** $n \times n$ real matrix M is said to be **positive definite** if the scalar $\boldsymbol{z}^{\boldsymbol{T}}\boldsymbol{M}\boldsymbol{z}$ **is strictly positive** for every non-zero column vector $\boldsymbol{z}$ of n real numbers.

  $\boldsymbol{z}^{\boldsymbol{T}}\boldsymbol{M}\mathbf{z} > 0$ for all $\boldsymbol{z} \in \mathbb{R}^n \setminus \mathbf{0}$

- Positive semi-definite matrices are defined similarly, except that the above scalars $\boldsymbol{z}^{\boldsymbol{T}}\boldsymbol{M}\boldsymbol{z}$ must be positive or zero (i.e. non-negative).

  $\boldsymbol{z}^{\boldsymbol{T}}\boldsymbol{M}\mathbf{z} \geq 0$ for all $\boldsymbol{z} \in \mathbb{R}^n$

Examples of Kernel Functions

- Linear: $K(x_i, x_j) = x_i^T x_j$

- Polynomial of power p: $K(x_i, x_j) = (1 + x_i^T x_j)^p$

- Gaussian (radial-basis function): $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

- Sigmoid: $K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1)$

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \ldots \alpha_N$ **such that**
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j)$ **is maximized and**
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$f(x) = \Sigma\alpha_i y_i K(x_i, x_j) + b$$

- Optimization techniques for finding α_i's remain the same!

Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly, simply by defining a kernel function
- The kernel function plays the role of the dot product in the feature space.