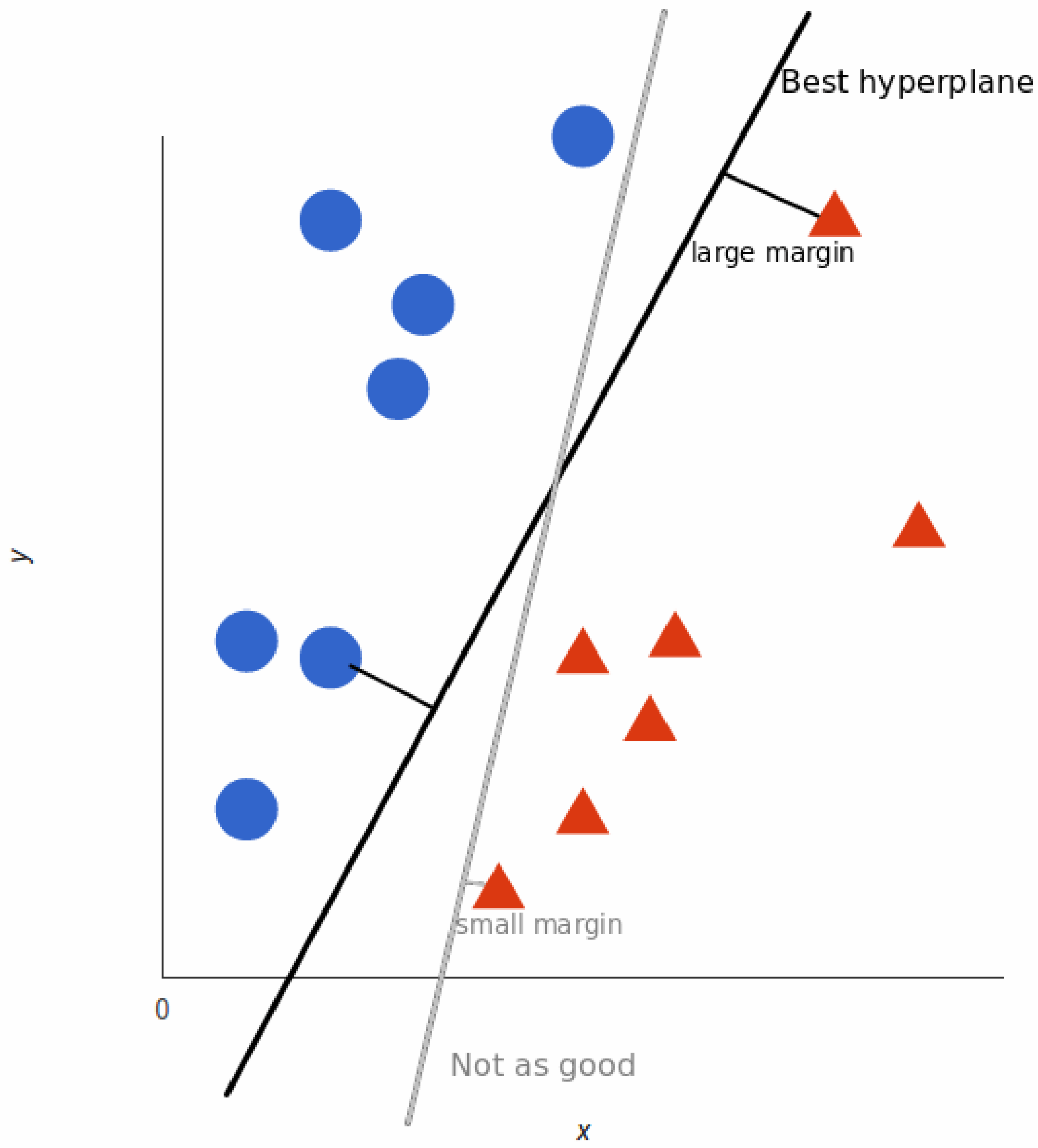
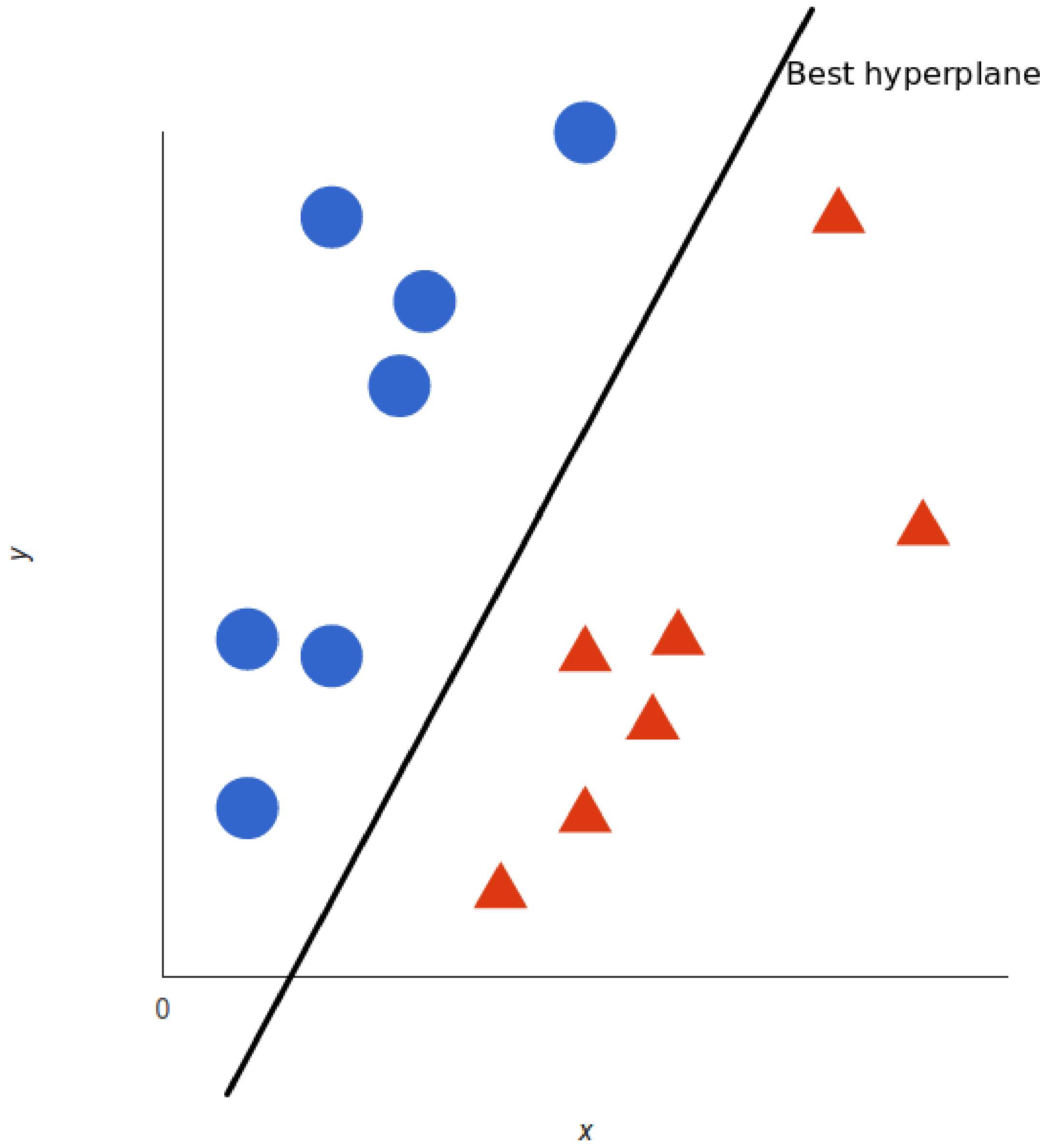
Support Vector Machine

Support Vector Machine (SVM)

- SVM is one of the most popular Supervised Learning algorithms, which is used for Classification
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes
- When a new data point is placed it is correctly categorized in the future.
- The best decision boundary is called a hyperplane for dimensions higher than 2



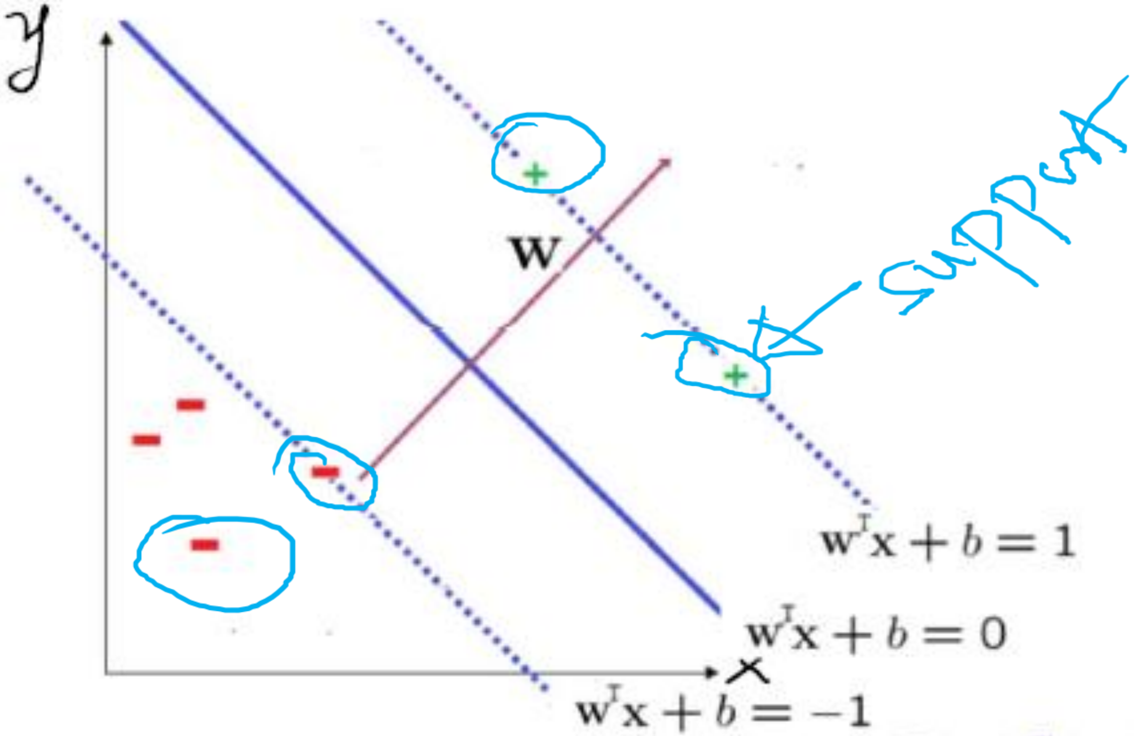
Best hyperplane
large margin
y
small margin
0
Not as good
x



Best hyperplane
y
0
x

Optimal Hyper-plane for linearly separable patterns

- Consider the following space consists of positive and negative examples.
- We separate positive examples from negative examples by drawing straight lines (or decision boundaries).
- There are many ways we can draw straight lines to separate these data.
- In support vector machines we discuss about how to draw this lines such a way it has the widest width that separate the positive samples from the negative samples.



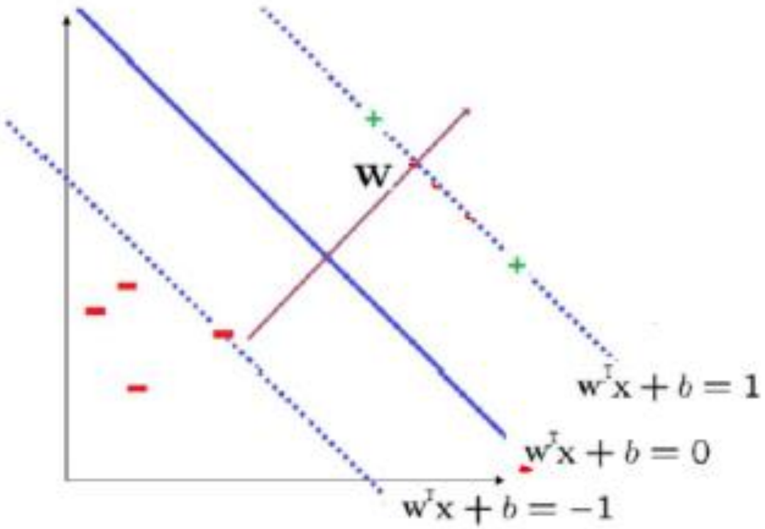
Optimal Hyper-plane for linearly separable patterns

- How to make a decision rule when you have the decision boundary? Let X is an unclassified example then we can say that,
  $W.X + b \geq 0$ then X is a positive example
  where W is the weight vector, X is a unclassified example, and b is the bias term.
- How to determine the W and b so that it gives the widest width to separate sample data?

For positive examples, what we can write is:
$\boxed{W.X_+ + b \geq +1}$ if 1 is the desired output for the positive examples.

For negative examples, what we can write is:
$\boxed{W.X_- + b \leq -1}$ if -1 is the desired output for negative examples.

We can introduce a variable y such that $y = +1$ for positive examples and $y = -1$ for negative examples.



$\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

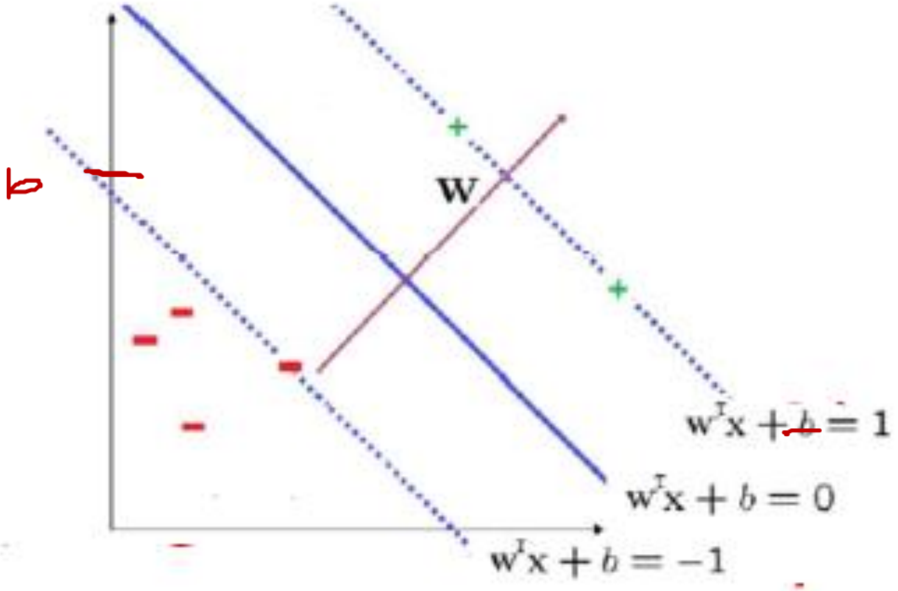
Optimal Hyper-plane for linearly separable patterns

- The particular data point that satisfy the $W.X_+ + b \geq +1$ and $W.X_- + b \leq -1$ with the equality sign are called support vectors.
- All the remaining examples in the training sample are completely irrelevant.
- In conceptual terms, support vectors are those that lie closest to the optimal boundary or hyper-plane and are therefore the most difficult to classify.

Now we can rewrite the both equations as follows:
multiply by y_i s of positive examples,
$y_i(W.X_i + b) \geq +1$

similarly multiply by by y_i s of negative examples such that $y_i(W.X_i + b) \geq +1$.

Since both statements are the same, we can write $\boxed{y_i(W.X_i + b) - 1 \geq 0}$.



Optimal Hyper-plane for linearly separable patterns

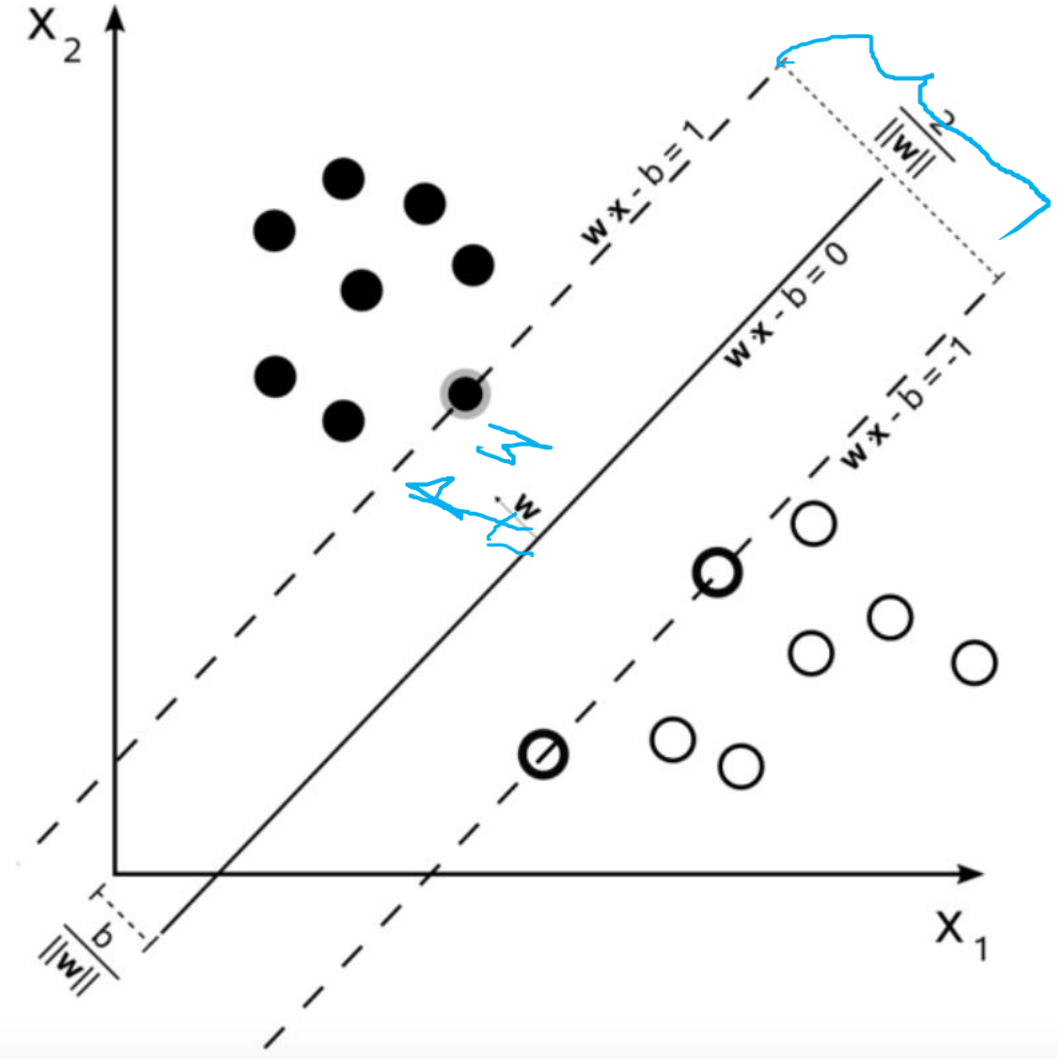
- For those examples that are on the lines we can say that $y_i(W.X_i + b) - 1 = 0$.

Using some algebraic measures we can show that width is $= \frac{2}{\|W\|}$

So in order to get the maximize width we have to maximize $\frac{2}{\|W\|}$ or

we have to minimize $\|W\|$ or

we have to minimize $\frac{1}{2}\|W\|^2$ where $\|W\| = W.W$.



Optimal Hyper-plane for linearly separable patterns

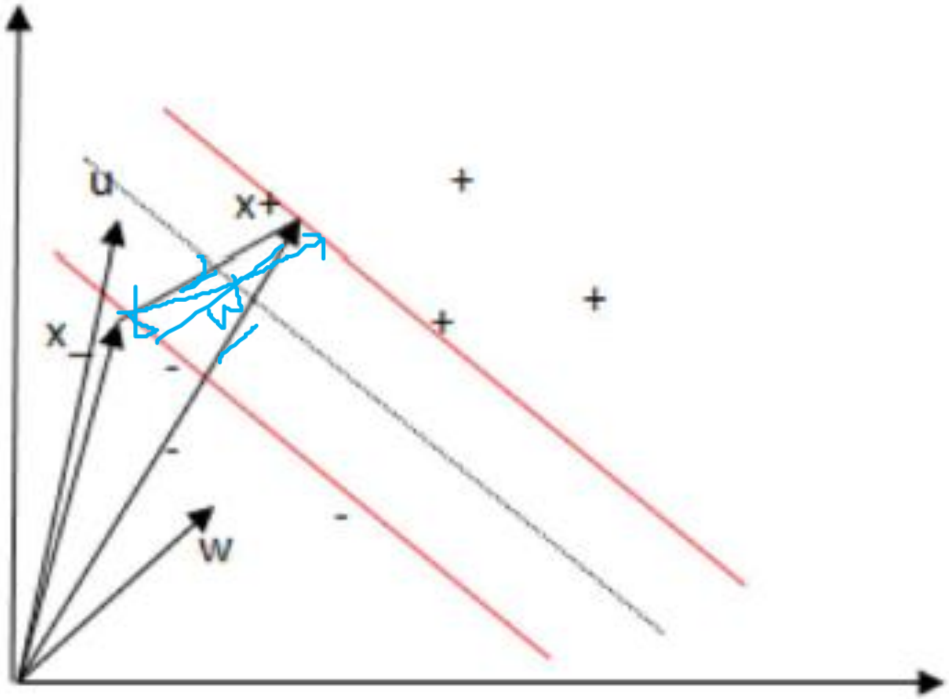
- For those examples that are on the lines we can say that $y_i(W.X_i + b) - 1 = 0$.

$$width = (x_+ - x_-).\frac{W}{\|W\|}$$

$$width = (1-b) + (1+b).\frac{1}{\|W\|}$$

$$width = \frac{2}{\|W\|}$$

And also for unknown vector that u, the decision rule is to $w.u \geq C$, and if $C = -b$ then we can write if $w.u + b \geq 0$ u is a positive example.



How to optimize the problem under constraints

- So we can now state the constrained optimization problem as follows: Given the training sample, $\{(x_i, y_i)\}_{i=1}^{n}$ find the optimum values of the weight vector W and bias b such that they satisfy the constraints $y_i(W.X_i + b) - 1 \geq 0$ and the weight vector minimizes the function $\frac{1}{2}\|W\|^2$.

  We solve the constrained problem by using the method of Lagrange Multipliers. Width of the boundaries is L and

  $$L(W, b, \alpha) = \underbrace{\frac{1}{2}W.W}_{*} - \underbrace{\sum_{i=1}^{N} \alpha_i \left(y_i \left(W.X_i + b\right) - 1\right)}_{+}$$

  where * = what you want to maximize or minimize, and + = constraints to be satisfied

- Only those multipliers that exactly satisfy the condition can have non zero α_i [1]

[1] Note that those examples which are closer to or effects to the boundary lines will have non-zero α_i and for those that lie far away from the boundary lines or the set of example which do not affect to the boundary lines will have zero α_i

How to optimize the problem under constraints

- In order to find the optimum value, we have to find the derivatives of L and set to zero.
- $L(W,b,\alpha) = \frac{1}{2}W.W - \sum_{i=1}^{N} \alpha_i \left(y_i \left(W.X_i + b\right) - 1\right)$
- $\frac{\partial L}{\partial W}(W,b,\alpha) = (W - \sum_{i=1}^{N} \alpha_i(y_i X_i)) = 0 \rightarrow W = \sum_{i=1}^{N} \alpha_i y_i X_i$
- $\frac{\partial L}{\partial b}(W,b,\alpha) = -\sum_{i=1}^{N} \alpha_i y_i = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$
- Let's try to find the optimum by substituting the value of W to L.
- $Q(W,b,\alpha) = optimize(L,b,\alpha) = \frac{1}{2}\sum_{i=1}^{N}(\alpha_i y_i X_i)\sum_{j=1}^{N}(\alpha_j y_j X_j) - \sum_{i=1}^{N} \alpha_i[y_i \sum_{j=1}^{N} \alpha_j y_j X_j]X_i + b) - 1]$
- $Q(W,b,\alpha_i) = \frac{1}{2}\sum_{i=1}^{N}(\alpha_i y_i X_i)\sum_{j=1}^{N}(\alpha_j y_j X_j) - \sum_{i=1}^{N}(\alpha_i y_i X_i)\sum_{j=1}^{N}(\alpha_j y_j X_j) + b\sum_{i=1}^{N} \alpha_i y_i - \sum_{i=1}^{N} \alpha_i$
- $Q(W,b,\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i \alpha_j y_i y_j X_i.X_j)$
- What we can notice here is that the optimization depends only on the dot product of the data set

How to optimize the problem under constraints

- Now we can restate our problem as follows:
  Given the training sample $\{(x_i, y_i)\}_{i=1}^{N}$, find the Lagrange Multipliers $\{\alpha i\}_{i=1}^{N}$ that maximize the objective function

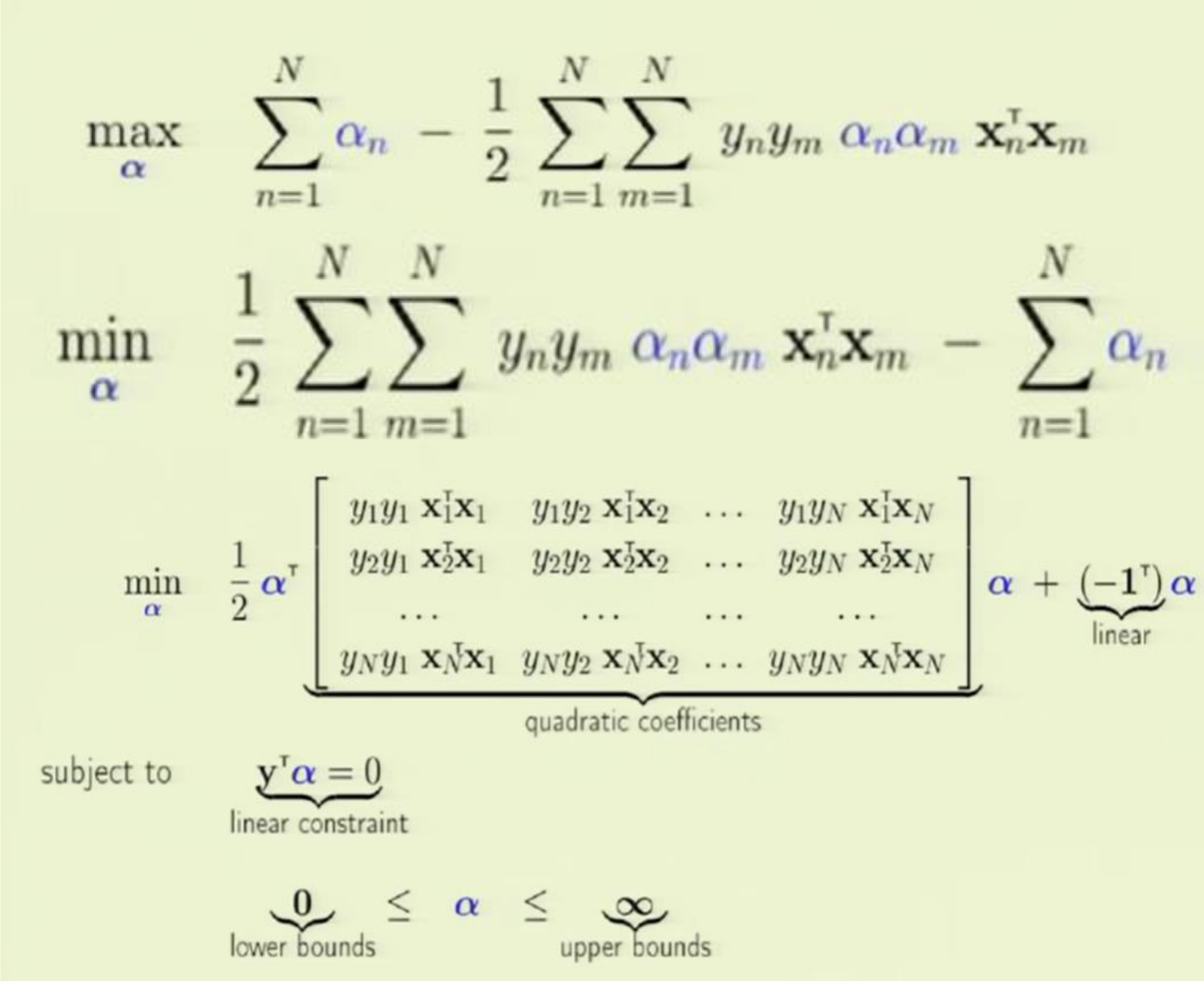
  $$Q(W, b, \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i \alpha_j y_i y_j X_i . X_j)$$

  subject to the constraints

  - $\sum_{i=1}^{N} \alpha_i y_i = 0$
  - $\forall i = 1(1)N, \alpha_i = 0$

- Accordingly having calculate the optimum Lagrange Multipliers denoted by α_i's.
- We compute the optimum weight vector W by $W = \sum_{i=1}^{N_1} \alpha_i y_i X_i$ where N_1 is the number of support vectors for which the Lagrange Multipliers α_i 's are all nonzero.
- Optimum bias b by $y_i(W.X_i + b) \geq 1 \rightarrow y_i(W.X_i + b) = 1$ because $y_i = 1$
- $b = 1 - \sum_{i=1}^{N_1} \alpha_i y_i X_i . X^s$
  Support vector X^s corresponds to any point in the training sample for which Lagrange Multipliers α_i's are nonzero.

The solution – Quadratic Programming



$$\max_{\boldsymbol{\alpha}} \quad \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_m$$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_m - \sum_{n=1}^{N} \alpha_n$$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \boldsymbol{\alpha}^{\mathsf{T}} \underbrace{\begin{bmatrix} y_1 y_1 \, \mathbf{x}_1^{\mathsf{T}} \mathbf{x}_1 & y_1 y_2 \, \mathbf{x}_1^{\mathsf{T}} \mathbf{x}_2 & \dots & y_1 y_N \, \mathbf{x}_1^{\mathsf{T}} \mathbf{x}_N \\ y_2 y_1 \, \mathbf{x}_2^{\mathsf{T}} \mathbf{x}_1 & y_2 y_2 \, \mathbf{x}_2^{\mathsf{T}} \mathbf{x}_2 & \dots & y_2 y_N \, \mathbf{x}_2^{\mathsf{T}} \mathbf{x}_N \\ \dots & \dots & \dots & \dots \\ y_N y_1 \, \mathbf{x}_N^{\mathsf{T}} \mathbf{x}_1 & y_N y_2 \, \mathbf{x}_N^{\mathsf{T}} \mathbf{x}_2 & \dots & y_N y_N \, \mathbf{x}_N^{\mathsf{T}} \mathbf{x}_N \end{bmatrix}}_{\text{quadratic coefficients}} \boldsymbol{\alpha} + \underbrace{(-\mathbf{1}^{\mathsf{T}})}_{\text{linear}} \boldsymbol{\alpha}$$

subject to $\underbrace{\mathbf{y}^{\mathsf{T}} \boldsymbol{\alpha} = 0}_{\text{linear constraint}}$

$$\underbrace{\mathbf{0}}_{\text{lower bounds}} \leq \boldsymbol{\alpha} \leq \underbrace{\infty}_{\text{upper bounds}}$$

QP hands us $\boldsymbol{\alpha}$

Solution: $\boldsymbol{\alpha} = \alpha_1, \cdots, \alpha_N$

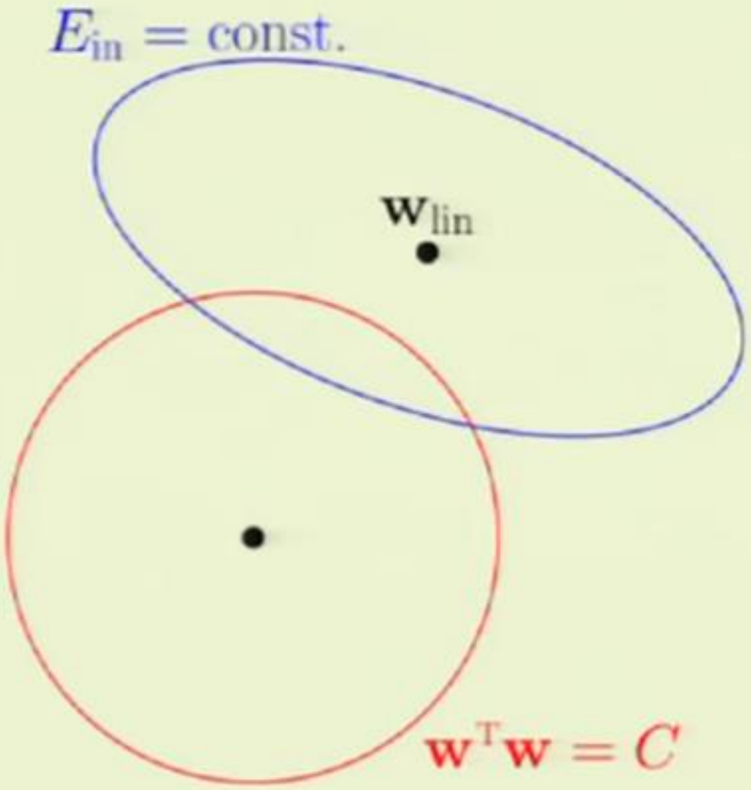
$$\Longrightarrow \quad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

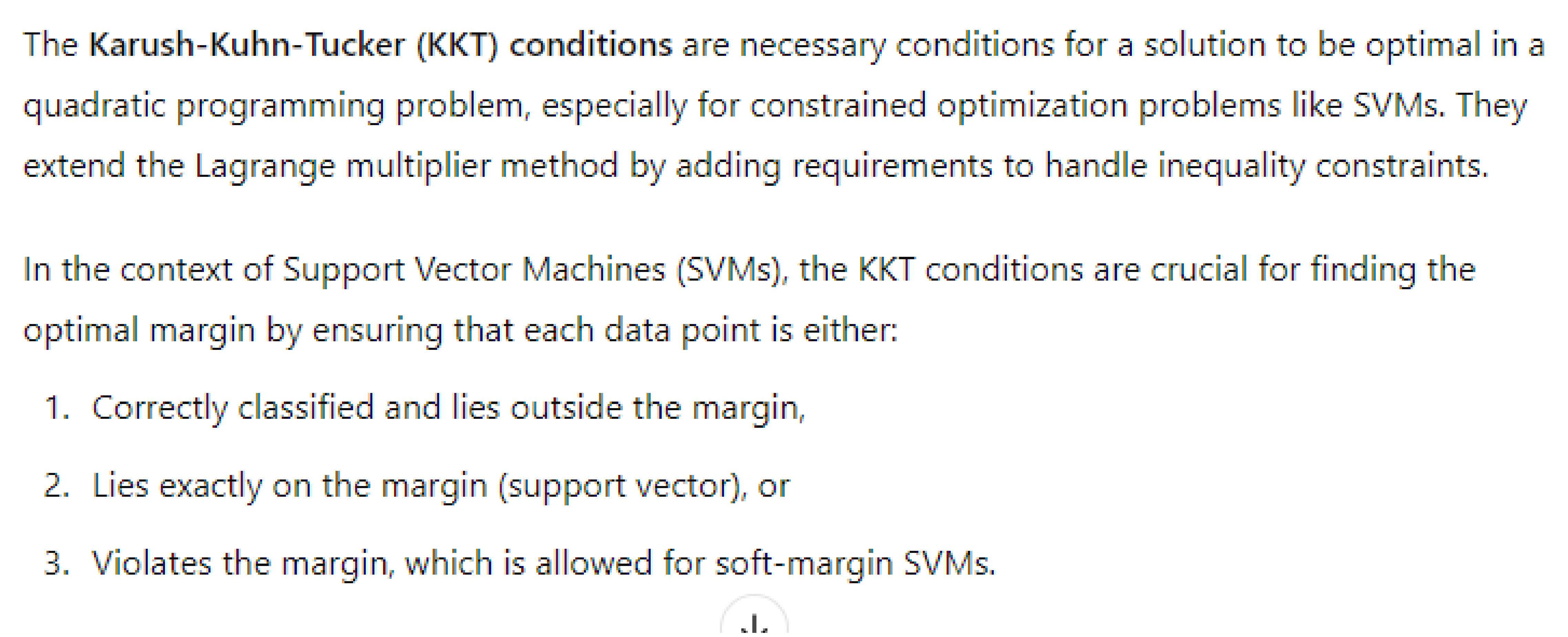
KKT condition: For $n = 1, \cdots, N$

$$\alpha_n \left(y_n \left(\mathbf{w}^{\mathsf{T}} \mathbf{x}_n + b \right) - 1 \right) = 0$$

We saw this before!

$\alpha_n > 0 \Longrightarrow \mathbf{x}_n$ is a $\boxed{\text{support vector}}$


$E_{\text{in}} = \text{const.}$
$\mathbf{w}_{\text{lin}}$
$\mathbf{w}^{\mathsf{T}}\mathbf{w} = C$



The **Karush-Kuhn-Tucker (KKT) conditions** are necessary conditions for a solution to be optimal in a quadratic programming problem, especially for constrained optimization problems like SVMs. They extend the Lagrange multiplier method by adding requirements to handle inequality constraints.

In the context of Support Vector Machines (SVMs), the KKT conditions are crucial for finding the optimal margin by ensuring that each data point is either:

1. Correctly classified and lies outside the margin,
2. Lies exactly on the margin (support vector), or
3. Violates the margin, which is allowed for soft-margin SVMs.

KKT Conditions Overview

Given a quadratic optimization problem:

$$\min_{w} \frac{1}{2}\|w\|^2$$

subject to the constraints:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

the KKT conditions are:

1. **Primal Feasibility**: The constraints must hold for the solution. For each i, we need:

$$y_i(w \cdot x_i + b) \geq 1$$

2. **Dual Feasibility**: The Lagrange multipliers α_i (associated with each constraint) must be non-negative:

$$\alpha_i \geq 0$$

3. **Complementary Slackness**: For each i, the product of α_i and the constraint should be zero:

$$\alpha_i \left(y_i(w \cdot x_i + b) - 1 \right) = 0$$

Testing

- For testing with a new data z:
- Compute $WZ + b = \sum_{i=1}^{N_1} \alpha_i y_i (X_i.Z) + b$ and classify Z as $y_i = +1$ if the sum is positive , $y_i = -1$ otherwise.
- Note that we do not need to form W explicitly.

Example

- Suppose we are given the following positively labeled data points in $\Re^2$:

$$\begin{pmatrix}3\\1\end{pmatrix}, \begin{pmatrix}3\\-1\end{pmatrix}, \begin{pmatrix}6\\1\end{pmatrix}, \begin{pmatrix}6\\-1\end{pmatrix}$$

  and

- the following negatively labeled data points in $\Re^2$:

$$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\-1\end{pmatrix}, \begin{pmatrix}-1\\0\end{pmatrix}$$

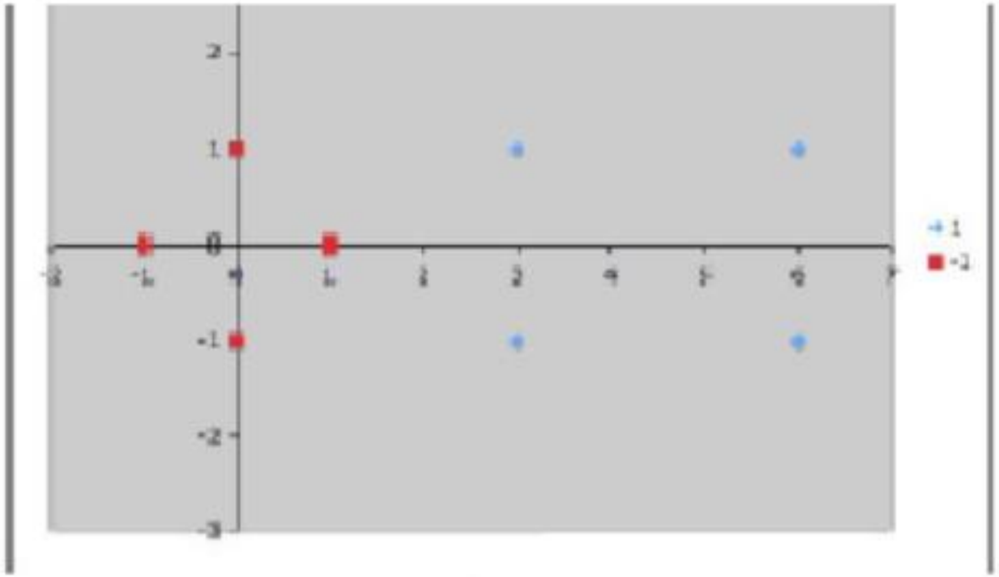


Figure 1: Sample data points in $\Re^2$. Blue diamonds are positive examples red squares are negative examples.

Lets define simple SVM that accurately discriminates the two classes. Since the data is linearly separable, we can use a linear SVM. it should be obvious that there are three support vectors:

$$s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \ s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$

- We will use vectors augmented with a 1 as a bias input, and for clarity we will differentiate these with an over-tilde. So, if s1 = (10), then $\tilde{s_1} = (101)$.

$$\tilde{s_1} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \tilde{s_2} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}, \ \tilde{s_3} = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

- Our task is to find values for the α_i such that,

$$\begin{aligned}
\alpha_i \tilde{s_1}.\tilde{s_1} + \alpha_2 \tilde{s_2}.\tilde{s_1} + \alpha_3 \tilde{s_3}.\tilde{s_1} &= -1 \\
\alpha_i \tilde{s_1}.\tilde{s_2} + \alpha_2 \tilde{s_2}.\tilde{s_2} + \alpha_3 \tilde{s_3}.\tilde{s_2} &= +1 \\
\alpha_i \tilde{s_1}.\tilde{s_3} + \alpha_2 \tilde{s_2}.\tilde{s_3} + \alpha_3 \tilde{s_3}.\tilde{s_3} &= +1
\end{aligned}$$

Example

- Our task is to find values for the α_i such that,

$$\begin{aligned} \alpha_i \tilde{s_1}.\tilde{s_1} + \alpha_2 \tilde{s_2}.\tilde{s_1} + \alpha_3 \tilde{s_3}.\tilde{s_1} &= -1 \\ \alpha_i \tilde{s_1}.\tilde{s_2} + \alpha_2 \tilde{s_2}.\tilde{s_2} + \alpha_3 \tilde{s_3}.\tilde{s_2} &= +1 \\ \alpha_i \tilde{s_1}.\tilde{s_3} + \alpha_2 \tilde{s_2}.\tilde{s_3} + \alpha_3 \tilde{s_3}.\tilde{s_3} &= +1 \end{aligned}$$

- Computing the dot product

For example, $\tilde{s_1}.\tilde{s_1} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = 1 \times 1 + 0 \times 0 + 1 \times 1 = 2$

$$\begin{aligned} 2\alpha_1 + 4\alpha_2 + 4\alpha_3 &= -1 \\ 4\alpha_2 + 11\alpha_2 + 9\alpha_3 &= +1 \\ 4\alpha_1 + 9\alpha_2 + 11\alpha_3 &= +1 \end{aligned}$$

$\alpha_1 = -3.5$ and $\alpha_2 = 0.75$ and $\alpha_3 = 0.75$

Example

- How to find the hyper-plane that discriminates the positive values?

$$\tilde{w} = \sum_{i=1}^{3} \alpha_i \tilde{s_i}$$

$$= -3.5 \times \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \times \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \times \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

- The bias b and w are:

$$w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } b = -2$$

Example

- Lets solve the equation,

$0 = W^T X + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and

$b = -2$, you will get

$$(1 \quad 0) \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - 2 = 0$$

$$1.x_1 + 0.x_2 - 2 = 0 \rightarrow x_1 = 2$$

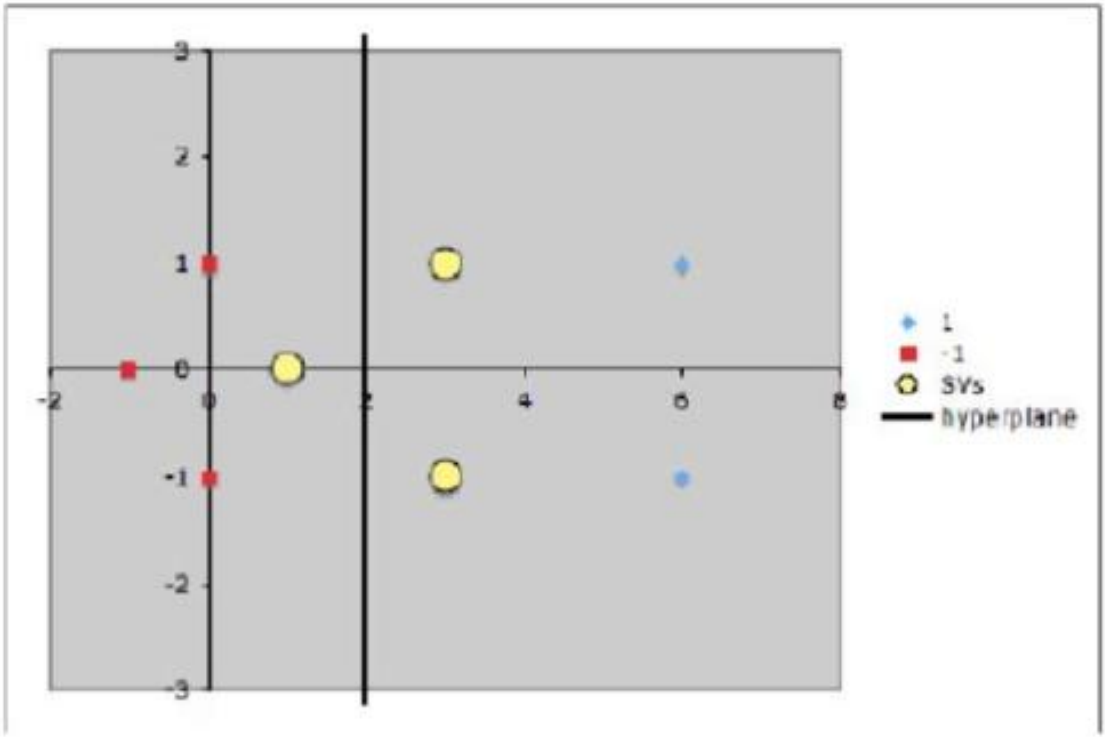
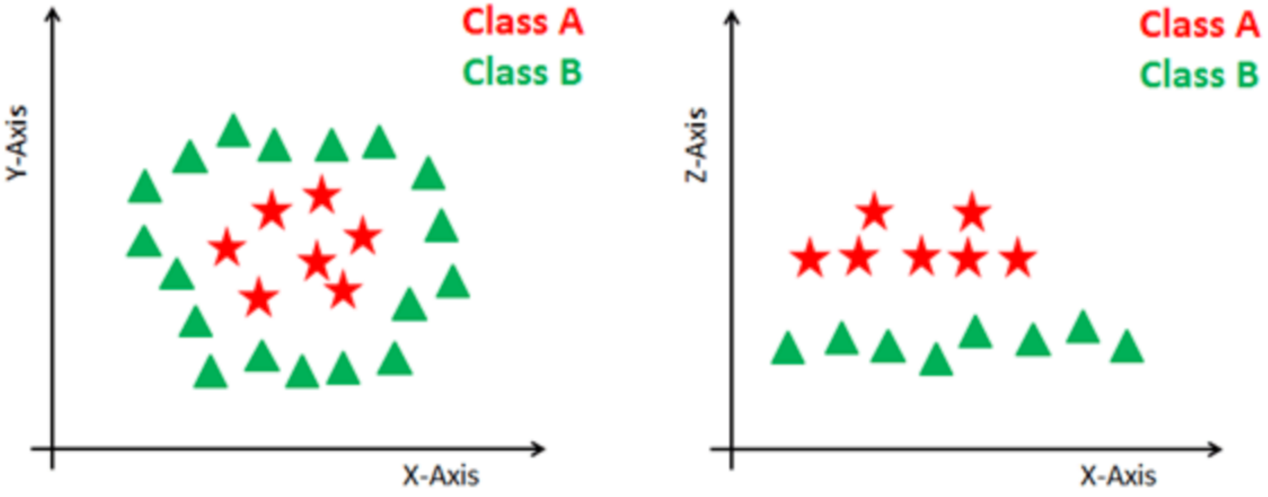
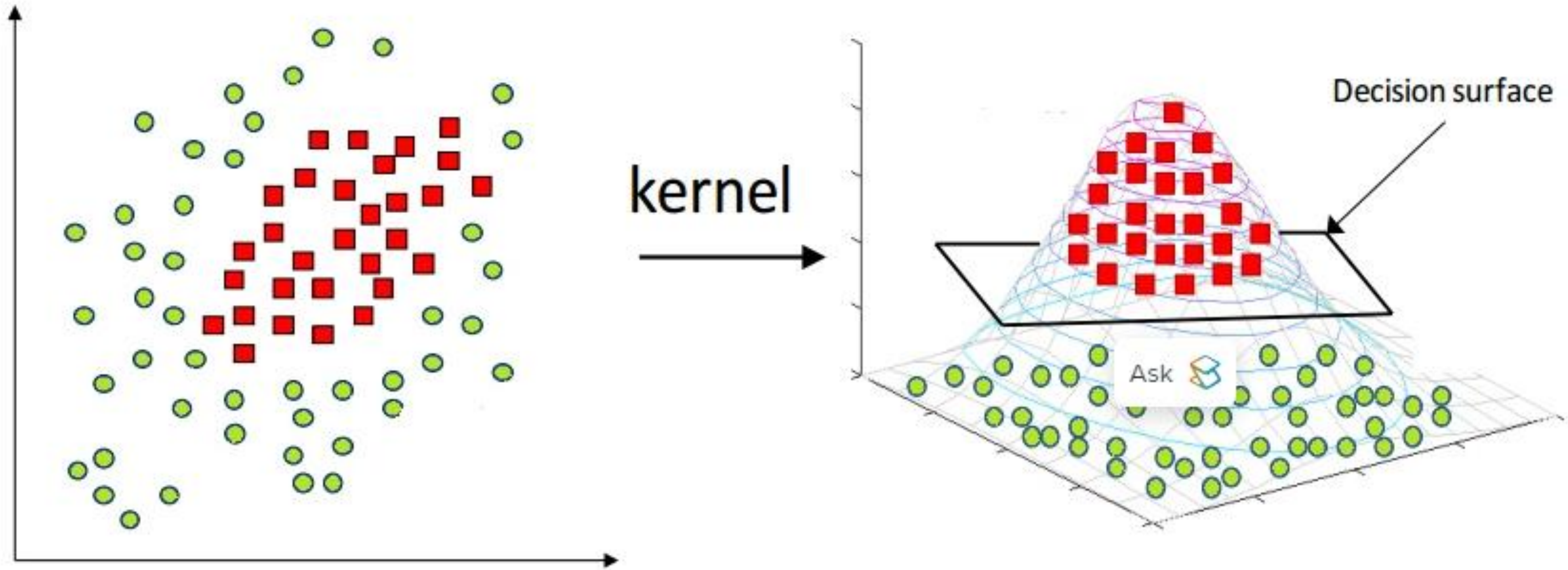


Figure 4: The discriminating hyperplane corresponding to the values c $-3.5, \alpha_2 = 0.75$ and $\alpha_3 = 0.75$.

Dealing with non-linear and inseparable planes



Kernels in Support Vector Machine



SVM Kernals

- The SVM algorithm is implemented in practice using a kernel.
- A kernel transforms an input data space into the required form.
- The kernel takes a low-dimensional input space and transforms it into a higher dimensional space.
- Simply, it converts linearly non-separable problem to separable problems by adding more dimension to it.
- Kernel trick helps to build a more accurate classifier.

SVM Kernals

- **Linear Kernel**
    - Can be used as a normal dot product in any two given observations.
    - The product between two vectors is the sum of the multiplication of each pair of input values.
- **Polynomial Kernel**
    - A more generalized form of the linear kernel.
    - The polynomial kernel can distinguish curved or nonlinear input space.
- **Radial Basis Function Kernel**
    - The Radial basis function kernel is a popular function commonly used in support vector machine classification.
    - RBF can map an input space in infinite dimensional space.

The Kernel-Trick to the SVM optimization problem

The optimization problem $Q(W,b,\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i\alpha_j y_i y_j X_i.X_j)$, the data points only appear as an inner product.

Inner Product Kernel :

- x denote a vector drawn from the input space of dimension m_0.
- Let $\Phi_{j}{}_{j=1}^{\infty}$ denote a set of nonlinear functions that, between them, transform the input space of dimension m_0 to a feature space of infinite dimensionality.
- Given this transformation, we may define a hyper-plane acting as the decision surface accordance with the formula, $\sum_{j=1}^{\infty} w_j \Phi_j$ where $w_{j=1}^{\infty}$ denotes large set of weights that transforms the feature space to the output space.
- It is in the output space where decision is made on whether the input vector X belongs to the negative examples or positive examples.
- $W = \sum_{i=1}^{N1} \alpha_i y_i \Phi(X_i)$ where the feature vector expressed as $\Phi(X_i) = [\Phi_1(x_1), \Phi_2(x_2), \ldots]^T$
- Therefore we can express the decision surface of the output space as $\sum_{j=1}^{\infty} w_j \Phi_j = 0$
- So after substituting the value for W, the above equation can be written as,

$$\sum_{i=1}^{N1} \alpha_i y_i \Phi_j(X_i)\Phi(X) = 0$$

The Kernel-Trick to the SVM optimization problem

$$\sum_{i=1}^{N1} \alpha_i y_i \underbrace{\Phi_j(X_i)\Phi(X)}_{k(x,x_j)} = 0$$

$$\sum_{i=1}^{N1} \alpha_i y_i k(x, x_j) = 0$$

- $k(x, x_j)$ is called inner-product kernel
- The kernel $k(x, x_j)$ is a function that computes the inner product of the images produced in the feature space under the embedding Φ of two data points in the input space.
- The function is symmetric about the center point X_i that is $K(X, X_i) = K(X_i, X); \forall X_i$ and it attains maximum value at the point $X = X_i$, and the total volume under the surface is a constant
- The kernel trick
  - $\sum_{i=1}^{N1} \alpha_i y_i k(x, x_j) = 0$, here we do not need to calculate the weight vector to calculate the decision surface. This is known as kernel trick.

Design of Support vector Machines as Kernel Machine

Examples for Kernel Machines

Polynomial (typical component of ϕ might be

$$K(\mathrm{q},\mathrm{q}') = (1 + \mathbf{q} \cdot \mathbf{q}')^k$$

Sigmoid (typical component $\tanh(q_1 + 3q_2)$)

$$K(\mathbf{q},\mathrm{q}') = \tanh(a\mathbf{q} \cdot \mathbf{q}' + b)$$

Gaussian RBF (typical component $\exp(-\frac{1}{2}(q_1 - 5)^2)$)

$$K(\mathrm{q},\mathrm{q}') = \exp(-\|\mathrm{q} - \mathbf{q}'\|^2/\sigma^2)$$

Optimal Hyper-plane for non-linearly separable patterns
Nonlinear Example (when Φ is non-trivial)

- Now suppose instead that we are given the following positively labeled data points in $\Re^2$

$$\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

- and the following negatively labeled data points in $\Re^2$

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

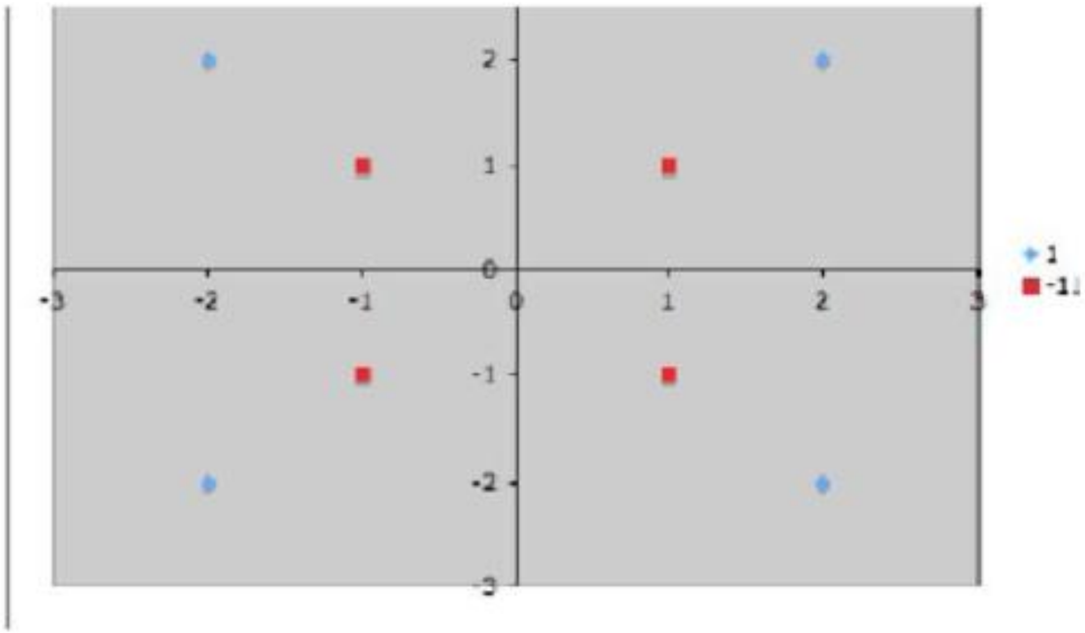
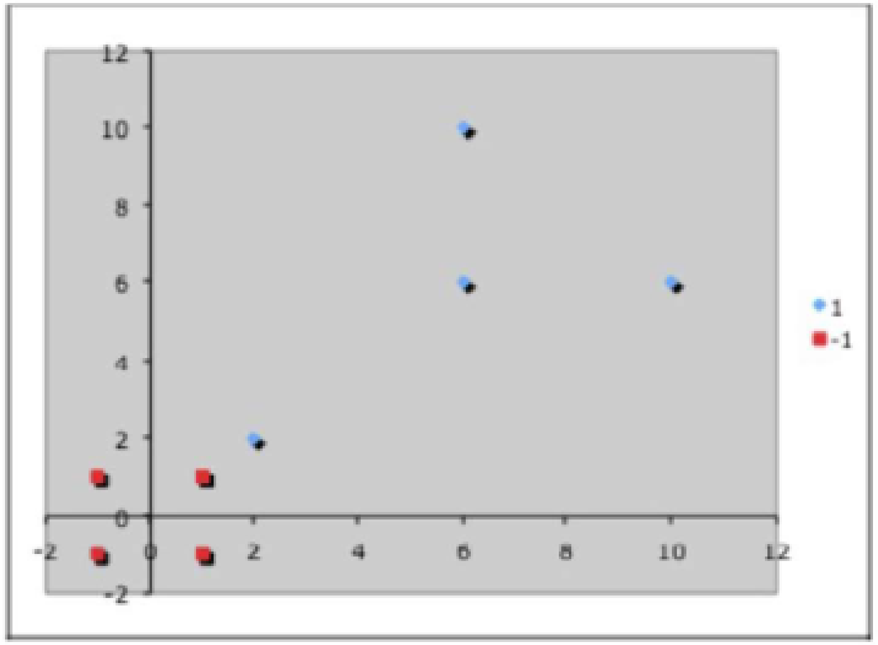


Figure 5: Nonlinearly separable sample data points in $\Re^2$. Blue diamo
positive examples and red squares are negative examples.

Optimal Hyper-plane for non-linearly separable patterns Nonlinear Example (when Φ is non-trivial)

- Goal is to discover a separating hyper-plane that accurately discriminates the two classes.
- Since data are not linearly separable, we use a nonlinear SVM (that is, SVM with mapping function Φ which is a nonlinear mapping from input space into some feature space)

$$\Phi_1\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases} \begin{pmatrix}4-x_2+|x_1-x_2|\\4-x_1+|x_1-x_2|\end{pmatrix} & \text{if } \sqrt{x_1^2+x_2^2}>2 \\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$



Optimal Hyper-plane for non-linearly separable patterns
Nonlinear Example (when Φ is non-trivial)

- Now we can re-write the the data in the feature space as, for positive examples

$$\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}6\\2\end{pmatrix},\begin{pmatrix}6\\6\end{pmatrix},\begin{pmatrix}2\\6\end{pmatrix}$$

- For negative examples,

$$\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix}$$

- Now the support vectors are:

$$s_1 = \begin{pmatrix}1\\1\end{pmatrix}, s_2 = \begin{pmatrix}2\\2\end{pmatrix}$$

Optimal Hyper-plane for non-linearly separable patterns Nonlinear Example (when Φ is non-trivial)

- Now lets add the bias term:

$$\tilde{s}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \tilde{s}_2 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

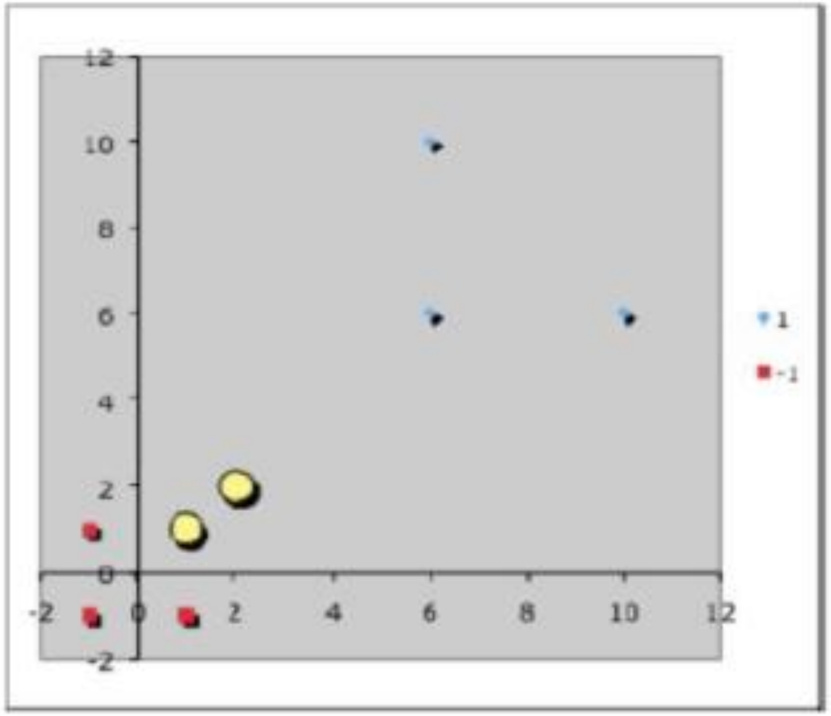
- so the set of equations is to solve is:

$$\alpha_1 \tilde{s}_1 . \tilde{s}_1 + \alpha_2 \tilde{s}_2 . \tilde{s}_1 = -1$$
$$\alpha_1 \tilde{s}_1 . \tilde{s}_2 + \alpha_2 \tilde{s}_2 . \tilde{s}_2 = +1$$

- That is,

$$3\alpha_1 + 5\alpha_2 = -1$$
$$5\alpha_1 + 9\alpha_2 = +1$$

- $\alpha_1 = -7$ and $\alpha_2 = 4$



Optimal Hyper-plane for non-linearly separable patterns
Nonlinear Example (when Φ is non-trivial)

- How to find the hyper-plane that discriminates the positive values?

$$\tilde{w} = \sum_{i=1}^{3} \alpha_i \tilde{s_i}$$

$$= -7 \times \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 4 \times \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

- The bias b and w are:

$$w = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } b = -3$$

Optimal Hyper-plane for non-linearly separable patterns
Nonlinear Example (when Φ is non-trivial)

- $y = w^T x + b$, with $w = (11)$ and $b = -3$

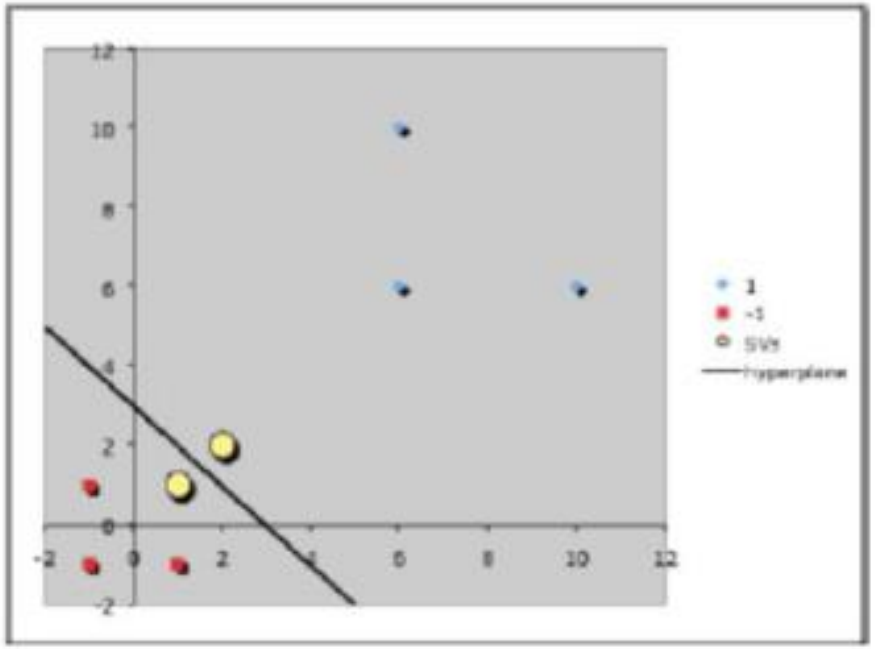
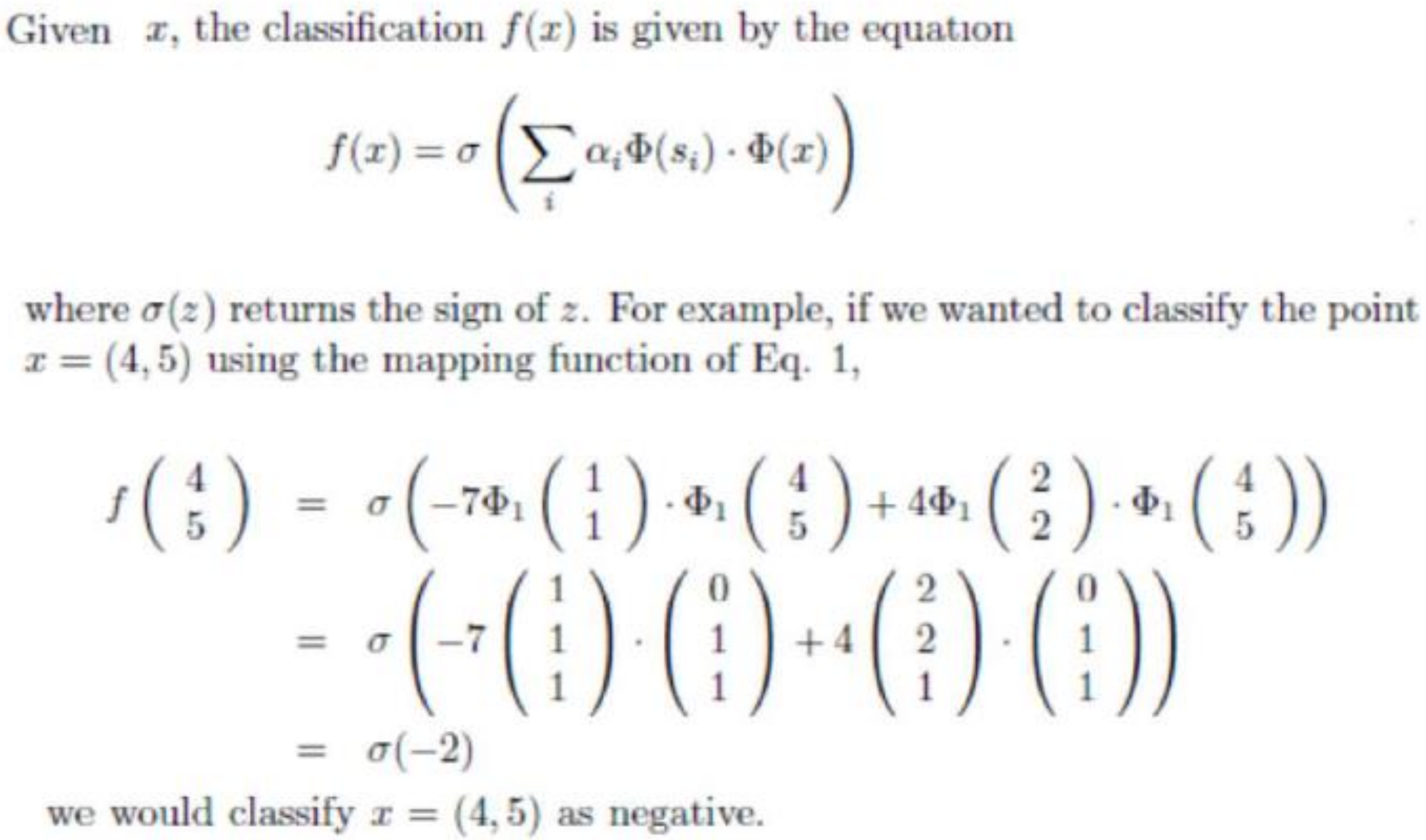


Figure 8: The discriminating hyperplane corresponding to the values $\alpha_1 = -7$ and $\alpha_2 = 4$

Optimal Hyper-plane for non-linearly separable patterns
How to classify data in non-trivial case



Given x, the classification $f(x)$ is given by the equation

$$f(x) = \sigma\left(\sum_i \alpha_i \Phi(s_i) \cdot \Phi(x)\right)$$

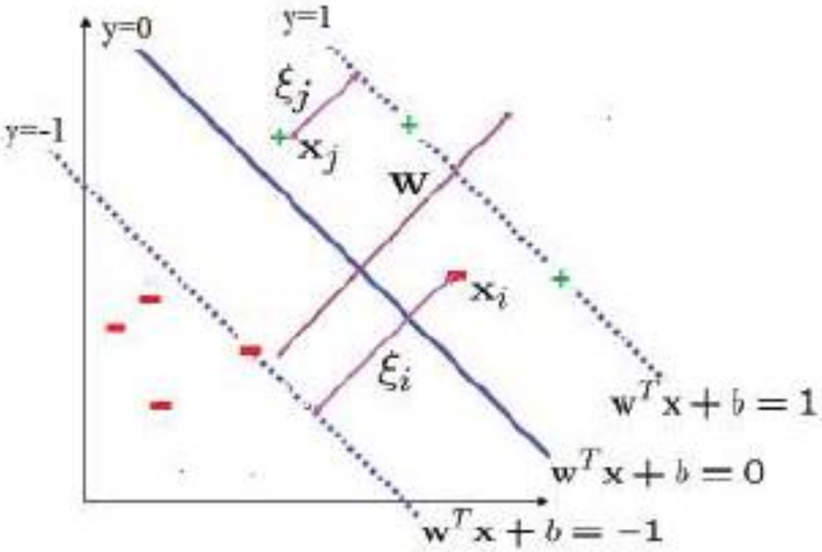
where $\sigma(z)$ returns the sign of z. For example, if we wanted to classify the point $x = (4, 5)$ using the mapping function of Eq. 1,

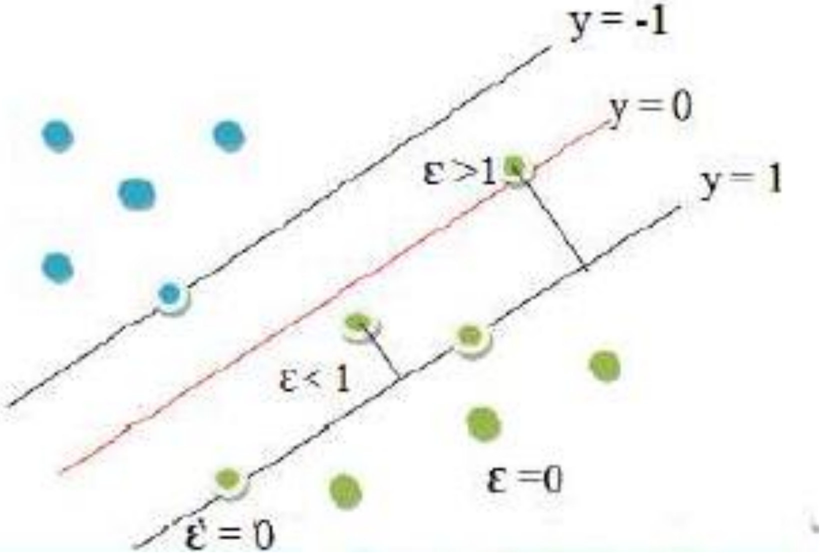
$$\begin{aligned} f\begin{pmatrix} 4 \\ 5 \end{pmatrix} &= \sigma\left(-7\Phi_1\begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot \Phi_1\begin{pmatrix} 4 \\ 5 \end{pmatrix} + 4\Phi_1\begin{pmatrix} 2 \\ 2 \end{pmatrix} \cdot \Phi_1\begin{pmatrix} 4 \\ 5 \end{pmatrix}\right) \\ &= \sigma\left(-7\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + 4\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}\right) \\ &= \sigma(-2) \end{aligned}$$

we would classify $x = (4, 5)$ as negative.

Optimal Hyper-plane for non-linearly separable patterns Theory

- Here we modify the SVM to allow some of the training points to be mis-classified.
- The data points are allowed to be on the wrong side of the margin boundary but with a penalty that increases with the distance from that boundary.
- Penalty is a linear function of this distance, and denoted by $\epsilon_n \geq 0$ where $n = 1, \ldots, N$
  - $\epsilon_n = 0$ for data points that are on or inside the correct margin boundary.
  - $\epsilon_n = |t_n - y_n|$ for other points.
  - data points on the decision boundary $y_n = 0$ and $\epsilon_n = 1$
  - Points for which $0 < \epsilon_n \leq 1$ lie inside the margin or on the correct side of the decision boundary.
  - Points for which $\epsilon_n \geq 1$ falls on the wrong side of the decision surface.
  - Thus Support Vectors are those that satisfy $y_i(W.X_i + b) - (1 - \epsilon) \geq 0$ if $\epsilon = 0$


y=0
y=1
y=-1
ξ_j
$\mathbf{x}_j$
$\mathbf{w}$
$\mathbf{x}_i$
ξ_i
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$



Optimal Hyper-plane for non-linearly separable patterns
Theory

- Our goal is to maximize the margin while softly penalize the points that lie on the wrong side of the margin boundary.
- $\therefore$ We minimize the parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.

$$C \sum_{n=1}^{N} \epsilon_n + \frac{1}{2} \|w\|^2$$

- Any point that is mis-classified has $\epsilon_N > 1$, $\sum_n \epsilon_n$ is an upper bound on the number of mis-classified points.

Given the training sample, $(x_i, y_i)_{i=1}^{N}$, find the optimum values of the weight vector W and bias b such that they satisfy the constraints $y_i(W.X_i + b) - (1 - \epsilon) \geq 0$ and the weight vector minimizes the function $C \sum_{n=1}^{N} \epsilon_n + \frac{1}{2} \|w\|^2$ where C is known as regulation parameter.

We can restate the above optimization as follows (after substituting the optimize weight vector and bias term to Lagrange Multipliers as such:

Optimal Hyper-plane for non-linearly separable patterns

Theory

Given the training sample $(x_i, y_i)_{i=1}^{N}$, find the Lagrange Multipliers $\alpha_i{}_{i=1}^{N}$ that maximize the objective function

$$Q(W,b,\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i \alpha_j y_i y_j X_i.X_j)$$

subject to the constraints

- $\sum_{i=1}^{N} \alpha_i y_i = 0$
- $\forall i = 1(1)N, 0 \leq \alpha_i \leq C$ where C is a user-specified positive parameter.
- Accordingly having calculate the optimum Lagrange Multipliers denoted by α_i's.
- We compute the optimum weight vector W by $W = \sum_{i=1}^{N_1} \alpha_i y_i X_i$ where N_1 is the number of support vectors for which the Lagrange Multipliers α_i 's are all nonzero.
- Optimum bias b by $y_i(W.X_i + b) \geq 1 \rightarrow y_i(W.X_i + b) = 1$ because $y_i = 1$
- For testing with a new data z:
- Compute $WZ + b = \sum_{i=1}^{N_1} \alpha_i y_i (X_i.Z) + b$ and classify Z as $y_i = +1$ if the sum is positive , $y_i = -1$ otherwise.

Main advantages and limitations of SVM

- Main advantages of SVMs:
    - Has rigorous theoretical foundation
    - Performs classification more accurately than most other methods in applications, especially for high dimensional data.
    - They can be applied also to non linear classification problems by using kernel functions. The "kernel trick" allows that the these problems are computationally tractable.
    - Different kernels can be plugged into the same learning machinery and studied independently of it.
- Main limitations of SVM:
    - Works only in a real valued space. For a categorical attribute, we need to convert its categorical values to numeric values.
    - Does only two class classification. For multiclass problems, some strategies can be applied, e.g., one against rest.
    - The hyper-plane produced by SVM is hard to understand by human users. The matter is made worse by kernel.