Machine Learning

Prof. Adil Khan

Objectives

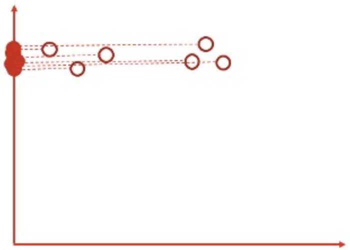
1. A quick recap of last week
2. Revisiting classification problems
3. How can we solve a classification problem using a “Separating Hyperplane”?
4. What are Support Vector Machines? What is their objective function? How is it motivated?
   - What is ”Margin”? Why do we need it? How can we use it to find the optimum separating hyperplane?
   - How to derive an expression for Margin? How to formulate the SVM’s objective function using the obtained expression?
   - How do we solve the objective function?
5. What are the two main Issues with SVMs?

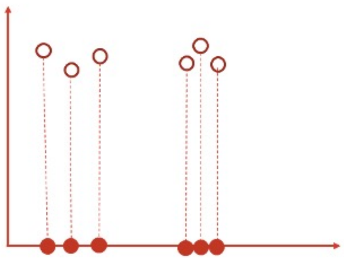
Recap (1)

High Dimensional Data is

- Difficult to visualize
- Difficult to analyze
- Difficult to understand – to get insight from the data (correlation and predictions)

Data Projection





Principal Component Analysis

- PCA reduces dimensionality by projecting data from a high dimensional space to a lower dimensional space
- It does so by using a set of vectors
- Vectors will be chosen such that they maximize the variance in the project space
- Furthermore, the vectors
  - Should be orthogonal
  - Have unit length
- Finally, data should have zero mean

Recap (2)

Given $X = \{(\boldsymbol{x}_i) | \boldsymbol{x}_i \in \mathbb{R}^p\}_{i=1}^{n}$, PCA Works as Follows

1. Transform the data to have zero mean by subtracting $\mu_{\boldsymbol{x}}$ from each point
2. Compute the sample covariance matrix C
3. Find p (eigenvector, eigenvalue) pairs of C
4. Find the eigenvectors corresponding to d highest eigenvalues $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_d$
5. Compute X' as $X' = XW$, where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_d]$

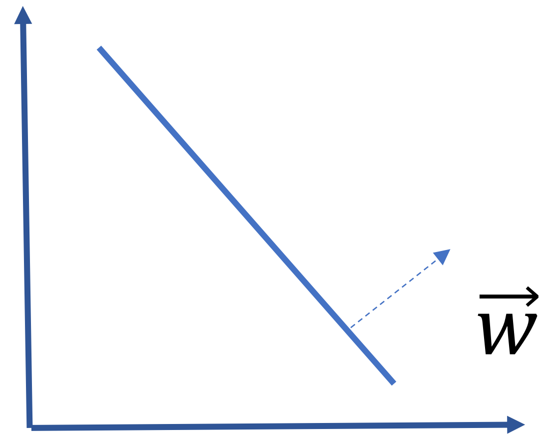
Hyperplane

Hyperplane

$$\vec{w}.\vec{x} + w_0 = 0$$

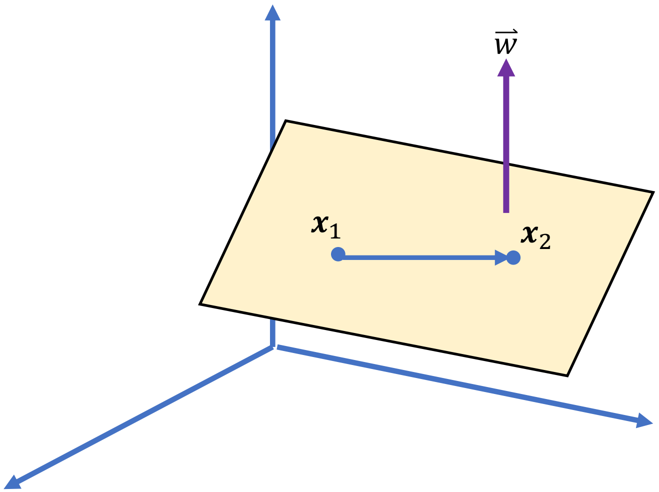
- A hyperplane in $\mathbb{R}^p$ dimensions is a set of points $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}\}$ that satisfy the following equation

$$w_o + w_1 x_1 + \cdots w_d x_p = 0$$



- If $p = 2$, a hyperplane is a line
- If $w_0 = 0$, the hyperplane goes through the origin
- The vector $\boldsymbol{w} = (w_1, w_2, \dots, w_p)$ is called the normal vector – it points in direction orthogonal to the surface of the hyperplane.

Normal Vector



$$\vec{w}.(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$$

$$\vec{w}.\boldsymbol{x}_1 - \vec{w}.\boldsymbol{x}_2 = 0$$

$$\vec{w}.\boldsymbol{x}_1 + w_0 = 0$$

$$\vec{w}.\boldsymbol{x}_2 + w_0 = 0$$

Separating Hyperplane

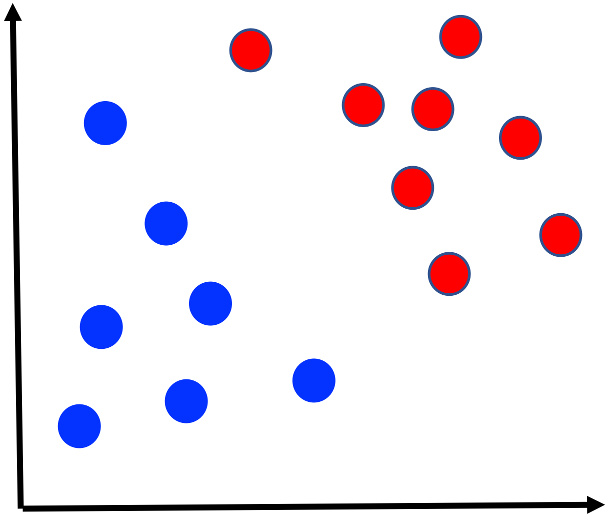
Binary Classification Problem

- Given a training dataset $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^p \times \{-1,1\}$

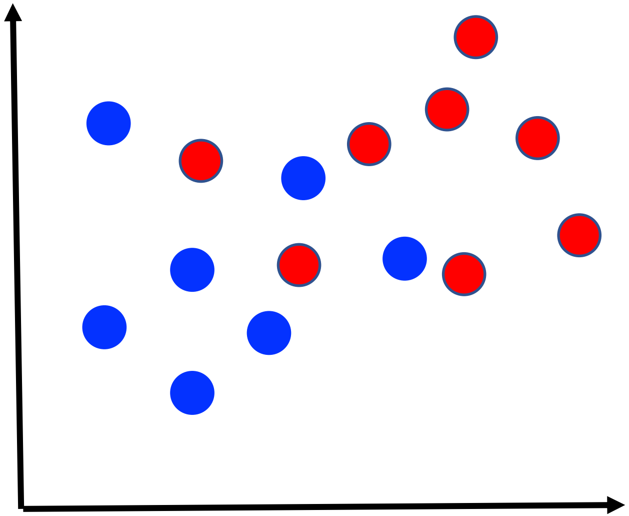
- We want to find a classifier $h(x): \mathbb{R}^p \rightarrow \{-1,1\}$

Two Cases

Linearly Separable



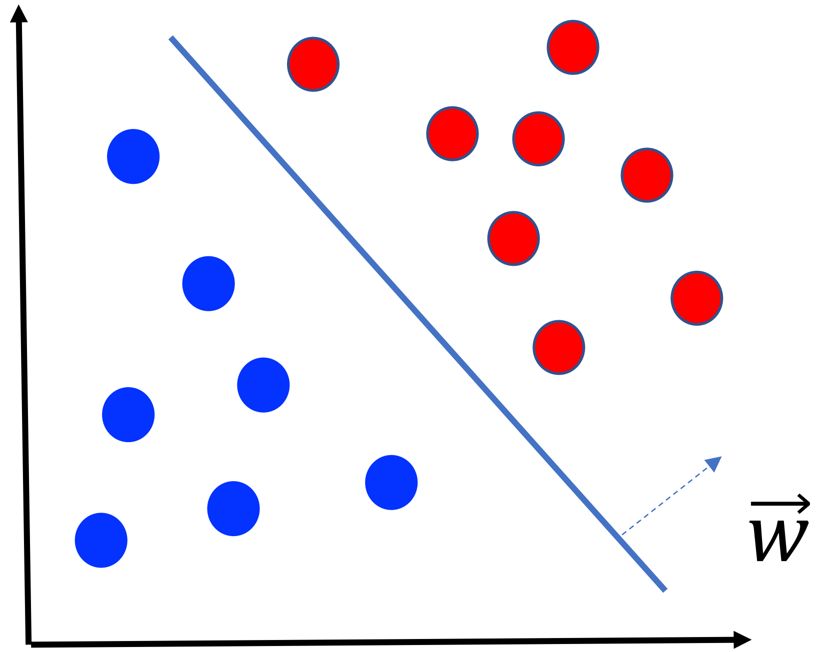
Non-Linearly Separable



For now, we will focus only on the linear cases

Separating Hyperplane

$$\vec{w}.\vec{x} + w_0 = 0$$



Separating Hyperplane (2)

$$w_o + w_1 x_1 + \cdots w_p x_p = 0$$

- Any point $x = (x_1, x_2, \ldots, x_p)$ that satisfies the above equation lies on the plane

Separating Hyperplane (3)

$$w_o + w_1 x_1 + \cdots w_p x_p = 0$$

1. However, $\boldsymbol{x}$ lies on positive side of the plane if

$$w_o + w_1 x_1 + \cdots w_p x_p > 0$$

2. Or the other if

$$w_o + w_1 x_1 + \cdots w_p x_p < 0$$

Separating Hyperplane (4)

- Thus a separating hyperplane has a property that

$$w_o + w_1x_1 + \cdots w_px_p > 0 \quad if\ y_i = 1$$

- And

$$w_o + w_1x_1 + \cdots w_px_p < 0 \quad if\ y_i = -1$$

- And our decision function for $\boldsymbol{x}_{new}$ is ,

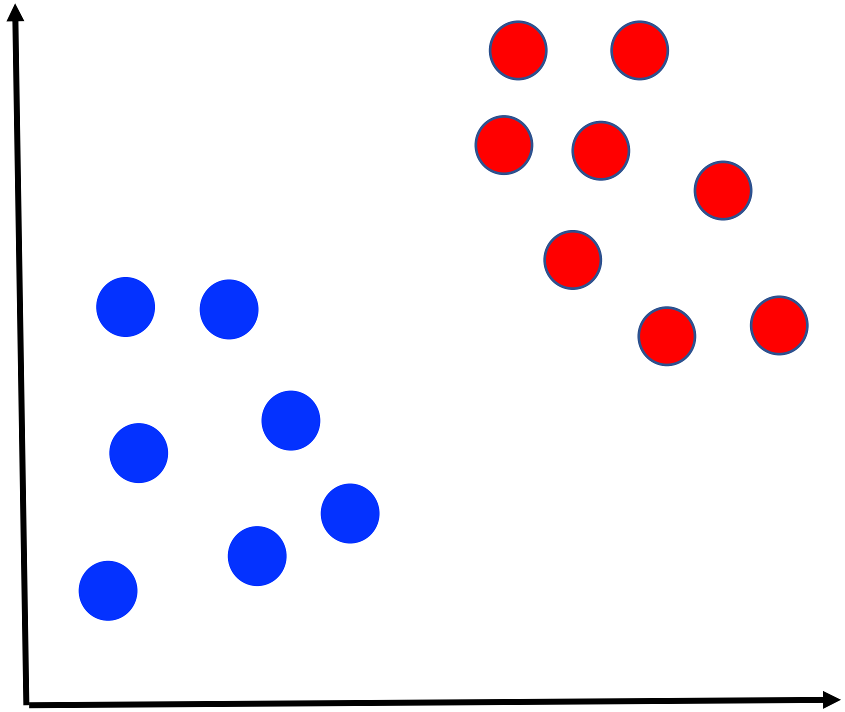
$$y_{new} = sign\left(w_o + w_1x_1 + \cdots w_px_p\right)$$

Separating Hyperplane (5)

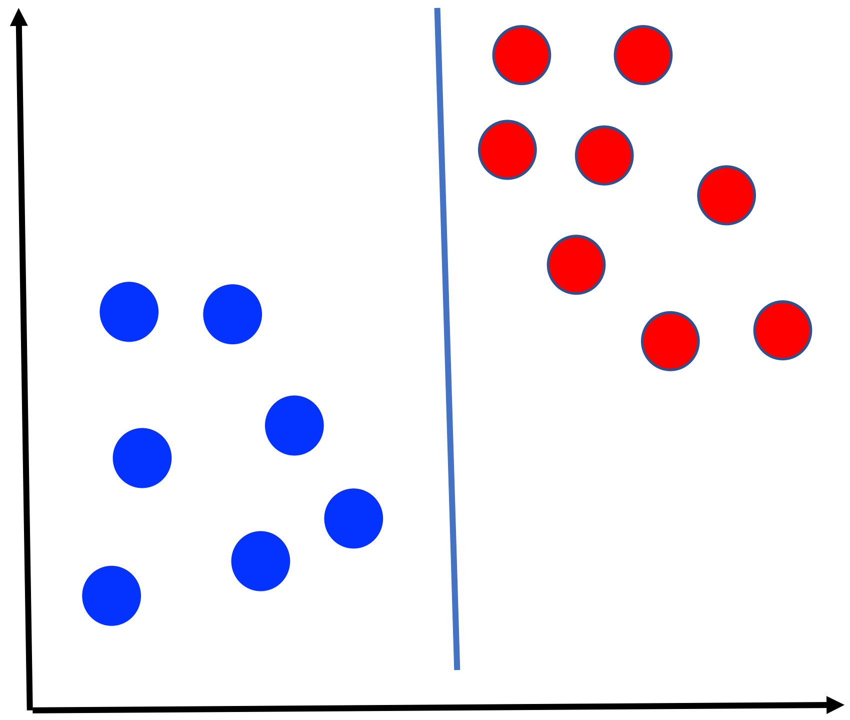
But there is one problem!

Infinite Many Separating Hyperplanes

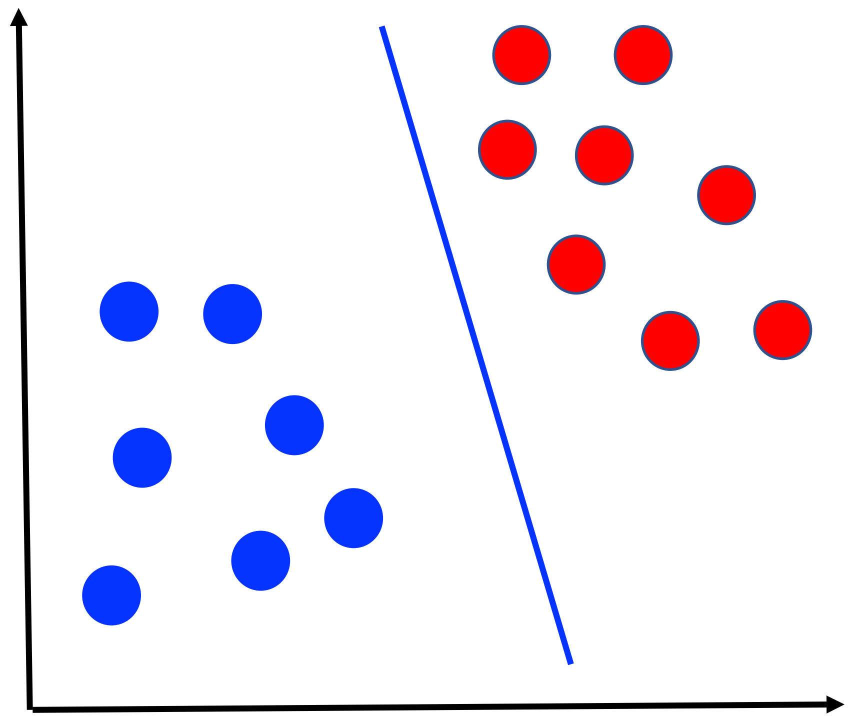
Choosing the Separating Hyperplane



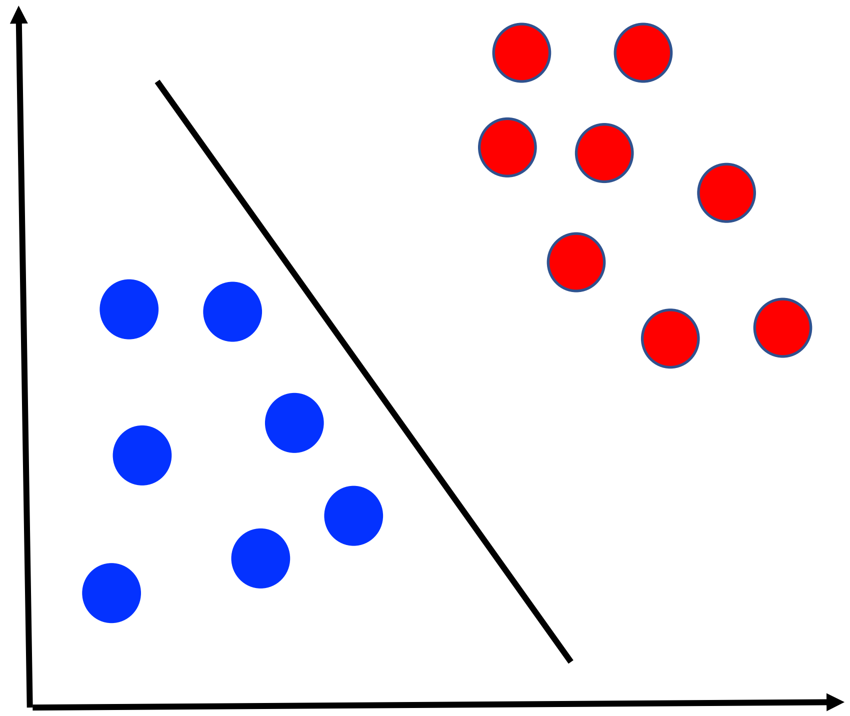
Should We Choose This One?



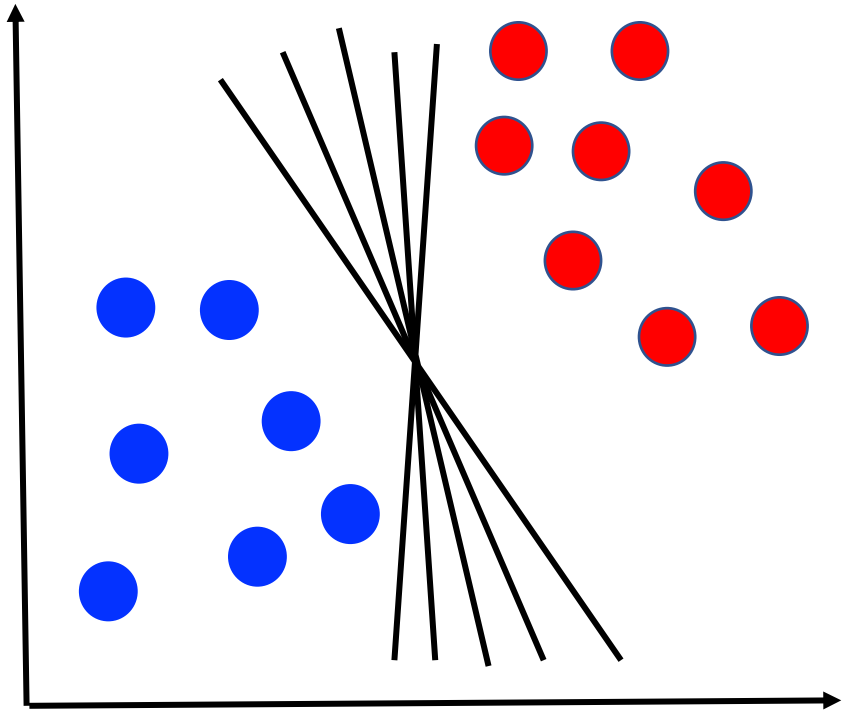
What About This One?



What About This One?



Infact, We have Infinite Many Such Hyperplanes!

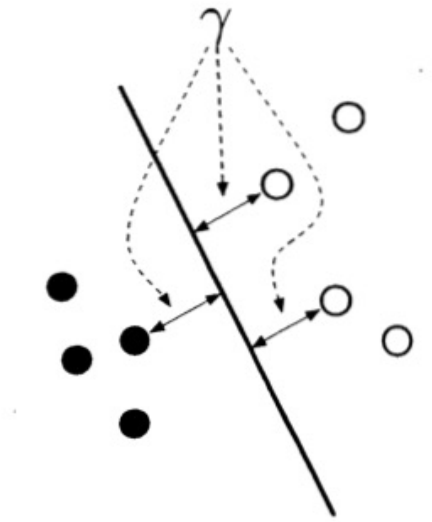


Infinite Many Hyperplanes

- Are all of them good?
- Or is one better than all others?
- To figure this out, let’s learn about the margin

Margin

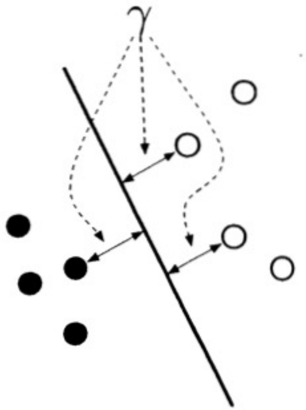
- Margin is the perpendicular distance from the decision boundary to the closest point on either side of the decision boundary



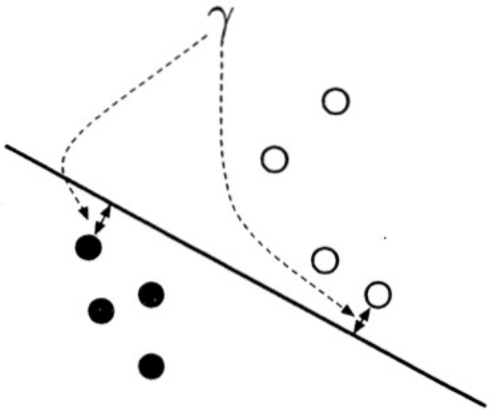
A First Course in Machine Learning, Chapter 5, Figure 5.12

Optimal Decision Bounday (or Separating Plane)

- One that miximizes the margin



(a) The decision boundary that maximises the margin



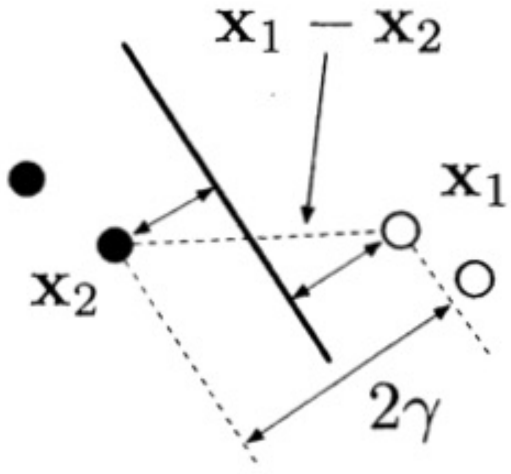
(b) A non-optimal decision boundary

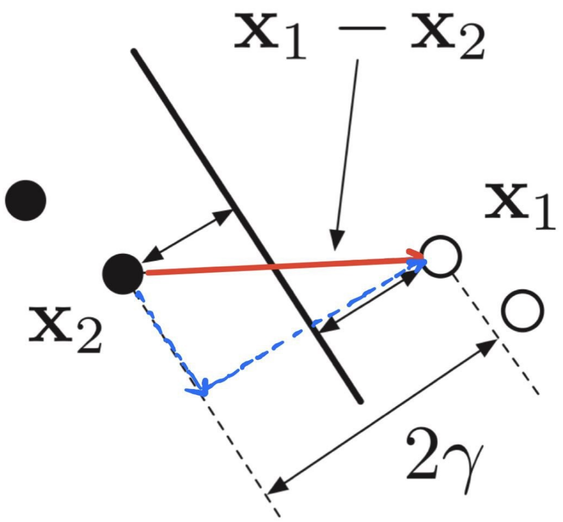
A First Course in Machine Learning, Chapter 5, Figure 5.12

Separating Hyperplane

Thus when finding a separating hyperplane we should seek one that maximizes the margin!

Mathematical Expression of Margin

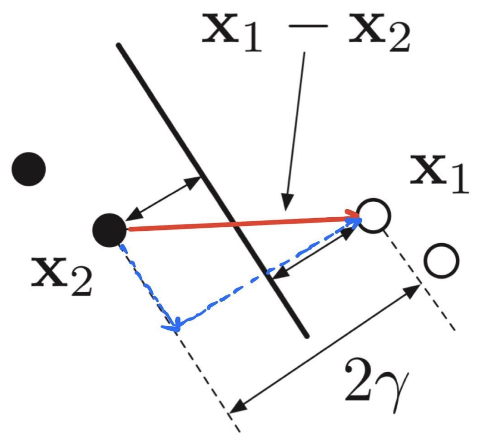




A First Course in Machine Learning, Chapter 5, Figure 5.13

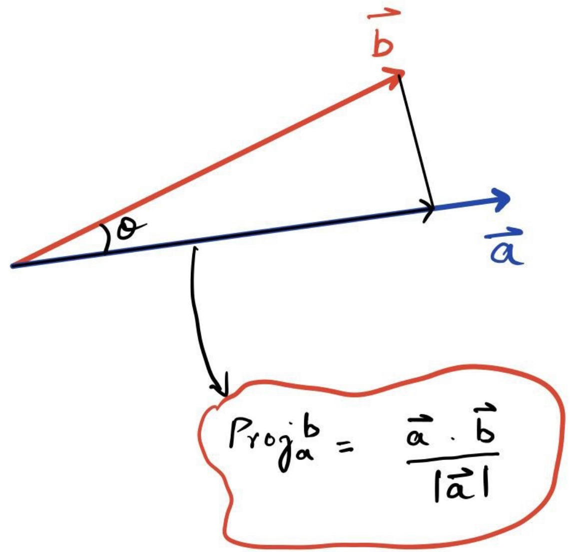
Mathematical Expression of Margin (2)

- Thus
- 2γ is equal to the component of the vector $(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ in the direction perpendicular to the boundary



A First Course in Machine Learning, Chapter 5, Figure 5.13

Recall Scalar Projection



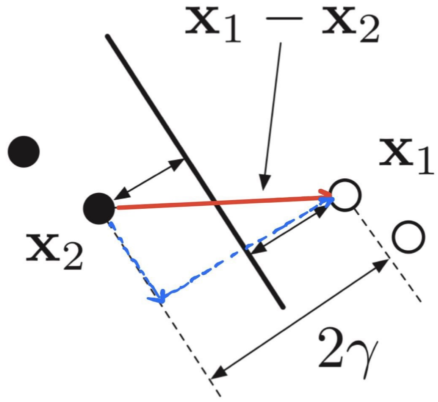
- This quantity is also called "the component of $\vec{b}$ in the direction of $\vec{a}$"

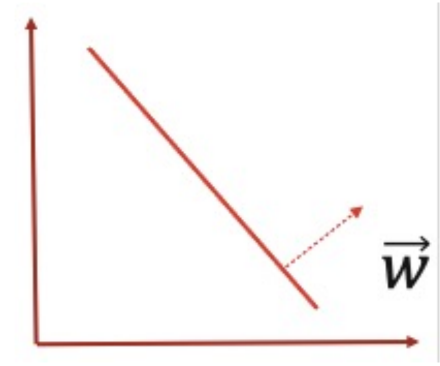
Mathematical Expression of Margin (3)

1. To get this expression, we just need to compute the "the component of vector $(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ which is perpendicular to the boundary"

2. We know that $\vec{w}$ is perpendicular to the boundary

3. Thus, we can simply compute the scalar projection of vector $(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ onto $\vec{w}$





Mathematical Expression of Margin (4)

- Thus

$$2\gamma = \frac{1}{\|\boldsymbol{w}\|}\boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2)$$

$$= \frac{1}{\|\boldsymbol{w}\|}\left(\boldsymbol{w}^T\boldsymbol{x}_1 - \boldsymbol{w}^T\boldsymbol{x}_2\right)$$

$$= \frac{1}{\|\boldsymbol{w}\|}\left(\boldsymbol{w}^T\boldsymbol{x}_1 + w_0 - \boldsymbol{w}^T\boldsymbol{x}_2 - w_0\right)$$

Mathematical Expression of Margin (5)

- Thus

$$2\gamma = \frac{1}{\|\boldsymbol{w}\|}\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{1}} + w_0 - (\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{2}} + w_0)\right)$$

- Now, recall our decision function

$$y_{new} = sign\left(w_o + w_1 x_1 + \cdots w_p x_p\right)$$

- It only cares about sign and does not care about the value

- Thus we can decide to fix the scaling of $\boldsymbol{w}$ and w_o such that

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

Mathematical Expression of Margin (6)

- Thus

$$2\gamma = \frac{1}{\|\boldsymbol{w}\|}(\mathbf{1} - (-\mathbf{1}))$$

$$= \frac{1}{\|\boldsymbol{w}\|}(\mathbf{1} + \mathbf{1})$$

$$= \frac{2}{\|\boldsymbol{w}\|} \Rightarrow \boxed{\gamma = \frac{1}{\|\boldsymbol{w}\|}}$$

Maximizing the Margin

- Thus to maximize the margin, we must maximize $\frac{1}{\|\boldsymbol{w}\|}$

- However, there are some constraints

$$y_i(w_o + w_1 x_1 + \cdots w_p x_p) \geq 1$$

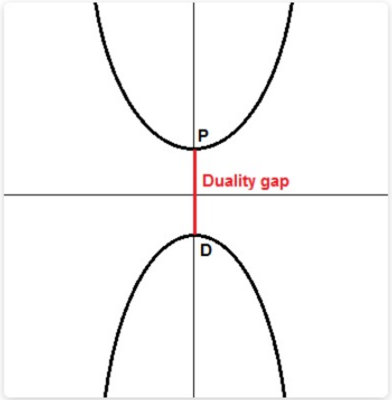
Maximizing the Margin (2)

- Thus our learning objective becomes

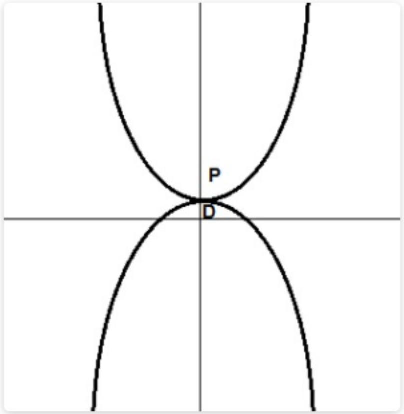
$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \ \frac{1}{\|\boldsymbol{w}\|}$$

subject to $y_i(w_o + w_1 x_1 + \cdots w_p x_p) \geq 1$ for all i

Recall Dual Optimization Problems



weak duality holds



strong duality holds

Maximizing the Margin (3)

- Practically, it is easier to solve the following (equivalent) optimization problem

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{w}\|^2$$

subject to $y_i\left(w_o + w_1 x_1 + \cdots w_p x_p\right) \geq 1$ for all i

Thus

- In Support Vector Machines

- Given a data set $\mathbb{D} = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1,1\}\}_{i=1}^{n}$, we solve the following

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{w}\|^2$$

subject to $y_i(w_o + w_1 x_1 + \cdots w_p x_p) \geq 1$ for all i

Remember Lagrange Multipliers!!

Thus (2)

- In Support Vector Machines

- Given a data set $\mathbb{D} = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1,1\}\}_{i=1}^n$, we solve the following

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i\left(y_i\left(\boldsymbol{w}^T \boldsymbol{x}_i + w_0\right) - 1\right)$$

subject to $\alpha_i \geq 0$ for all i

What Have We Learned So Far?

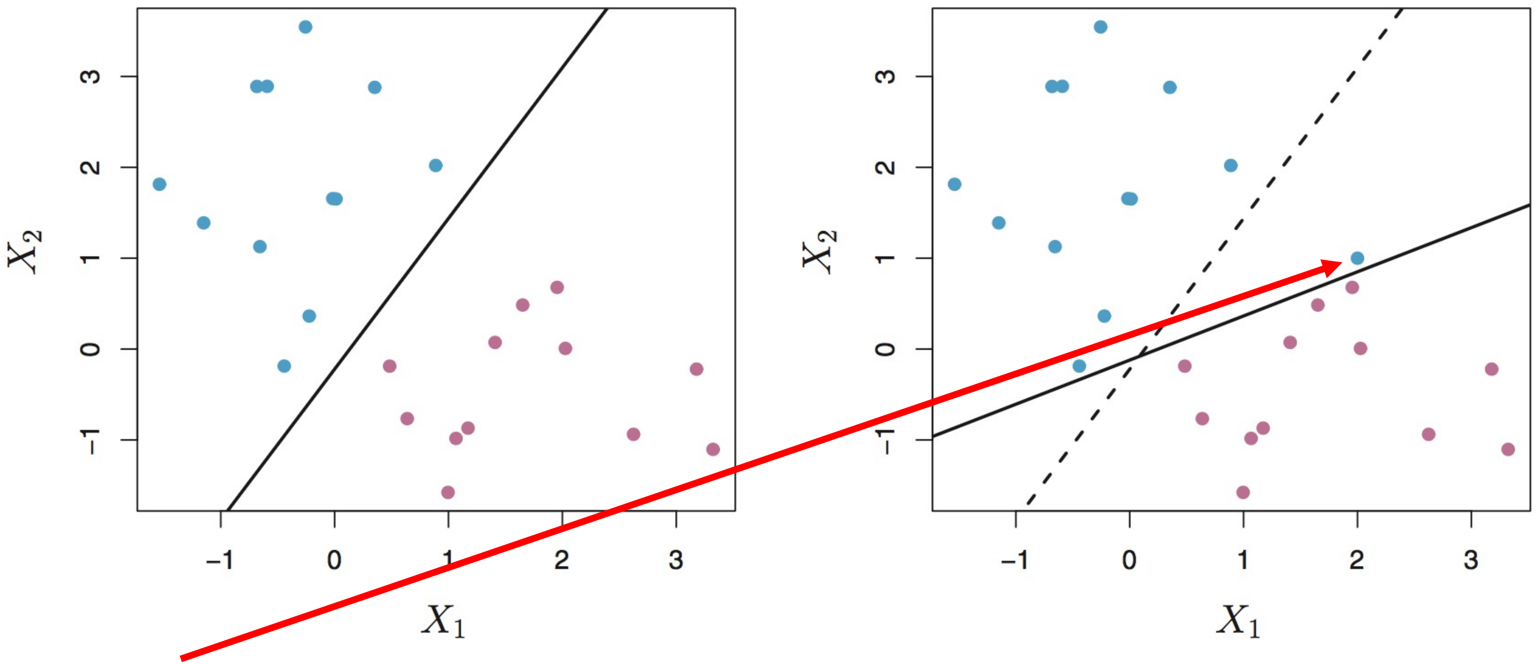
1. How a hyperplane can be used to separate linearly separable data?
2. But there are inifite many such hyperplanes
3. The best among them is the one that has the maximum margin
4. Support Vector Machines find that hyperplane by solving the following objective

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + w_0 \right) - \mathbf{1} \right)$$

subject to $\alpha_i \geq 0$ for all i

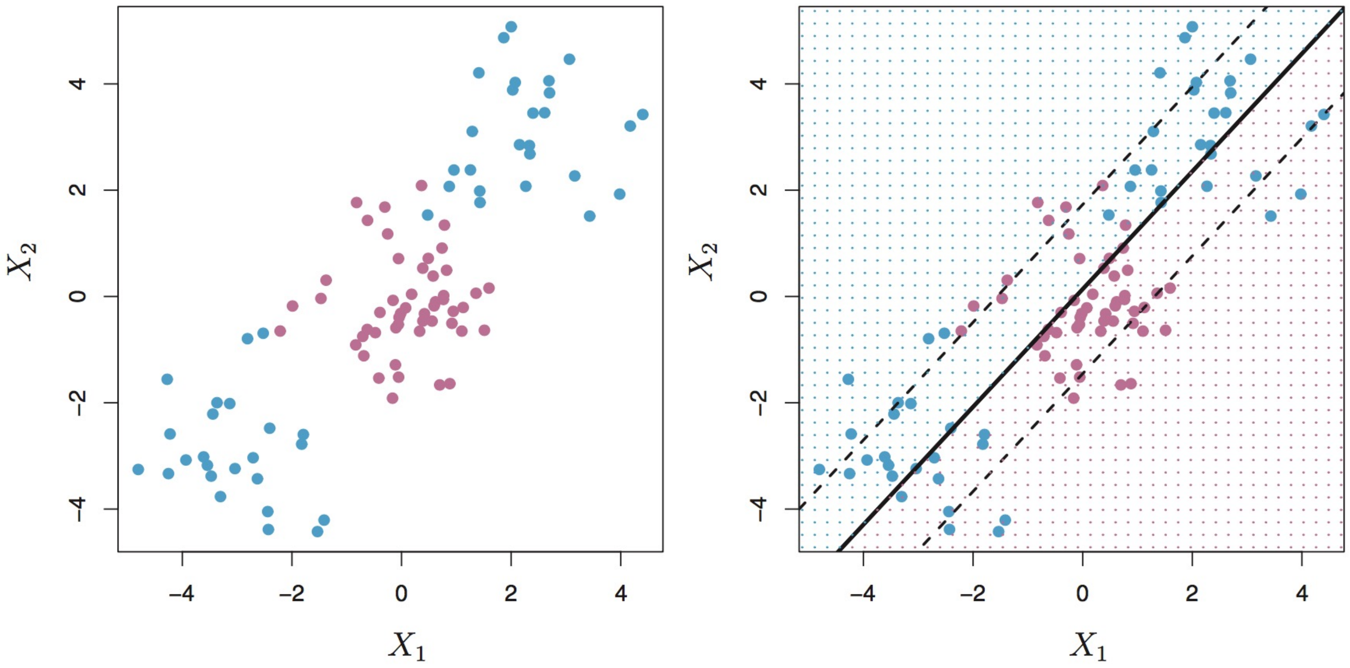
Issues with SVM

Issue (1): Sensitivity to Noise



An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane. This is kind of overfitting.

Issue (2): Non-linear Data



Soft Margin SVM

Why Does SVM Overfit?

1. To understand that, we will look at the constraints of the original optimisation problem

$$y_i\left(w_o + w_1 x_1 + \cdots w_p x_p\right) \geq 1 \text{ for all } i$$

2. This means all training data must sit on the right side of the decision boundary

Soft Margin SVM

1. To reduce overfitting, we must allow points to lie on the wrong side of the (margin or boundary)

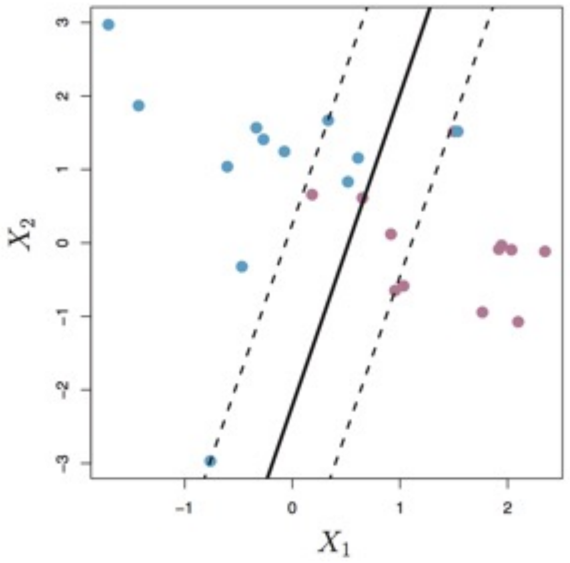
2. Thus we need to realx the constraints

$$y_i(w_o + w_1 x_1 + \cdots w_p x_p) \geq 1 - \xi_i \quad \text{for all } i$$

$$\text{where } \xi_i \geq 0$$

Slack Variables

- The slack variable ξ_i tells us where the i-th observation is located, relative to the hyperplane and relative to the margin.
- If $\xi_i = 0$ then the sample is on the correct side of the margin
- If $\xi_i > 0$ then the sample is on the wrong side of the margin
- If $\xi_i > 1$ then the sample is on the wrong side of the hyperplane



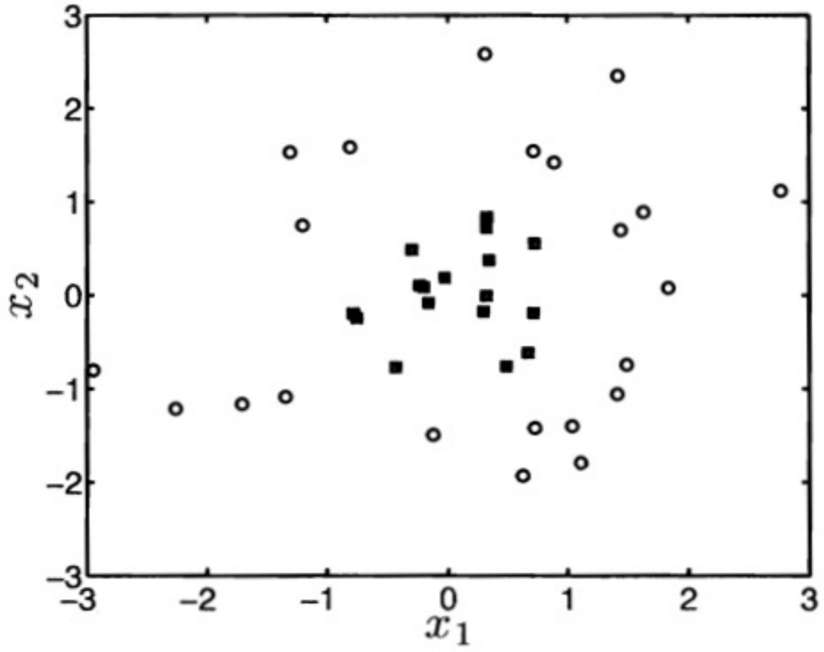
Soft Margin SVM Objective

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \ \frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{w}\|^{2} + \mathrm{C}\sum_{i=1}^{n} \xi_i$$

subject to $\xi_i \geq 0, y_i\left(w_o + w_1 x_1 + \cdots w_p x_p\right) \geq 1 - \xi_i$ for all i

Kernel SVM

A Non-linear Classification Example



A First Course in Machine Learning, Chapter 5, Figure 5.17

Recall: Extending Linear Regression

- What did we do to get better results where the underlying relationship between the response variable and the predictors was linear?

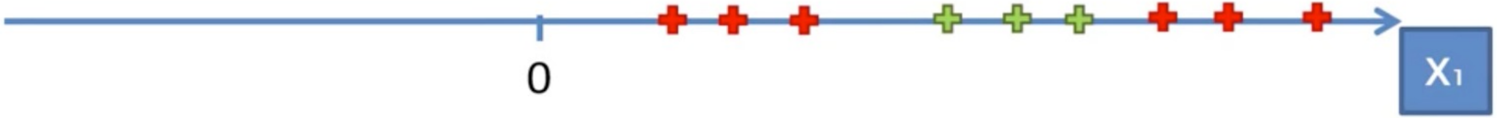
- Introduced polynomial features

- To apply SVM for non-linear problems, we do a similar thing:

  - Project our data to a high-dimensional space
  - Apply linear SVM there

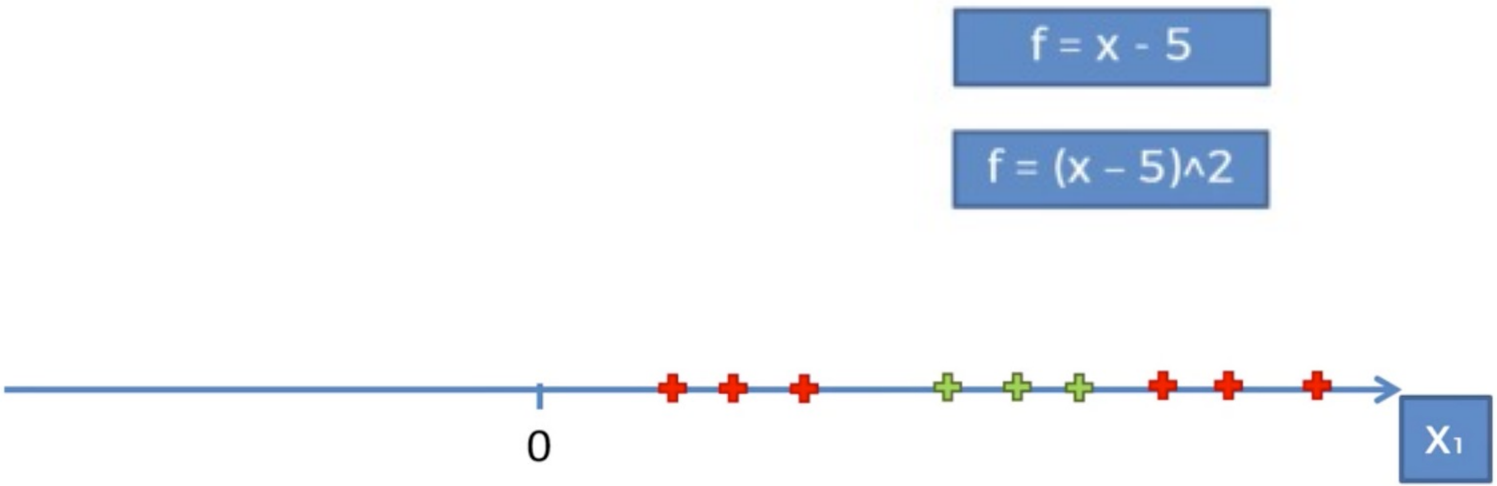
Example

- Data lies along a single dimension
- Not linearly separable



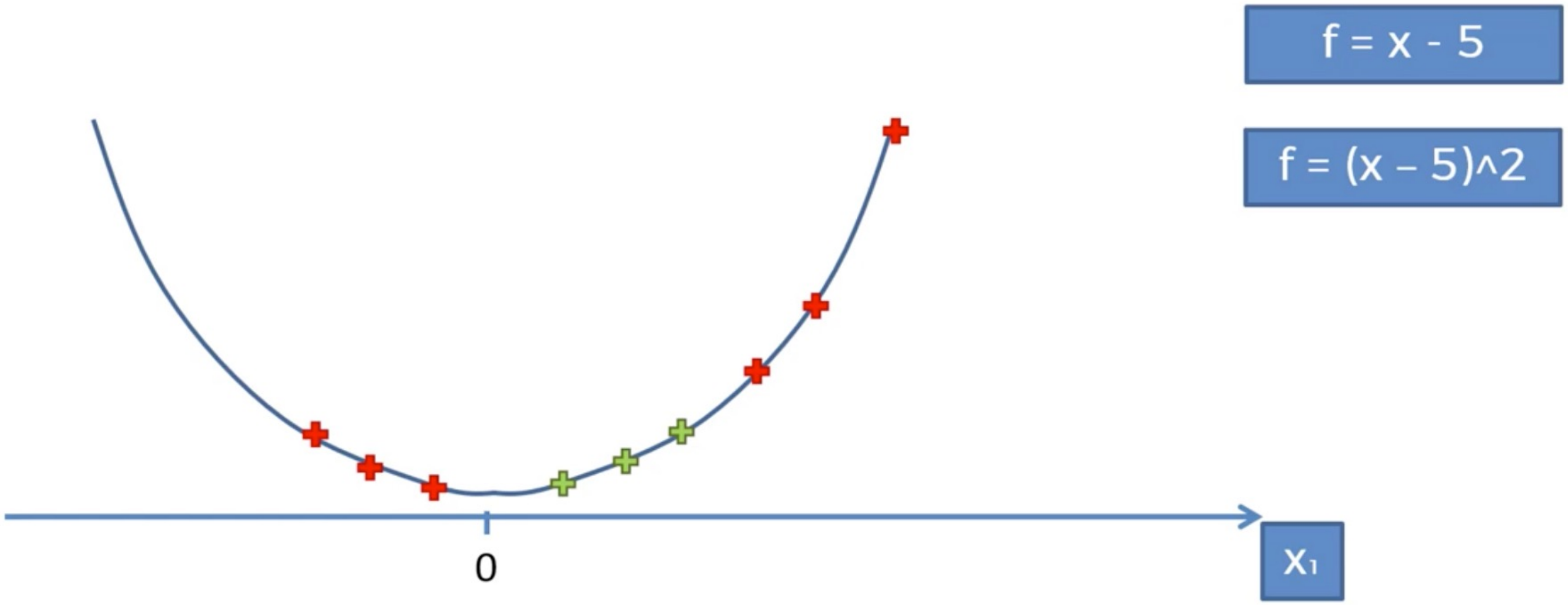
Example (2)

- Let’s apply the following transformation to our data



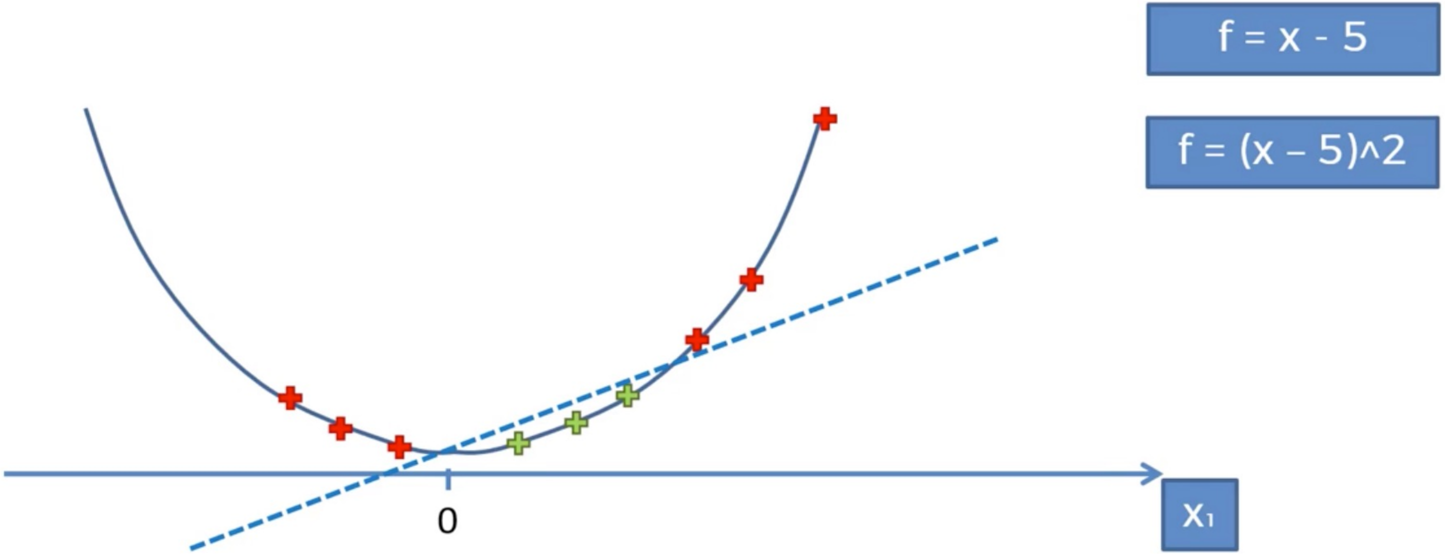
Example (3)

- The data after the transformation



Example (4)

- Which can now be solved using a linear hyperplane



Summary

1. Take the non-linearly separable data and transform the data such that it becomes linearly separable

2. Invoke the SVM to find the best (linear) decision boundary

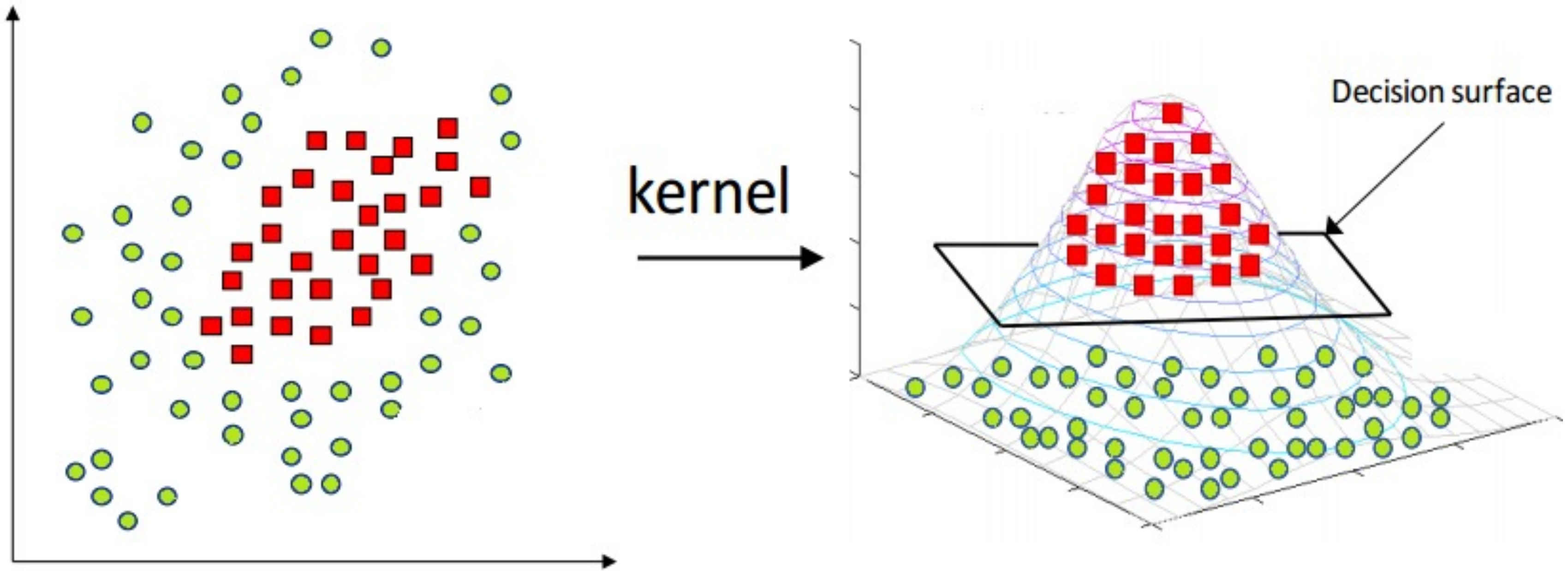
3. Project all of it back to the original space to get the non-linear decision boundary

But how do we determine what transformation to apply?

Kernel Trick

- The good news is that we do not need to implement our own transformations
- We have a list of special functions available for that – called **Kernel Functions**
- We simply try multiple of them and select the one that gives us the best results – kind of hyper-parameter tuning

Kernel Trick



Most Commonly Used Kernel Functions

- Linear Kernel
- Polynomial Kernel
- RBF Kernel
- Sigmoid
- Etc

Recommended Reading

1. Section 5.3.2, *A First Course in Machine Learning*, by Simon Rogers and Mark Girolami

Summary

1. Solving a classification problem using a “Separating Hyperplane”
2. Finding the optimum separating hyperplane by maximizing the margin
3. Formulating SVM’s objective function – a constrained optimization problem
4. Solving the objective function using Lagrange Multipliers
5. Issues with SVMs
6. Using SVM to solve non-linear classification problem by means of Kernel functions