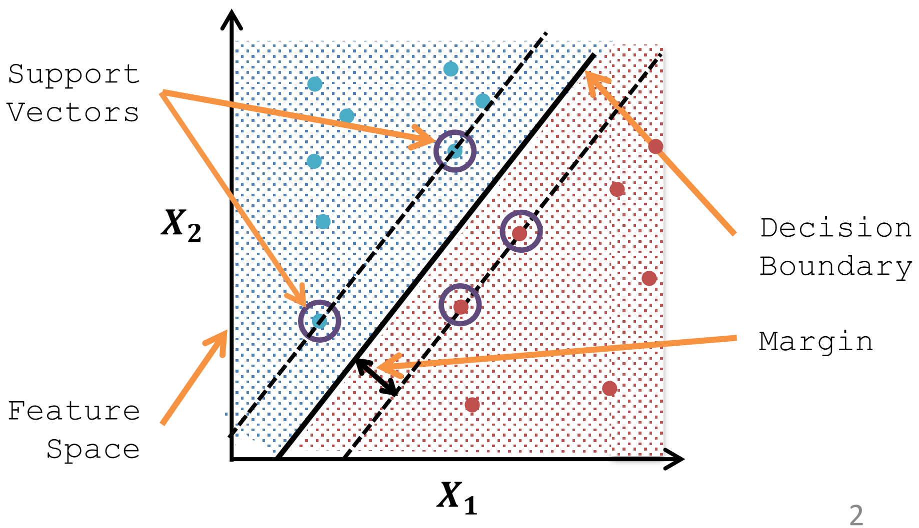
- Module 10 -
Support Vector Machines

Outline

- Maximal Margin Classifier
- Support Vector Classifier
- Support Vector Machines

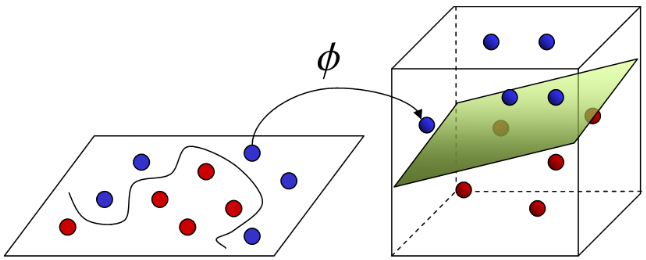
Support Vector Machines (1)

- **Support Vector Machines** (SVMs) have been shown to perform well in a variety of classification settings
- SVM is a generalization of a simple and intuitive classifier called the **maximal margin classifier**
  - maximal margin classifier constructs a linear **decision boundary** separating training samples of two classes by maximizing the perpendicular distance (or **margin**) between the decision boundary and the closest samples (or **support vectors**) from either class
  - the predicted class of a new sample is then determined by the side of the decision boundary it falls on



Support Vector Machines (2)

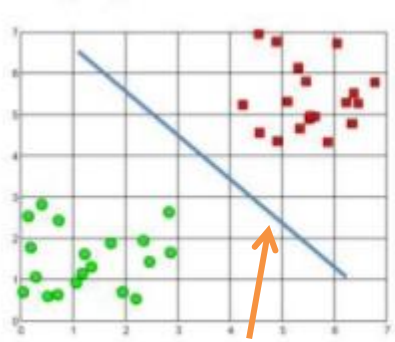
- Properties that make support vector machines attractive:
  - constructs a **maximum margin separator** - decision boundary with the largest possible distance (margin) to the sample points
    - generalizes well on unseen data
  - non-parametric - retain training samples and potentially need to store all training samples (however in practice, the location of the decision boundary depends only on a small fraction of the training samples (support vectors))
  - combines advantages of non-parametric (with flexibility to represent complex functions) and parametric models (resistant to overfitting)
  - able to separate training samples that are not linearly separable in the original input space by using **kernel tricks** to map samples into a higher-dimensional space, where samples are linearly separable



Hyperplane (1)

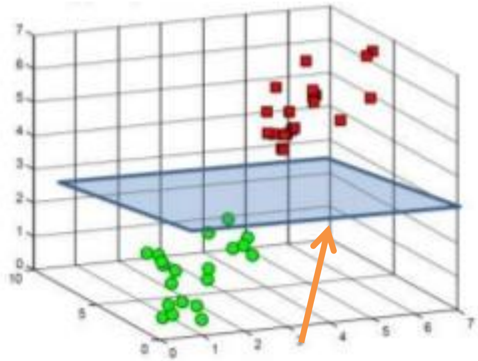
- A **hyperplane** in a p-dimensional space (with p features) is a flat subspace of dimension $p-1$
  - for $p=2$, a hyperplane is a flat 1-D subspace, i.e., a line
  - for $p=3$, a hyperplane is a flat 2-D subspace, i.e., a plane

$p = 2$



$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$

$p = 3$



$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 = 0$

- Recall in a p-dimensional space, a hyperplane is defined by the equation

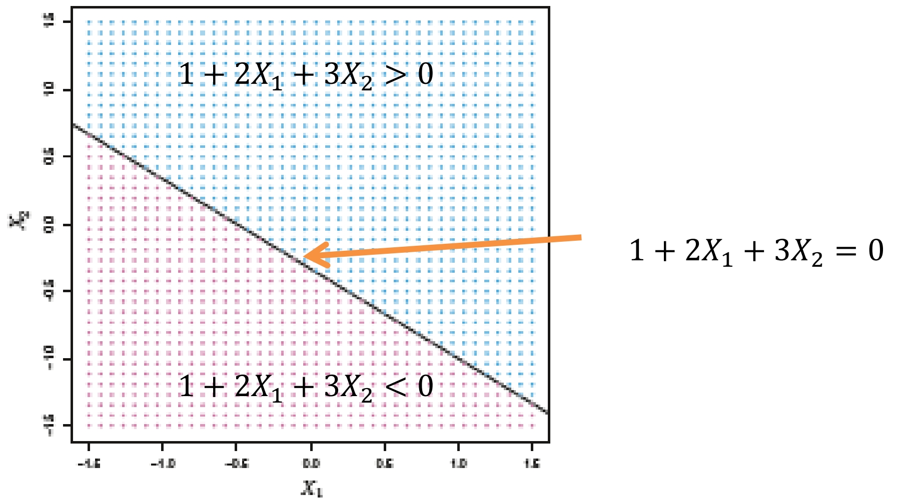
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots + \beta_p X_p = 0$$

Hyperplane (2)

- The hyperplane divides a p-dimensional space into two halves

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0 \text{ and } \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$$

- **Example:** A 2-dimensional space and the hyperplane $1 + 2X_1 + 3X_2 = 0$ shown below

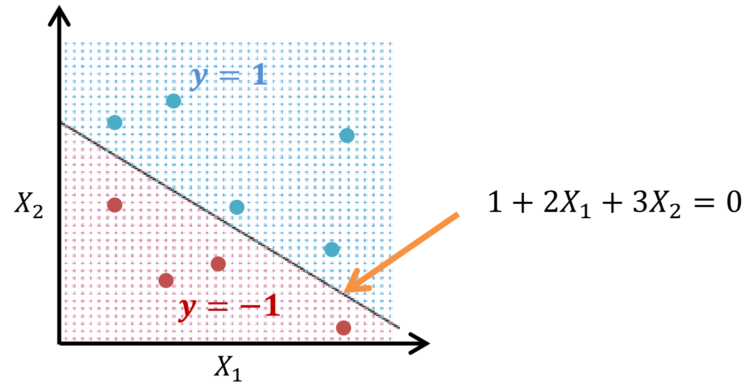


  - **blue region** is the set of points for which $1 + 2X_1 + 3X_2 > 0$
  - **red region** is the set of points for which $1 + 2X_1 + 3X_2 < 0$

Hyperplane: Separating Hyperplane (3)

- Consider a training set consisting of
  - $N = 9$ samples
  - $p = 2$ features (p-dimensional space, $x_i \in \mathbb{R}^p$)
  - class labels, $y_i \in \{-1, 1\}$,
    where -1 represents one class and 1 the other class
    (e.g., -1 represents **red** and 1 represents **blue**)

- Suppose that it is possible to construct a hyperplane that separates the training samples perfectly according to their class labels



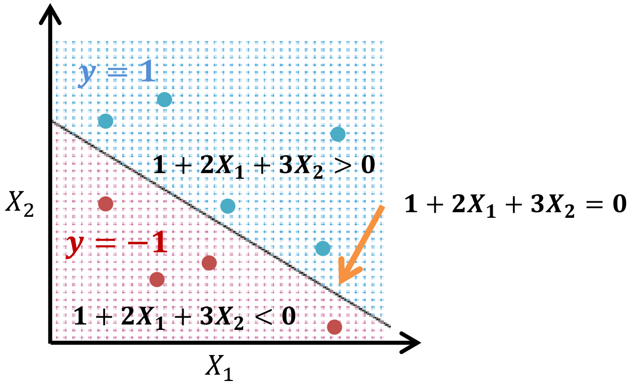
Hyperplane: Separating Hyperplane (4)

- Such a hyperplane is called a **separating hyperplane** and has the property that

$$\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p} > 0 \text{ for } y_i = 1$$

and

$$\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p} < 0 \text{ for } y_i = -1$$



or equivalently, all training samples satisfy the following condition

$$y_i(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p}) > 0, \ \forall\, i = 1, \ldots, N$$

➔ A test sample x is then classified according to the sign of

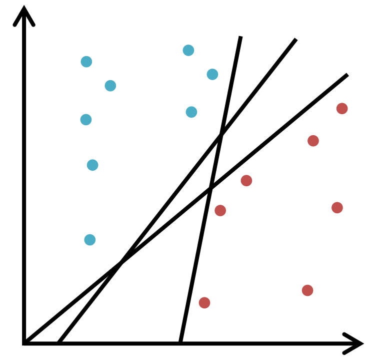
$$f(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

➔ $\hat{y} = 1$ if $f(x) > 0$

➔ $\hat{y} = -1$ if $f(x) < 0$

Maximal Margin Classifier (1)

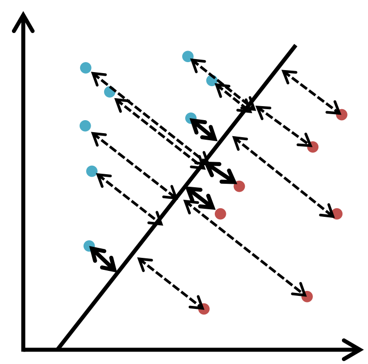
- In general, if the data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes
  ➔ need additional constraints to decide which of the infinite possible separating hyperplanes to use



- **Approach:** Select the separating hyperplane that has the largest distance to the closest training samples from any class

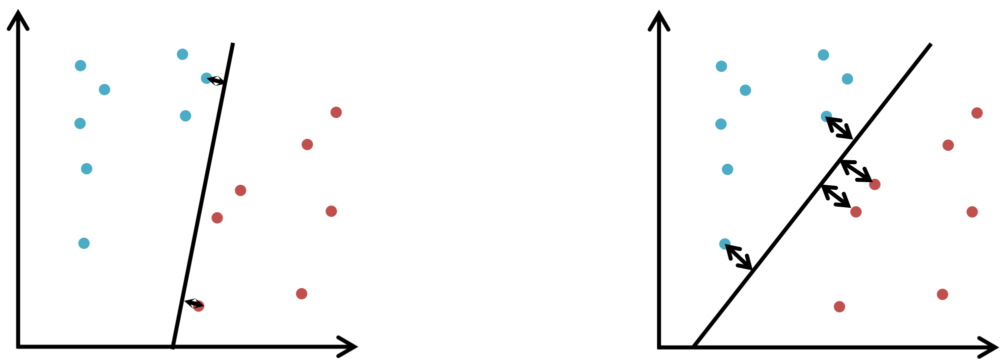
  Step 1: Compute the perpendicular distance from each training sample to a given separating hyperplane

  Step 2: Compute the margin for the given hyperplane as the smallest distance from the training samples to the hyperplane



Maximal Margin Classifier (2)

- The optimal separating hyperplane (also known as **maximal margin hyperplane**) is the separating hyperplane for which the margin is largest



  - the intuition of the maximal margin classifier is that a large margin on the training set will lead to good separation on the test set

    ➔ i.e., lower error on unseen data (lower generalization error)

Maximal Margin Classifier (3)

- The maximal margin hyperplane is constructed by solving the following **optimization problem**

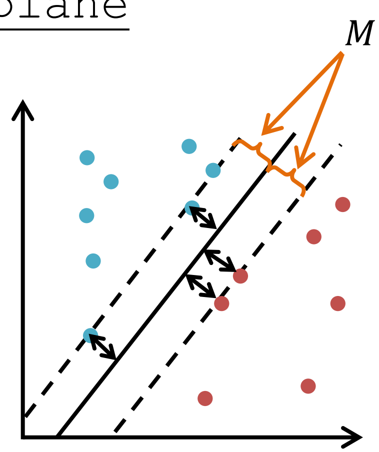
Objective Function

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

Constraints

$$\begin{cases} \text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1\,, \\ y_i\left(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p}\right) \geq M\,, \\ \qquad \forall\, i = 1, \ldots, N \end{cases}$$

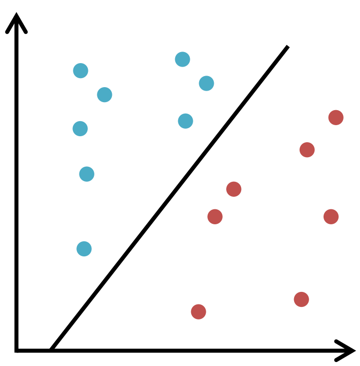
where M represents the margin of the hyperplane

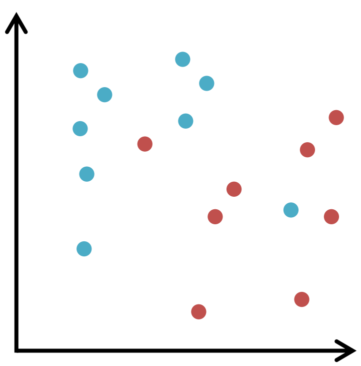


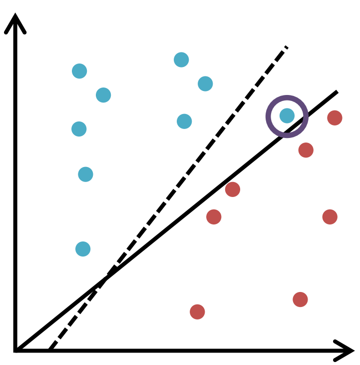
  - the objective of this optimization problem is to choose $\beta_0, \beta_1, \ldots, \beta_p$ so as to maximize the margin, M
  - the second constraint(s) ensure that each training sample is on the correct side of the hyperplane and at least a distance M from the hyperplane

Support Vector Classifier (1)

- However, training samples are not necessarily separable into two classes by a hyperplane
  - even if a separating hyperplane does exist, it may overfit the training data and yield a small margin, M







➔ It is desirable to construct a hyperplane which maximizes the margin while softly penalizing samples that lie on the wrong side of the hyperplane

- for greater robustness to individual samples
- better classification of most of the training samples

Support Vector Classifier (2)

- The intuition is that it is worthwhile misclassifying a few training samples in order to do a better job in classifying the remaining samples
  ➔ instead of minimizing expected empirical loss on the training, attempt to minimize expected generalization loss

- The resulting classifier is called the **support vector classifier** (SVC) (or a soft margin classifier)
  – soft because the classification can be violated by some of the training samples

Support Vector Classifier (3)

- The support vector classifier is constructed by solving the following optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1\,,$$

$$y_i\big(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ \ldots + \ \beta_p x_{i,p}\big) \ \geq M(1-\epsilon_i),\ \forall i = 1,\ldots,N$$

$$\epsilon_i \geq 0,$$

$$\sum_{i=1}^{n} \epsilon_i \leq constant$$

where $\epsilon_i, \ldots, \epsilon_N$ are slack variables that allow individual samples to be on the wrong side of the margin or the hyperplane

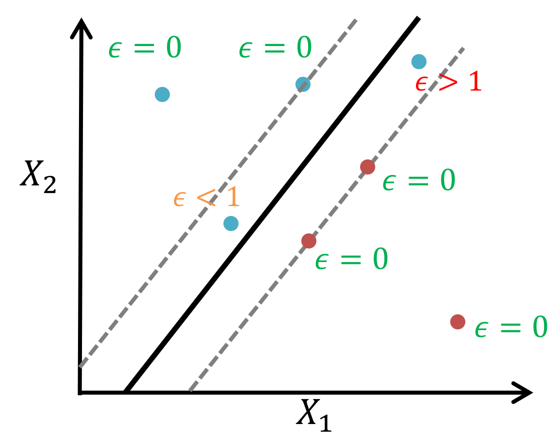
- the objective of this optimization problem is to choose $\beta_0, \beta_1, \ldots, \beta_p$ to maximize the margin, M
- the second constraint allows misclassifications of training samples
- the fourth constraint bounds the total number of misclassifications by a constant

Support Vector Classifier (4)

- Effects of ϵ_i on the second constraint

$$y_i\underbrace{\left(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ \dots + \ \beta_p x_{i,p}\right)}_{f(x_i)} \geq M(1 - \epsilon_i)$$

  - the variables, $\epsilon_i, \dots, \epsilon_N$, one for each training sample, indicates where the i-th sample is located, relative to the margin and the hyperplane
  - If $\epsilon_i = 0$, the i-th sample is on the correct side of the hyperplane $(y_i f(x_i) > 0)$ and on/outside the margin $(y_i f(x_i) \geq M)$
  - If $0 < \epsilon_i < 1$, the i-th sample is on the correct side of the hyperplane $(y_i f(x_i) > 0)$ but inside the margin $(y_i f(x_i) < M)$
  - If $\epsilon_i > 1$, the i-th sample is on the wrong side of the hyperplane $(y_i f(x_i) < 0)$



Support Vector Classifier (5)

- To make the optimization problem easier to solve computationally, it is re-expressed in the following form

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \ \sum_{j=1}^{p} \beta_j^2 + C\sum_{i=1}^{N} \epsilon_i$$

$$\text{subject to} \ \ y_i(\beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ \ldots + \ \beta_p x_{i,p}) \geq 1 - \epsilon_i, \forall i = 1, \ldots, N$$

$$\epsilon_i \geq 0$$

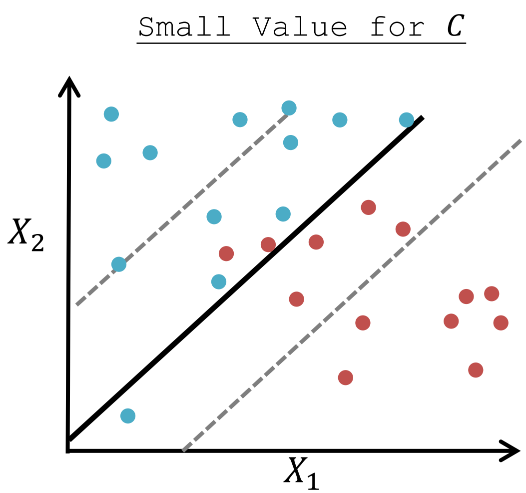
- Notes:
  - it can be shown geometrically that $M = \frac{1}{\sqrt{\sum_{j=1}^{p} \beta_j^2}}$

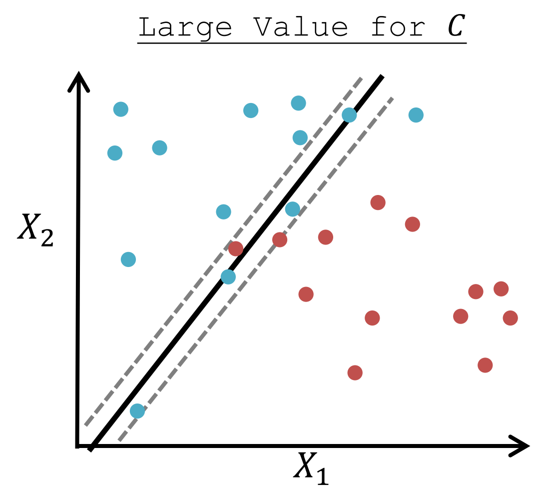
    (maximizing M is equivalent to minimizing $\frac{1}{M} = \sqrt{\sum_{j=1}^{p} \beta_j^2}$ or simply $\sum_{j=1}^{p} \beta_j^2$ as both are increasing functions)

  - The (previous) fourth constraint, $\sum_{i=1}^{n} \epsilon_i \leq constant$, can be replaced by the penalty term, $C\sum_{i=1}^{N} \epsilon_i$, in the objective function, where C is the **cost parameter** that penalizes misclassifications of training samples

Support Vector Classifier (6)

- The cost parameter, C, controls the width of the margin and the bias-variance tradeoff





  - as C is increased, the optimization problem will further minimize $\sum_{i=1}^{N} \epsilon_i$, i.e., fitting to the training samples better, and obtaining a classifier that potentially has lower bias but higher variance

    ➔ in the limit $C = \infty$, the resulting classifier is the maximal margin classifier where no training sample is allowed lie within the margin or on the wrong side of the hyperplane

➔ the best value of C is chosen using cross-validation, i.e., select C that gives the lowest cross-validation estimate of prediction error