Support Vector Machines

One of the most commonly used of machine learning algorithms is support vector machine. It was developed in the 1990s by Vladimir Vapnik and colleagues and that has become more popular since then. The support vector machine is a supervised learning approach for classification, which is generally suited for binary classification. There are further extensions of support vector machines to accommodate cases of more than two classes, but it is outside of the scope of our seminar project.

Maximal Margin Classifier

It all started with a simple classifier called the maximal margin classifier. We will firstly define the concept of an hyperplane.

Hyperplane
In a $n-$dimensional space, a flat affine subspace of dimension $n-1$ is called a hyperplane, e.g. a line is a flat one-dimensional subspace of two-dimensional space for that a line is a hyperplane of two-dimensional space. Another example is, in three dimensional space, a plane is a flat two-dimensional subspace that is also called hyperplane.

The definition of a $n-$dimensional hyperplane is defined by the equation:

$$h(X) := \beta_0 + \beta^T X = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n = 0,$$
$$\beta_0 \in \mathbb{R},\ \beta \in \mathbb{R}^n.$$

If a point $X = (X_1, X_2, \ldots, X_p)^T$ satisfies the hyperplane equation, then X lies on the hyperplane.

Suppose that X has the following property:

$$h(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n < 0$$

or

$$h(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n > 0.$$

Since we have inequality instead of equality, it follows that X lies on one side of the hyperplane.

Classification Using a Hyperplane
Given a data set of m training observations in $n-$dimensional space:

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}, \ldots, x_m = \begin{pmatrix} x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix},$$

where the dimension of n corresponds to size of features and we define two classes where these observations fall into: $\{-1, 1\} \ni y_1, \ldots, y_m$. One can say that each observation x_i has a property of y_i that is either -1 or 1.

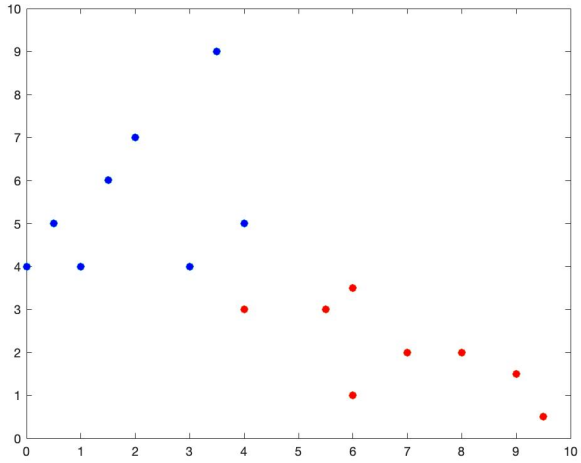

Figure 1: Two classes of observations, shown in blue and in red

We will now construct some hyperplanes that separate the training observations perfectly. Three separating hyperplanes are shown by figure 3.

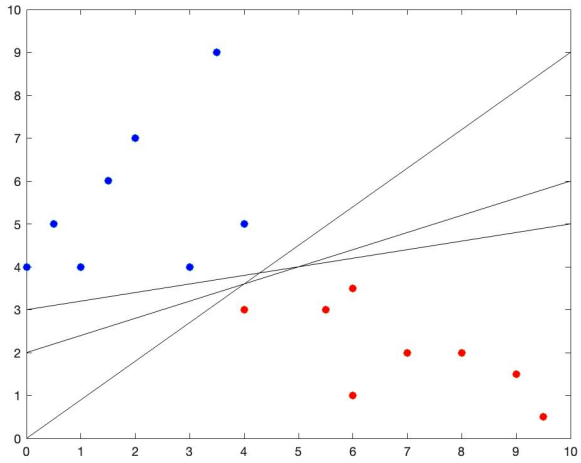

Figure 2: Separating hyperplanes

Suppose the observations from the blue class have the property of $y_i = 1$ and those from the red class $y_i = -1$. Then all separating hyperplanes have the property:

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} < 0 \text{ if } y_i = -1$$

and

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in} > 0 \text{ if } y_i = 1.$$

It is equivalent to:

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in}) > 0$$

for all $i = 1, \ldots, n$.

Maximal Margin Classifier

If the data sets can be perfectly separated, then there will be an infinite number of such hyperplanes. Imagine we tilt the separating hyperplane by a little bit, there exists many hyperplanes that do not touch the observations. Therefore we have to decide which hyperplane to use. One of the reasonable way is using the maximal margin classifier as method to decide which is the optimal separating hyperplane. Optimal separating hyperplane is a hyperplane that is farthest from the training observations. The smallest distance between the observation to the separating hyperplane is known as the margin and is given by:

$$M = \frac{1}{\|\beta\|_2} + \frac{1}{\|\beta\|_2} = \frac{2}{\|\beta\|_2}.$$

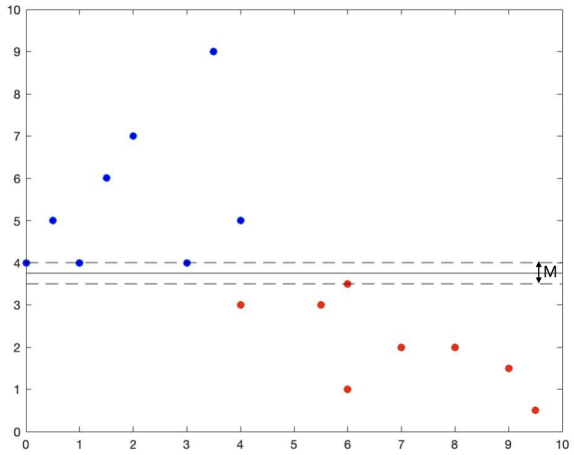

Figure 3: Separating hyperplanes and its margin

Distance between any observation $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T,\ i = 1, \ldots, m$ and the hyperplane $\beta_0 + \beta^T X = 0$ is defined by:

$$d(x_i) := \frac{|\beta_0 + \beta^T X|}{\|\beta\|_2}.$$

Figure 4 shows a separating hyperplan and its margin. However, this hyperplane is not optimal, since we can find another separating hyperplane with greater margin. It could be said that optimal separating hyperplane is the separating hyperplane, which has largest margin.

This can be considered as an optimization problem. The optimal separating hyperplane is the solution of the following optimization problem:

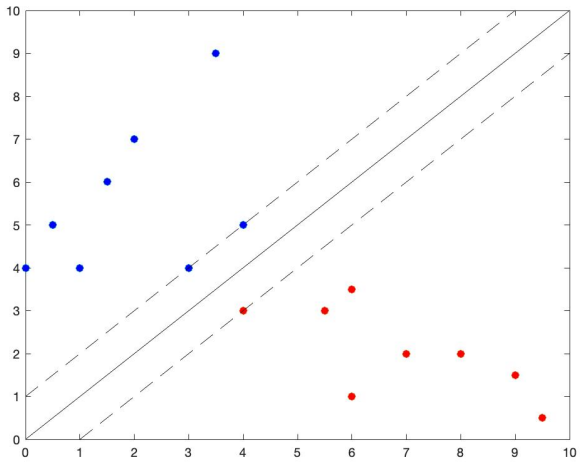

Figure 4: Optimal separating hyperplanes

$$\min_{\beta_0,\beta} \frac{\|\beta\|^2}{2}$$
$$\text{s.t. } y_i(\beta_0 + \beta^T X_i) \geq 1, \ \forall i \in \{1, \ldots, m\}.$$

The maximal margin classifier can be applied when training observations are linearly separable, or it can be said, if a separating hyperplane exists. Unfortunately most data sets are linearly non-separable, that means the maximal margin classifier is not applicable, since the classes should be separable by a linear boundary.

Soft Margin Classifier

Soft Margin Classifier is an extension of maximal margin classifier. The difference between soft margin classifier and maximal margin classifier is, that in soft margin classifier some observations are allowed to be on the incorrect side of the margin or even on the wrong side of hyperplane. We can say, that the margin can be violated by a small subset of observations.

In the optimization problem we introduce slack variables ϵ_i, which measure and penalise the degree of misclasification of x_i. Then the hyperplane is the solution to the optimization problem:

$$\min_{\beta_0,\beta,\epsilon_i} \frac{\|\beta\|^2}{2} + C\sum_{i=1}^{m} \epsilon_i$$
$$\text{s.t. } y_i(\beta_0 + \beta^T X_i) \geq 1 - \epsilon_i \ , \ \epsilon_i \geq 0 \ , \quad \forall i \in \{1, \ldots, m\},$$
$$\sum_{i=1}^{m} \epsilon_i \leq C,$$

where C is a tuning parameter, that is nonnegative. The slack variable ϵ_i describes the position of a given point x_i. If $\epsilon_i = 0$ then there is no misclassification and it follows that x_i is on the correct side. If $\epsilon_i \in (0, 1)$ then x_i lies between the support hyperplane and the separating hyperplane. If $\epsilon_i > 1$ then x_i is on the wrong side of the separating hyperplane. The tuning parameter C has a role as upper bound of the sum of ϵ_i's, so it determines how much we will tolerate.

As the parameter C increases, the margin will widen and it implies that more violation are allowed to the margin. When C is small, the margins are narrow and rarely violated. We generally choose the tuning parameter C via cross-validation. The maximal margin classifier and the soft margin classifier deal only with linear separable data sets. However as we have hinted, most problems are linearly non-separable. That means, there exists no hyperplane that can separate the data sets. Therefore we apply the support vector machines.

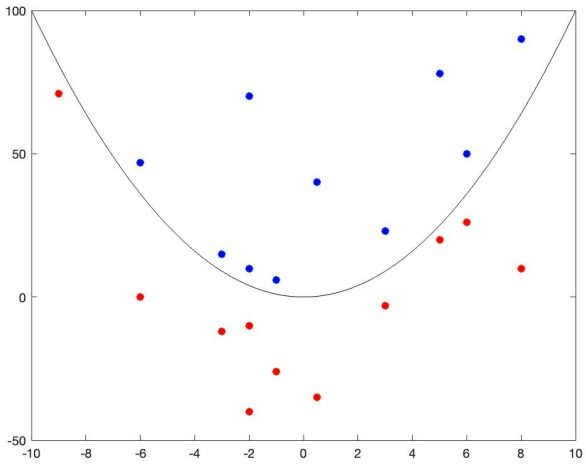

Figure 5: Non-linear problem

Support Vector Machines

Support vector machine is a generalisation of soft margin classifier. Since the problem is not linearly separable in input space, we enlarge the space using kernels. First of all we define the generalized definition of hyperplane. Let V be a vector space with inner product $\Phi : \mathbb{R}^n \to V$, then the hyperplane can be written as:

$$h(X) = \beta_0 + \beta^T \phi(X).$$

The function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,

$$K(X_i, X_j) = \phi(X_i)^T \phi(X_j),$$

is referred to as kernel function.

Then the modified optimization is defined by:

$$\min_{\beta_0,\beta,\epsilon_i} \frac{\|\beta\|^2}{2} + C\sum_{i=1}^{m}\epsilon_i$$

$$\text{s.t. } y_i(\beta_0 + \beta^T\Phi(X_i)) \geq 1 - \epsilon_i \ , \ \epsilon_i \geq 0 \ , \quad \forall i \in \{1,\ldots,m\},$$

$$\sum_{i=1}^{m}\epsilon_i \leq C.$$

There are some popular kernel functions. For instance, we simply take:

$$K(X_i, X_j) = X_i^T X_j,$$

that is known as a linear kernel. The linear kernel leads us back to the soft margin classifier, because the soft margin classifier is also linear in the input space. Some other kernel functions are:

1. Polynomial kernel of degree d

$$K(X_i, X_j) = (1 + X_i^T X_j)^d,$$

2. Sigmoid kernel

$$K(X_i, X_j) = tanh(\alpha X_i^T X_j + \gamma),$$

3. Gaussian radial basis kernel

$$K(X_i, X_j) = exp(-\gamma\|X_i - X_j\|^2).$$