Lecture 101

Date: 30 March 2020
Scribes: Akash Gupta
Topics: SVM part 1

0.1 Understanding Hyperplanes

A Hyperplane is simply an **Affine** subspace of a vector space. So a p dimensional vector space, a hyperplane is a flat affine subspace to dimension $p-1$. Therefore for a $2-D$ space, the hyperplane is a line and for a $3-D$ space the hyperplane would be a $2-D$ plane. Note that an affine subspace is a subspace that does not necessarily pass through the origin. So for a $2-D$ and $p-D$ space hyperplanes can be represented as :-

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \tag{1}$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{2}$$

Note that in equation 1 the hyperplane follows the equation of a line and any point $X = (X_1, X_2)^T$ that lies on the hyperplane satisfies its equation as well. Taking some other cases, there can also be situations where that X point satisfies not an equation but rather an inequality as follows :-

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \tag{3}$$

In this situation, any X vector that satisfies the above equation lies to one side of the hyperplane. In short, we can say that the hyperplane divides the $p-D$ space in two sections.

0.1.1 Classification : separating hyperplane

Consider an $n \times p$ data matrix X with p attributes, making it a p dimensional space where each attribute has n training observations. The n vectors of observations for each attribute can be written as :-

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1p} \end{bmatrix}, x_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{np} \end{bmatrix} \tag{4}$$

These n observation vectors correspond to n class response values : $y_1, y_2, \cdots, y_n$ and all these class values essentially take 2 possible values since this is a binary classification setting : $(-1, 1)$. Also suppose we have a test observation vector of p attributes given by :-

$$x^* = (x_1^*, x_2^*, \cdots, x_p^*)^T \tag{5}$$

We want to essentially develop a classifier model based on the training data and then be able to classify the test observation with a certain degree of accuracy. For this

we will use the concept of the **Separating hyperplane**. Idea behind this is simple : Suppose there exists such a hyperplane in the $p-D$ space in which n observation vectors of p dimensions belong, such that the hyperplane can perfectly separate the observation vectors belonging to the two classes. We want a hyperplane that divides the space such that all observation vectors on either side of the hyperplane belong perfectly to the two classes. A pictorial depiction is shown :- Suppose the

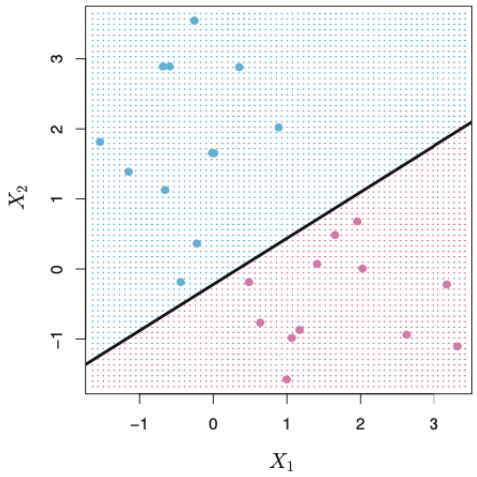

Figure 1: separating hyperplane

observations from the blue class are $y_i = 1$ and from purple class are $y_i = -1$. We then notice that the separating hyperplane has the following properties :-

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0, \ if \ y_i = 1 \tag{6}$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0, \ if \ y_i = -1 \tag{7}$$

A general property of a hyperplane is :-

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0 \tag{8}$$

So basically a test observation vector is assigned a class depending on which side of the hyperplane it belongs to. Finally we classify our test observation based on its sign.

$$f(x*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^* \tag{9}$$

So in short if $f(x*) < 0$ it gets the class $y^* = -1$ and the other class otherwise. Magnitude also plays a role since higher the magnitude of the signed value, the more farther away it is from the hyperplane boundary and more confident we are about its class. Basically the blue and purple grids in the figure indicate the decision rule that the separating hyperplane uses to classify observations.

0.1.2 Maximal Margin Classifier

Since it is usually possible to construct infinite number of hyperplanes to separate out the classes, we need a way to select the most optimal hyperplanes out of all

possible choices. Hence we use the **maximal margin hyperplane** which is a hyperplane that is farthest from the training observations. We calculate a quantity known as the **margin** which is the perpendicular distance of a particular hyperplane from the training observations (which also happens to be the smallest distance of an observation from the hyperplane). Note that the maximal margin hyperplane is one for which the margin is the largest. A test observation is then classified as per which side of the maximal margin hyperplane it belongs to and this whole setup makes up the **maximal margin classifier**. This can lead to overfitting when p is large. The coefficients of the maximal margin hyperplane are given by :-

$$\beta_0, \beta_1, \cdots, \beta_p \tag{10}$$

The classifier specifies the class of an observation vector as per the sign of the equation :-

$$f(x*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^* \tag{11}$$

The maximal margin hyperplane is infact associated with the largest minimum distance between hyperplane and obersation - largest margin. It can be thought of as the **mid-line between the widest slab that can be inserted between two classes**. Pictorial representation below :- Note in particular the three points on

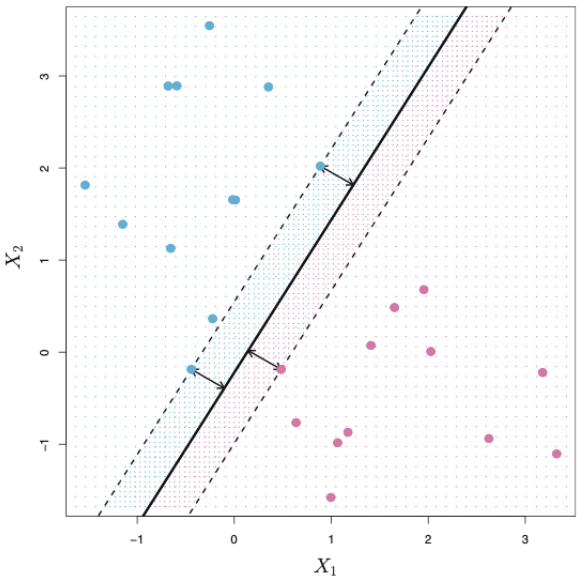

Figure 2: max margin hyperplane

either side of the hyperplane closest to it. These points essentially from the 'slab' or the margin width. They lie along the indicated dashed lines. An important point is that these points/vectors/obervations are the **support vectors** that support the maximal margin hyperplane in that if they are moved around, then even the max margin plane would move. So we can say that the maximal margin hyperplane depends only on a small subset of observations (support vectors) and not on all observations.

0.1.3 Constructing the max margin classifier

Before constructing, we will consider n observation vectors of p dimension : $x_1, x_2, \cdots, x_n \in R^p$ and their corresponding binary class response values : $y_1, y_2, \cdots, y_n \in (-1, 1)$. Constructing this classifier involves solving an optimization problem :-

$$\max_{\beta_0, \beta_1, \cdots, \beta_p} M \tag{12}$$

$$Subject\ to \rightarrow \sum_{j=1}^{p} \beta_j^2 \tag{13}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > M\ \forall i = 1, \cdots, n \tag{14}$$

The constraint given by 14 simple states that each observation will be on the correct side of the hyperplane provided some positive value of M. This constraint is actually the same thing as that given by equation 8 which essentially ensures each observation is on the right side of the hyperplane, with some cushion of the margin - some positive value M. The two constraints make sure that not only are the observations on the correct side of the hyperplane but also at least a distance of M away from it. The perpendicular distance of any observation with the hyperplane is given by :-

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \tag{15}$$

To sum it up, we are choosing β_i values such that margin M is maximized to construct the maximal margin hyperplane. Note that incase we cannot exactly separate out the observations, maximal margin classifier cannot be used since the optimization with constraint $M \geq 0$ will have no solution. In case we extend this concept to include a **soft margin** and approximate a max margin classifier using a concept called **support vector classifier**. Used usually for non separable situations.