S21CSE471: Statistical Methods in AI **Lecture Date: 1st Feb, 2021**

Support Vector Machine

Vishal Patel (2020201082), Pullamma Mayakuntla (2018101119), Snehangshu Bhattacharjee (2020701002)

1 Introduction

The Support Vector Machine, created by Vladimir Vapnik in the 60s, but pretty much overlooked until the 90s is still one of most popular machine learning classifiers.

The objective of the Support Vector Machine is to find the best splitting boundary between data. In two dimensional space, you can think of this like the best fit line that divides your dataset. With a Support Vector Machine, we're dealing in vector space, thus the separating line is actually a separating hyperplane.

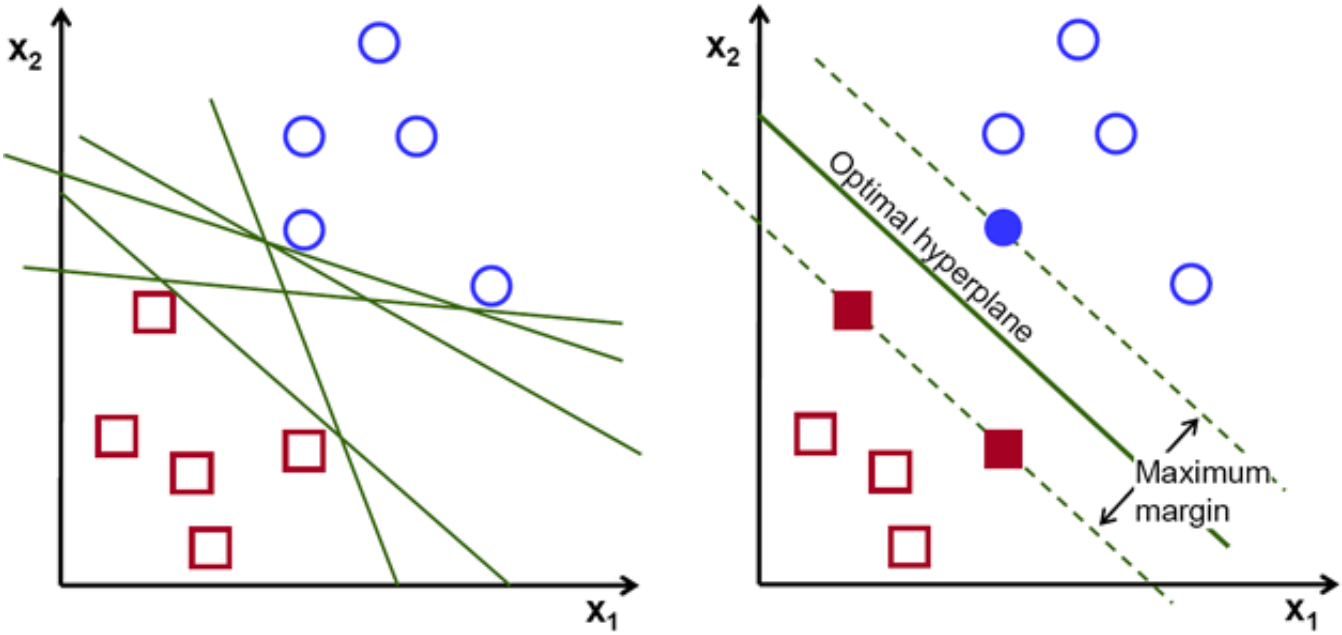

Figure 1: Possible hyperplanes

2 What is hyperplane?

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

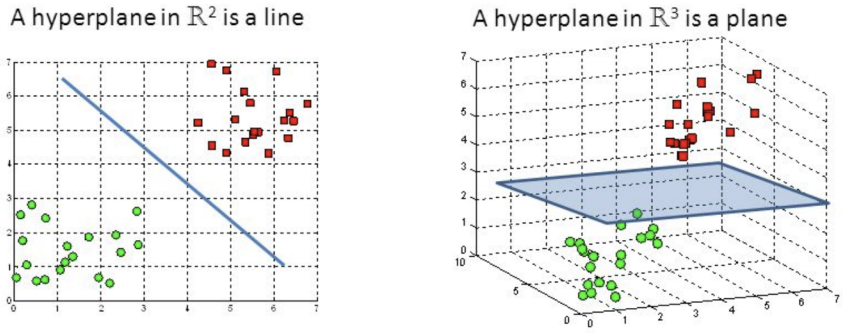

Figure 2: Above figures corresponding to hyperplanes in 2D and 3D respectively

3 What is Support Vectors?

The vector points that the margin lines touch are called as Support Vectors.in the given below figure,encircled points are support vectors

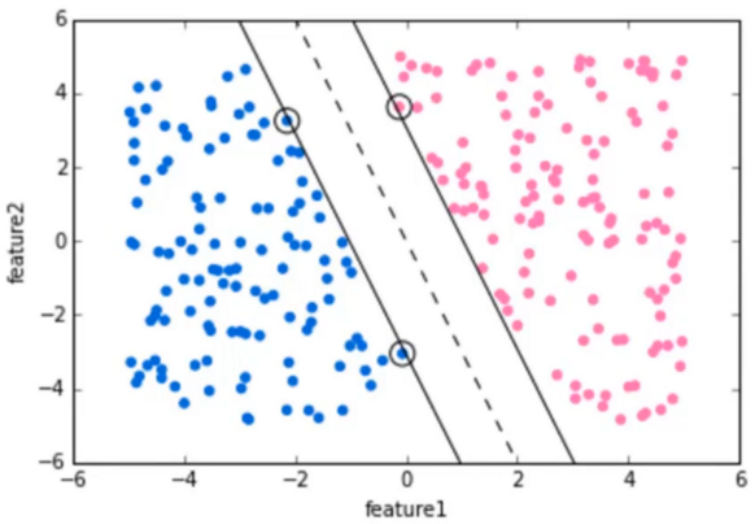

Figure 3: Support Vector illustration example

Support Vectors influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane.

4 What if Data points are not linearly seperable?

Now consider what if we had data as shown in image below?It's pretty clear that there's not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them.

So here's what we'll do: we will add a third dimension. Up until now we had two dimensions: x and y. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$ (you'll notice that's the equation for a circle).

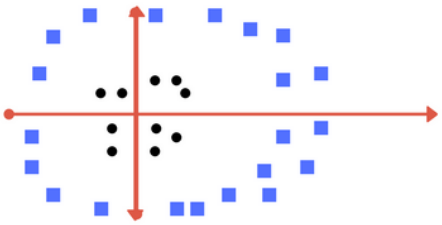

Figure 4: A more complex dataset

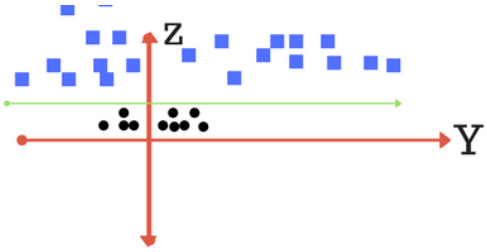

Figure 5: plot of zy axis. A separation can be made here.

Solution : kernel trick

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

Note that since we are in three dimensions now, the hyperplane is a plane parallel to the x axis at a certain z.

When we transform back this line to original plane, it maps to circular boundary as shown in image E. These transformations are called kernels.

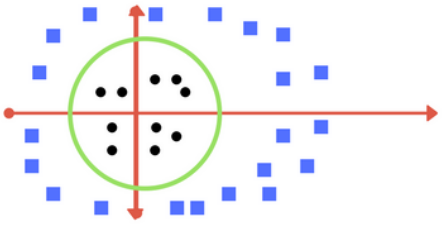

Figure 6: Back to our original view, everything is now neatly separated

5 Multiple Solutions exist for linearly seperable data

In the figure 7, for the linearly seperable data,we see that there are three solutions.Now,we have to decide if all the solutions are equally good and if not, which is good and why?

In the figure 7, the black line is preferred as good solution because it has more margin(Here margin is the width of the band around decision boundary without any training samples basically can be assumed

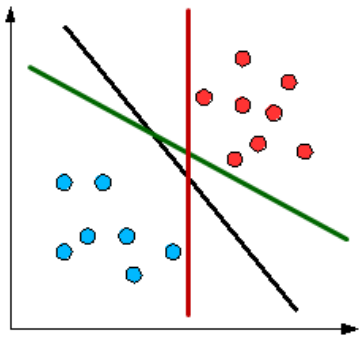

Figure 7: Back to our original view, everything is now neatly separated

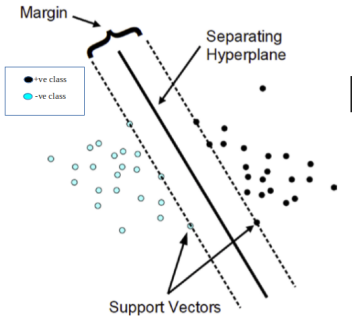

Figure 8: Back to our original view, everything is now neatly separated

as the width of the road.) from both sides.Why the solution with more margin preferred because,if the margin is more it will perform well in future on test data even its performance is low now on training data.It performs well when there is noise.For example if we consider the red line as a good solution for a while,if a new test point comes and takes place just beside the point that is now immediately succeeding the red line,then we classify the category of the new point to be the blue category.But if the solution line has more margin from the data even if many test data points take place beside already existing train data we will not get wrong category as in the case of red line.

Bubbles around samples

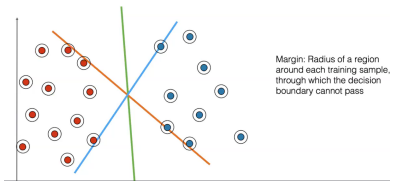

Figure 9: A really Awesome Image

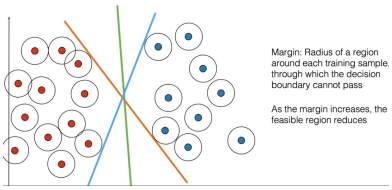

Figure 10: A really Awesome Image

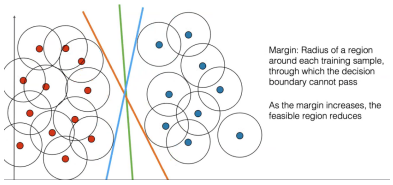

Figure 11: A really Awesome Image

If we introduce bubbles (Bubbles represent that the class of pints around a point within its bubble boundary is similar to the point for which the boundary is drawn.) around points, we have to align the solution lines and as we increase the bubble size we eventually get to the middle line with more margin as the final solution.

The margin as shown in figure 8 will not be determined by all the training data it is decided by a few points which will not allow the margin to go beyond them in both sides.And those points are called the support points or vectors which decide the line.

Even there is imbalance in the data,the algorithm which we will discuss further to determine the line

that seperates the classes is good because it only depends on support vectors.

Samples that supports the boundary are called support vectors.The algorithm for the line in classification is called support vector machine.

6 Training and Test Performance

Vapnik and Vladmir has proposed a function which bounds test performace based on the performance on train data. The function is as given in the figure 12.In the figure 12, Here

$$R(\alpha) - Performanceontestdata$$

,

$$R_train(\alpha) - Performanceontraindata$$

,

$$\sqrt{\frac{f(h)}{N}}$$

-Is a monotonically increasing function of h. Since f(h) is monotonically increasing function of h if we decrease h f(h) decreases.So If we find a solution with small h we will have good bound on the test performance. **How to find h.** h is defined as a relative margin = $\rho_{\overline{D}}$

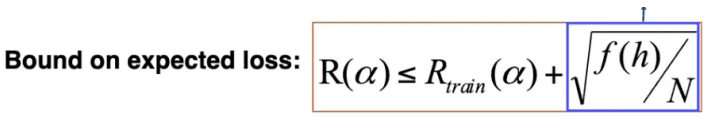

Figure 12: Back to our original view, everything is now neatly separated

$$f(h) = h + h\log(2N) - h\log(h) - c$$

Figure 13: Back to our original view, everything is now neatly separated

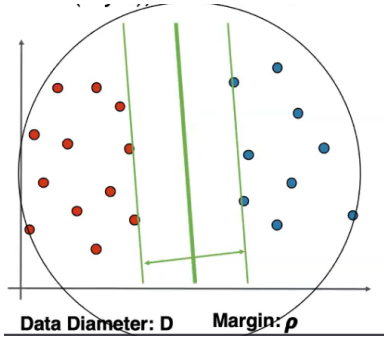

Figure 14: Back to our original view, everything is now neatly separated

7 Concept of Relative Margin

If we assume a circle covering all the data points in our 2D space (we are considering two dimensions here for the sake of simplicity), then let us denote the diameter of that circle by D and the margin by ρ. Please refer to Figure 15, for the pictorial representation of the same.

If we scale every data point by a positive factor of magnitude more than 1, the circle around the data

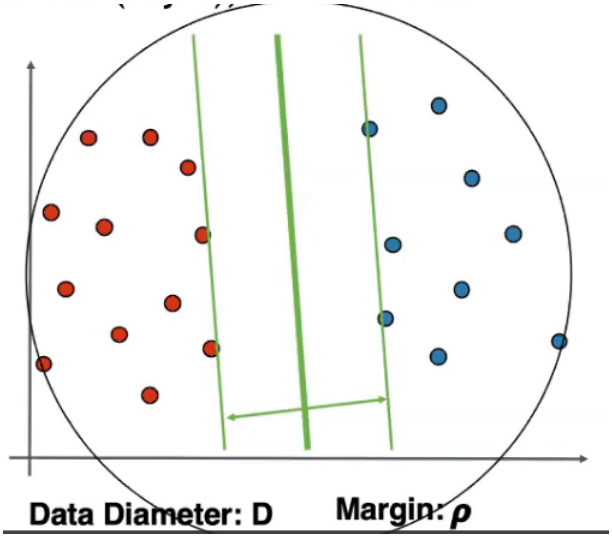

Figure 15: Pictorial representation of Data Diameter and Margin

points expands and thus expanding the margin as well, however the relative margin given by $\frac{\rho}{D}$ remains constant whereas the absolute margin increases. Also, scaling the data points by a positive factor of magnitude less than 1, shrinks the data set which further reduces the margin, however the relative margin remains the same whereas the absolute margin decreases. So we are interested in maximising the relative margin more than the absolute margin.

8 Formalizing the margin

Since the data set consists of two classes, we formulate the problem by considering a positive label for one class and a negative label for another class, positive classes represented by $y_i = +1$ and negative classes by y_i = -1

A linear classifier (in 2D, a line) can formally be written as W^TX + b, where W^T is the transpose of the learnable parameters and X represents the data points and b is the constant term.

For a linear classifier, points lying on it gives W^TX + b = 0, however points lying on either side of the lines gives W^TX + b $<$ 0 or W^TX + b $>$ 0 accordingly.

Considering a positive value k to be the perpendicular distance between the linear classifier and the support vectors we reformulate our problem as,

For positive class points ($y_i = +1$):

$$W^T X + b - k \geq 0 \tag{1}$$

For negative class points (y_i = -1):

$$W^T X + b + k \leq 0 \tag{2}$$

Using equation (1) and (2) we conclude:

$$y_i(W^T X + b) \geq k, \forall i \tag{3}$$

Figure 16, is the pictorial representation of our formulation, where $g(x) = W^T X + b$

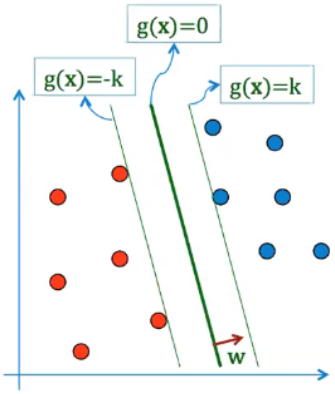

Figure 16: Linear Classifier at a distance of k from the support vectors

9 Formulating the problem as an Optimization Problem

Our objective is to find a line that separates the two classes linearly and also maximizes the margin. While formulating this as an optimization problem, we assume k = 1, such that the closest points from each class (X_p for the positive class and X_n for the negative class) to the linear classifier satisfies, $W^T X_p$ + b = +1 and $W^T X_n$ + b = -1 respectively. Subtracting the second equation from the first we get:

$$W^T(X_p - X_n) = 2 \tag{4}$$

dividing equation (4) by $\|W\|$,

$$\frac{W^T}{\|W\|}(X_p - X_n) = \frac{2}{\|W\|} \tag{5}$$

In equation (5), the LHS represents the margin which has to be maximised. So by maximising the RHS we would also solve the same problem and further to maximise the RHS we minimize the denominator $\|W\|$.

Since $\|W\|$ is WW^T, we finally formulate the optimization problem as:

$$\begin{aligned} \min \quad & \frac{1}{2} W^T W \\ \text{subject to:} \quad & y_i(W^T X_i + b) \geq 1, \forall i \end{aligned} \tag{6}$$

References

[1] https://cse.iitkgp.ac.in/ dsamanta/courses/da/resources/slides/10SupportVectorMachine.pdf

[2] https://www.youtube.com/watch?v=FG0TcQrWN5kt=847s

[3] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/