A PROJECT REPORT ON

LINEAR ALGEBRA IN MACHINE LEARNING – SUPPORT VECTOR MACHINE CLASSIFICATION

SUBMITTED BY

MS. AMISHA CHAKRABORTY (03)

MS. AMULYA JOSHI (08)

MS. ANUKRITI TRIPATHI (28)

UNDER THE GUIDANCE OF

MS. POONAM MORE

USHA MITTAL INSTITUTE OF TECHNOLOGY

SNDT WOMEN'S UNIVERSITY

JUHU TARA ROAD, SANTACRUZ WEST, MUMBAI-400049

2019-2020

INDEX

1. AIM:

To learn applications of support vector machine classification in linear algebra.

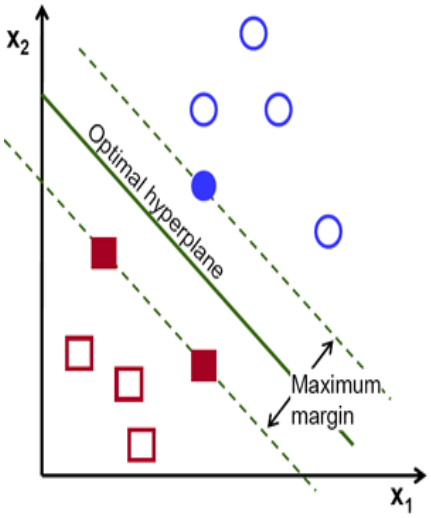
2. INTRODUCTION:

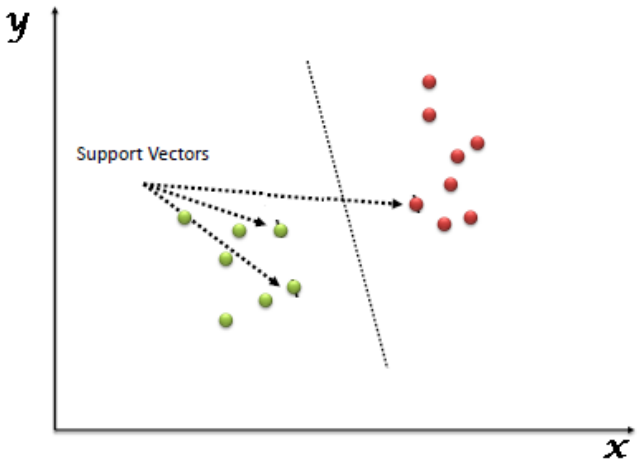
Support vector machine is another simple algorithm that every machine learning expert should have in his/her arsenal. Support vector machine is highly preferred by many as it produces significant accuracy with less computation power. Support Vector Machine, abbreviated as SVM can be used for both regression and classification tasks. But it is widely used in classification objectives.

It is an application of the concept of **Vector Spaces in Linear Algebra.**

Support Vector Machine, or SVM, is a discriminative classifier that works by finding a decision surface. It is a supervised machine learning algorithm.

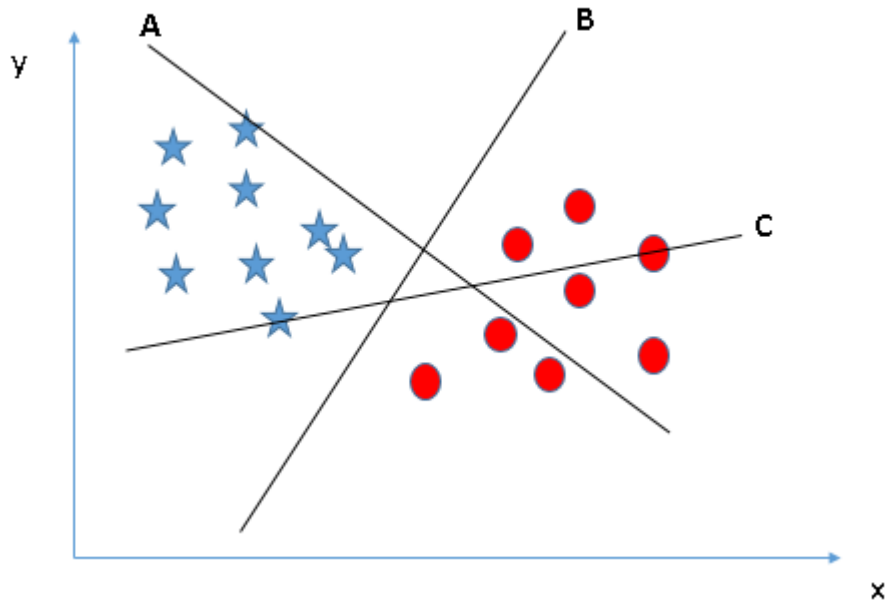
The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.





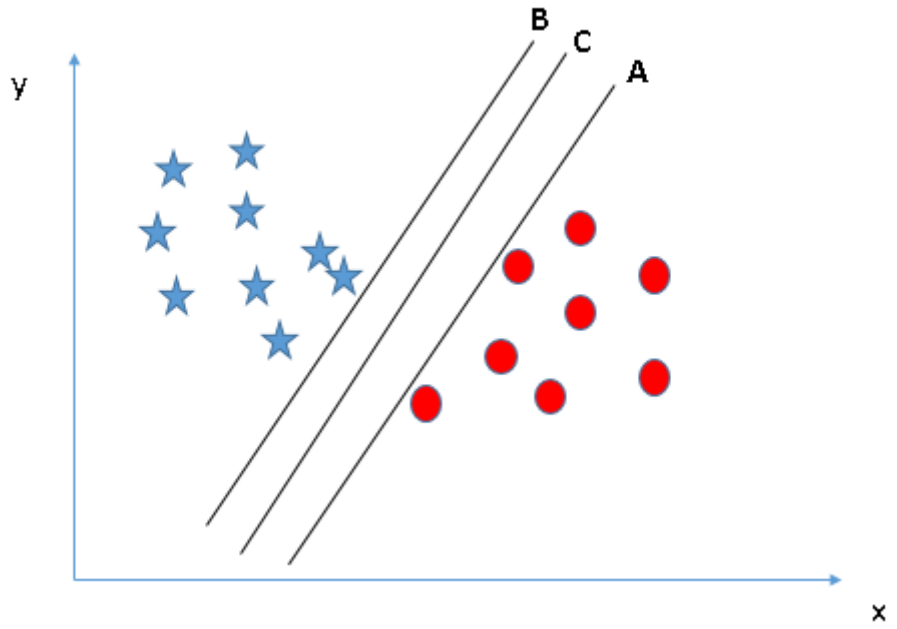
3. WORKING:

- Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

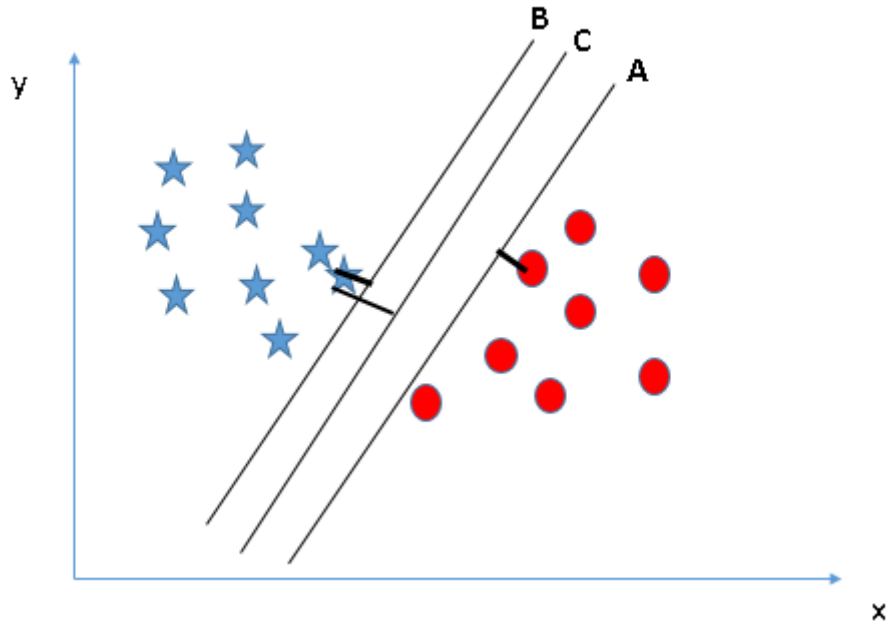


You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, how can we identify the right hyper-plane?



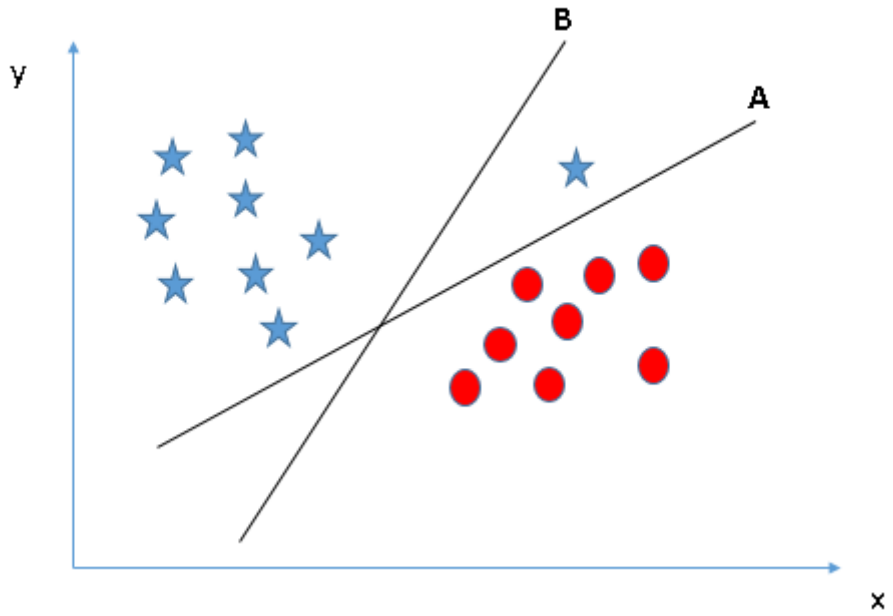
Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin. Let's look at the below snapshot:



Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C.

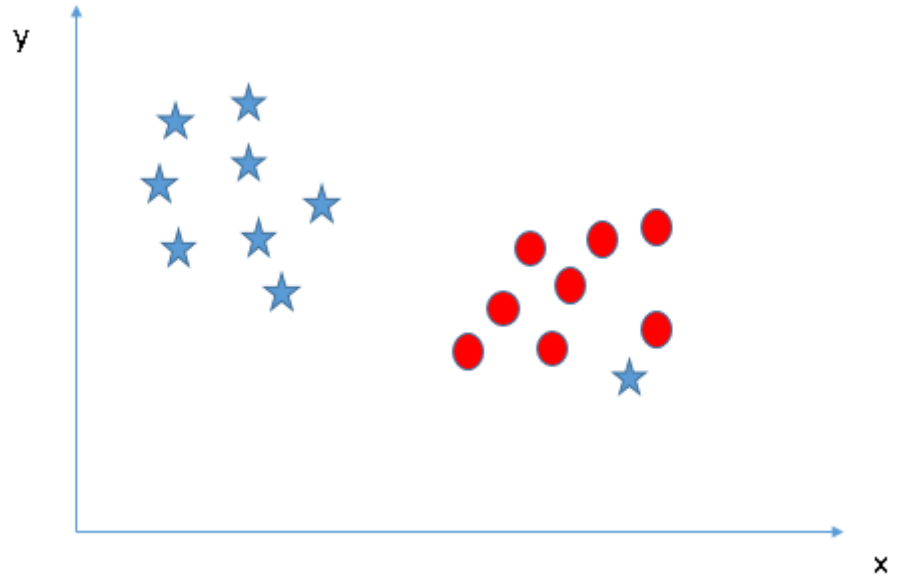
Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- Identify the right hyper-plane (Scenario-3): Hint: Use the rules as discussed in previous section to identify the right hyper-plane.

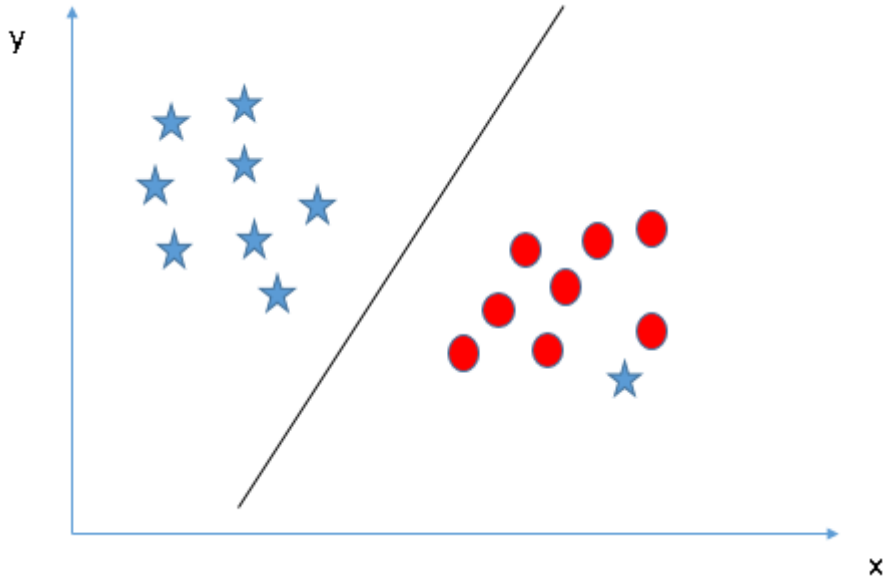


Some of you may have selected the hyper-plane B as it has higher margin compared to A. But here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.

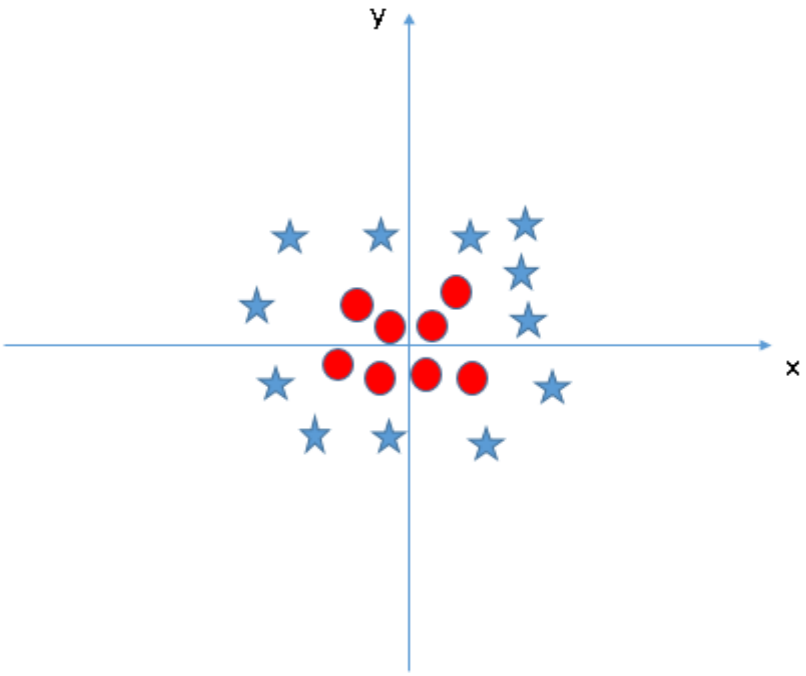
- Can we classify two classes (Scenario-4)? : Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



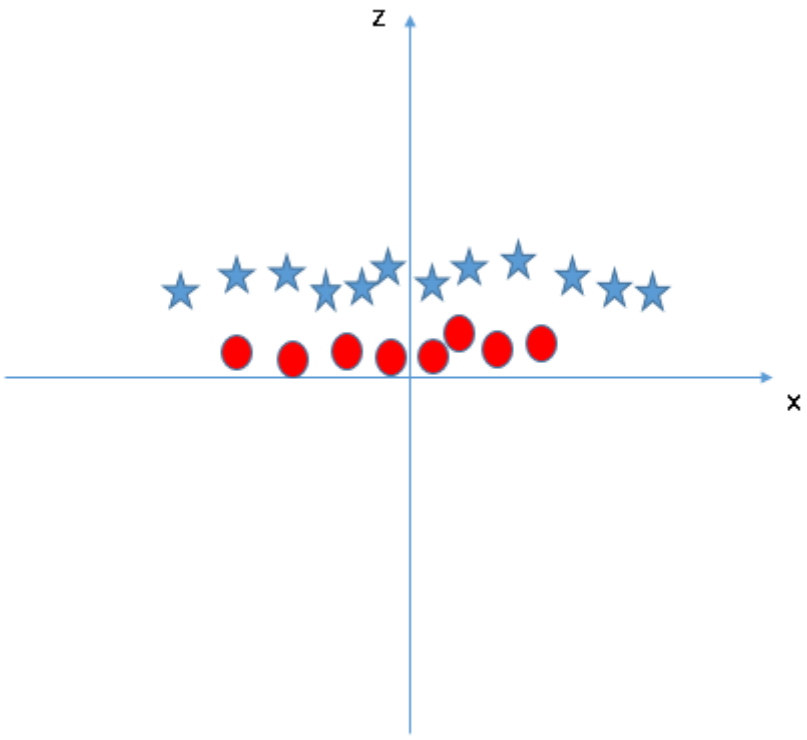
As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let’s plot the data points on axis x and z:

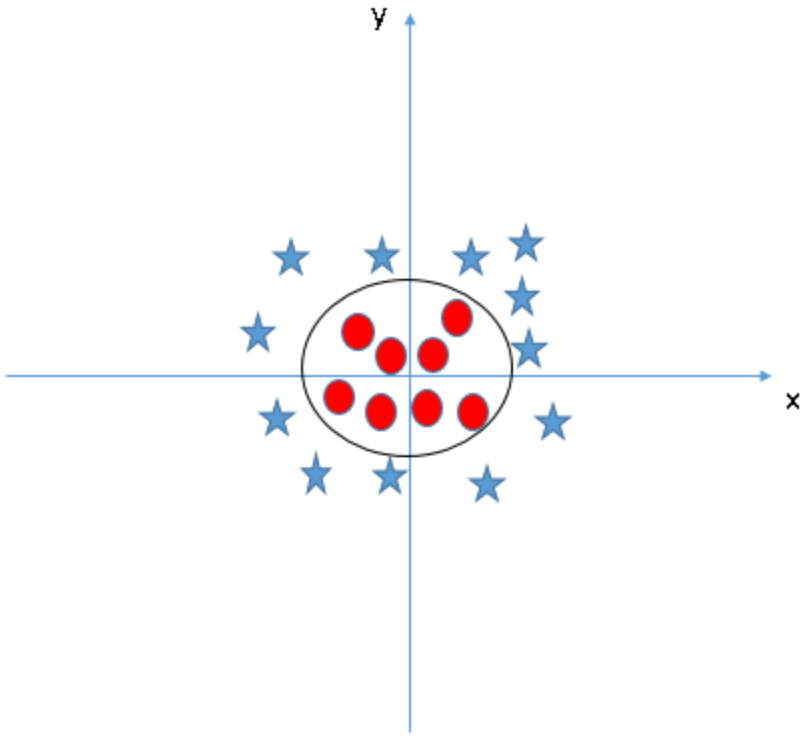


In above plot, points to consider are:

1. All values for z would be positive always because z is the squared sum of both x and y
2. In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

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.

When we look at the hyper-plane in original input space it looks like a circle:



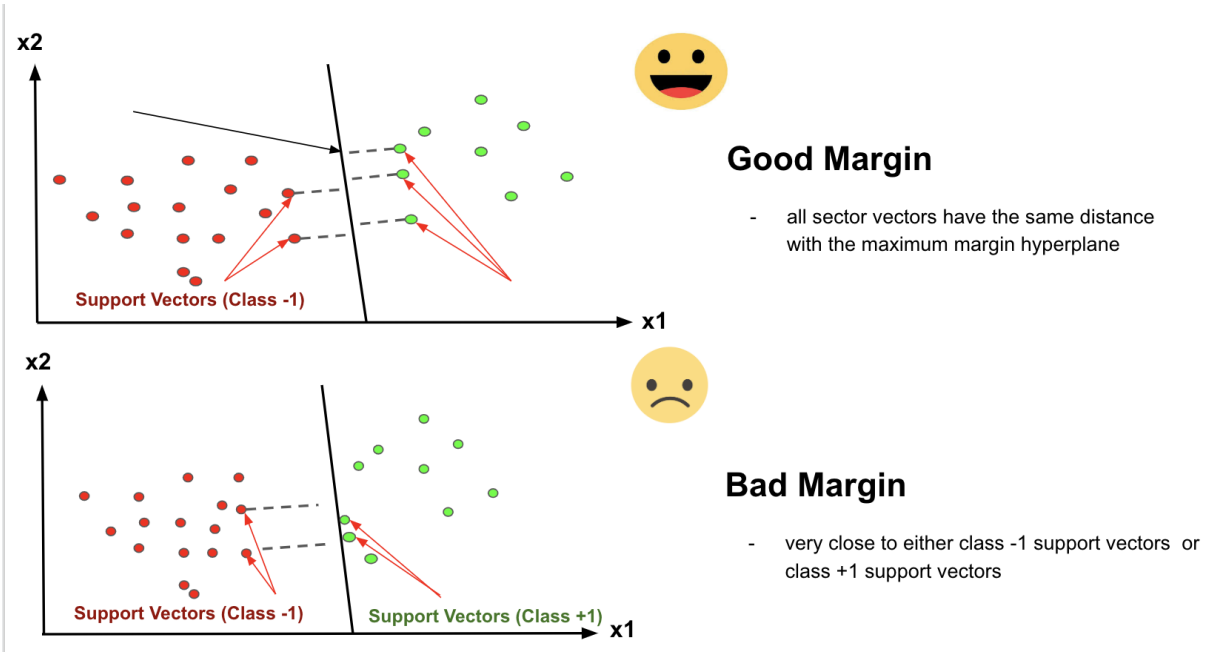
4. BASICS of SVM ALGORITHM:

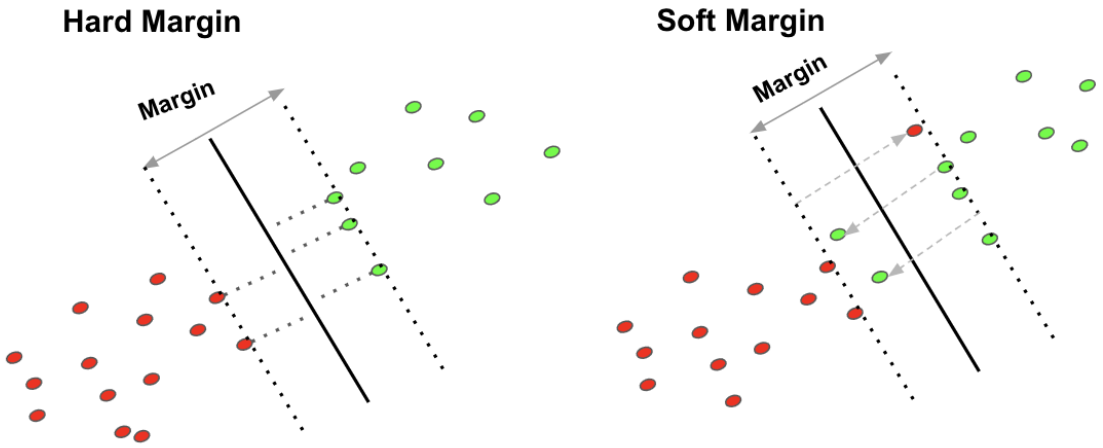
Support vector machine classification is a machine learning algorithm that supports classification of virtual objects using the vector points closest to a specific kind of line/line segment. The line/line segment can be a margin or a hyperplane. The Support Vector Machine Classification can also be used without the reference of a line/line segment. For e.g. the technique of using the kernel trick. These are functions that take low dimensional input space and transform it into a higher-dimensional space, i.e., it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems.

5. ALGORITHM:

The vector points closest to the hyperplane are known as the **support vector points** because only these two points are contributing to the result of the algorithm, and other points are not. If a data point is not a support vector, removing it has no effect on the model. On the other hand, deleting the support vectors will then change the position of the hyperplane.
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. It becomes difficult to imagine when the number of features exceeds 3.

The distance of the vectors from the hyperplane is called the ***margin,*** which is a separation of a line to the closest class points. We would like to choose a hyperplane that maximises the margin between classes. The graph below shows what good margin and bad margin are.





Hard Margin

If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible.

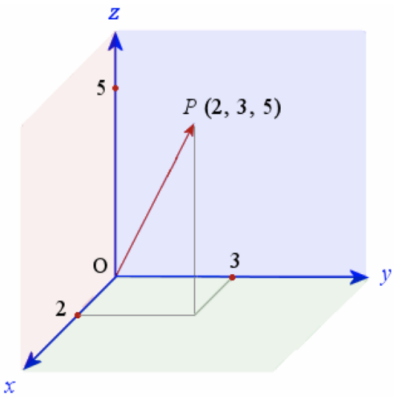
Soft Margin

As most of the real-world data are not fully linearly separable, we will allow some margin violation to occur, which is called soft margin classification. It is better to have a large margin, even though some constraints are violated. Margin violation means choosing a hyperplane, which can allow some data points to stay in either the incorrect side of the hyperplane and between the margin and the correct side of the hyperplane.

In order to find the **maximal margin**, we need to maximize the margin between the data points and the hyperplane. In the following session, I will share the mathematical concepts behind this algorithm.

Linear Algebra Revisited

Before we move on, let's review some concepts in Linear Algebra.



We can expand our 2-dimensional (x-y) coordinate system into a 3-dimensional coordinate system, using x-, y-, and z-axes. The general point P (2, 3, 5) is shown in graph.

A vector is denoted by a bolded alphabet (e.g., $\mathbf{x}$).

The magnitude of a n-dimensional vector can be calculated as follows:

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Note that

$$y = ax + b$$

is the same thing as

$$y - ax - b = 0$$

Given two vectors $\mathbf{w}\begin{pmatrix} -b \\ -a \\ 1 \end{pmatrix}$ and $\mathbf{x}\begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$

$$\mathbf{w}^T\mathbf{x} = -b \times (1) + (-a) \times x + 1 \times y$$

$$\mathbf{w}^T\mathbf{x} = y - ax - b$$

Maximising the Margin

You probably learned that an equation of a line is **y=ax+b**. However, you will often find that the equation of a hyperplane is defined by:

Note that

$$y = ax + b$$

is the same thing as

$$y - ax - b = 0$$

Given two vectors $\mathbf{w}\begin{pmatrix} -b \\ -a \\ 1 \end{pmatrix}$ and $\mathbf{x}\begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$

$$\mathbf{w}^T\mathbf{x} = -b \times (1) + (-a) \times x + 1 \times y$$

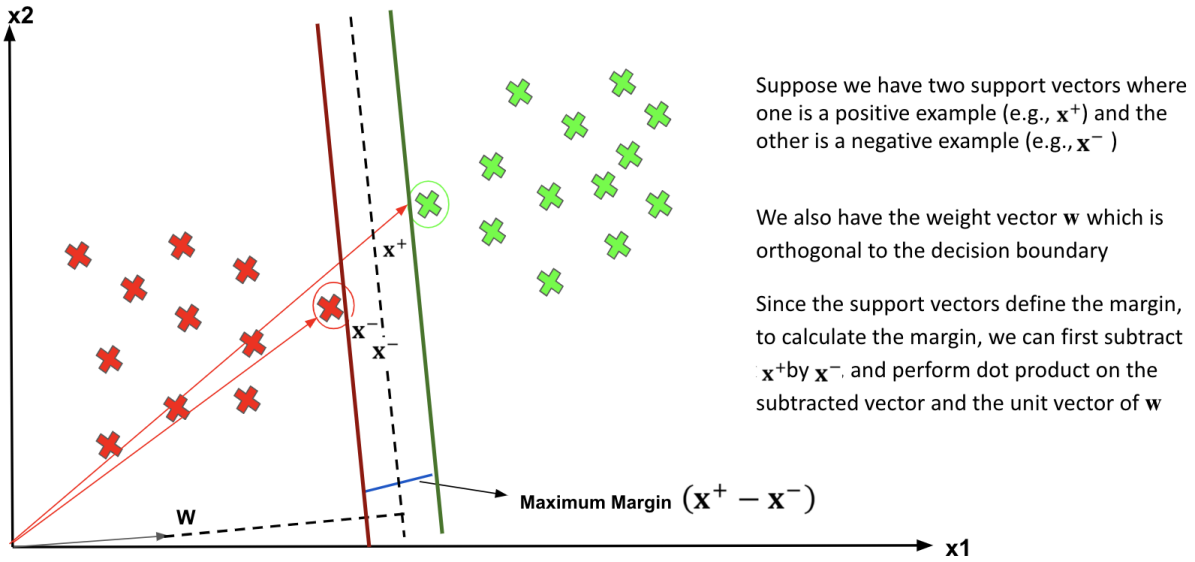
$$\mathbf{w}^T\mathbf{x} = y - ax - b$$

The two equations are just two different ways of expressing the same thing.

For **Support Vector Classifier** (SVC), we use **w**T**x**+*b* where **w** is the weight vector, and *b* is the bias.

$$\mathbf{w}^T\mathbf{x} + b = 0$$

You can see that the name of the variables in the hyperplane equation are **w** and **x,** which means they are vectors! A vector has magnitude (size) and direction, which works perfectly well in 3 or more dimensions. Therefore, the application of "**vector"** is used in the SVMs algorithm.

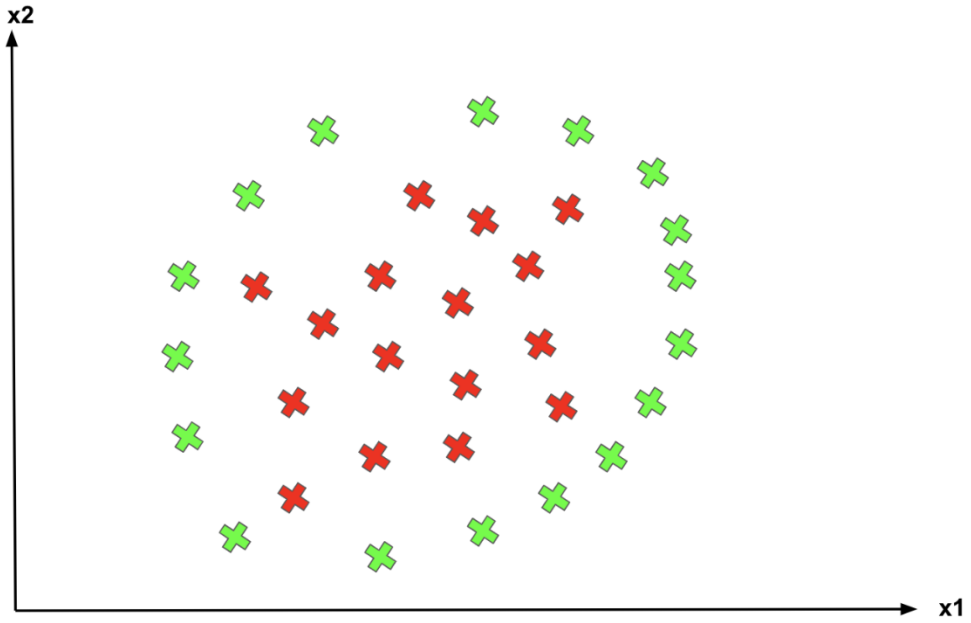


$$(\mathbf{x}^+ - \mathbf{x}^-) \cdot \hat{\mathbf{w}} = (\mathbf{x}^+ - \mathbf{x}^-) \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{x}^+ \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} - \mathbf{x}^- \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

The equation of calculating the Margin.

Classifying non-linear data

What about data points are not linearly separable?

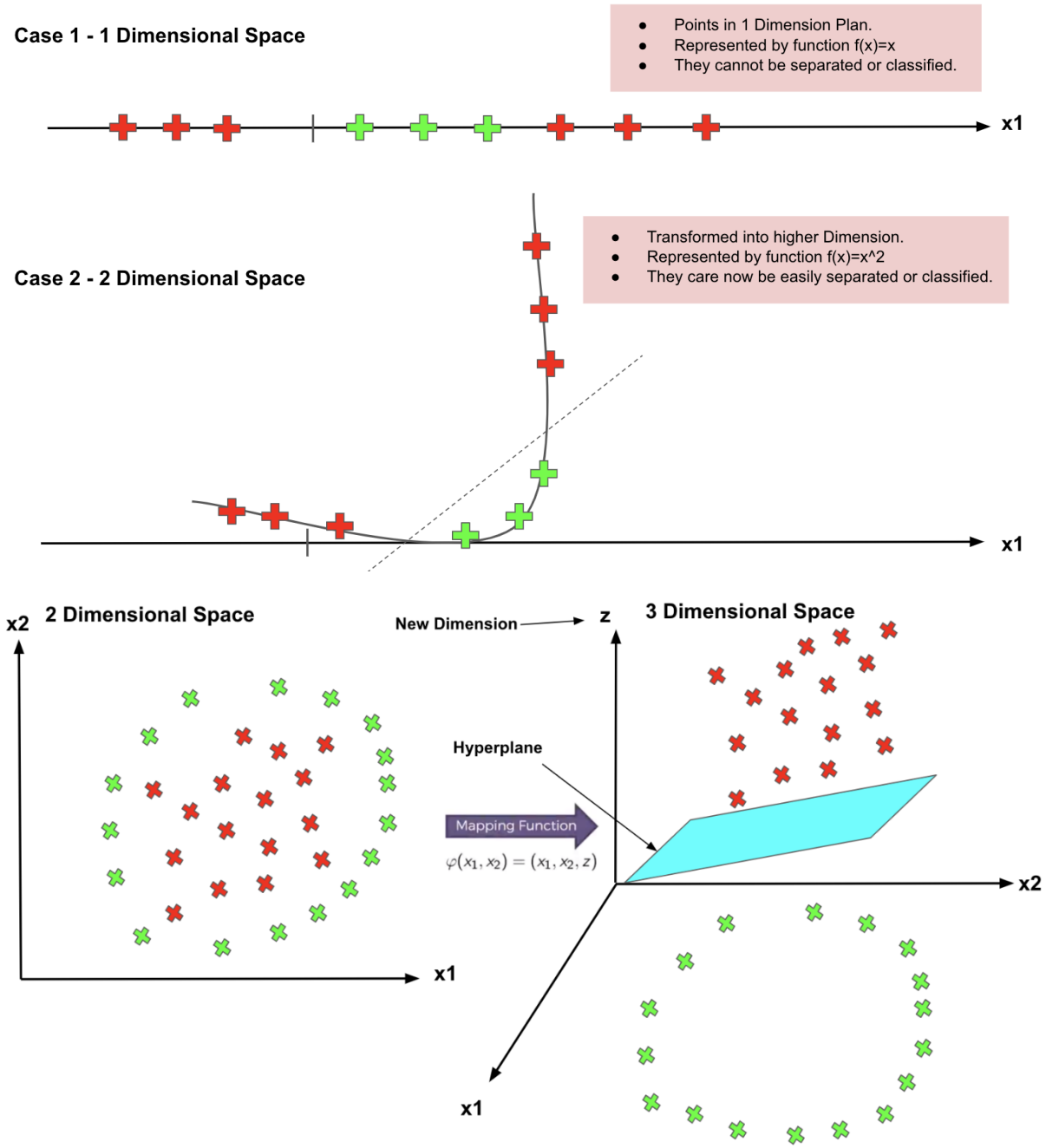


Non-linear separate.

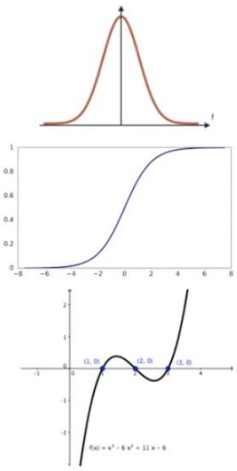
SVM has a technique called the **kernel trick**. These are functions that take low dimensional input space and transform it into a higher-

dimensional space, i.e., it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. This is shown as follows:

Mapping to a Higher Dimension



Some Frequently Used Kernels



Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

Sigmoid Kernel

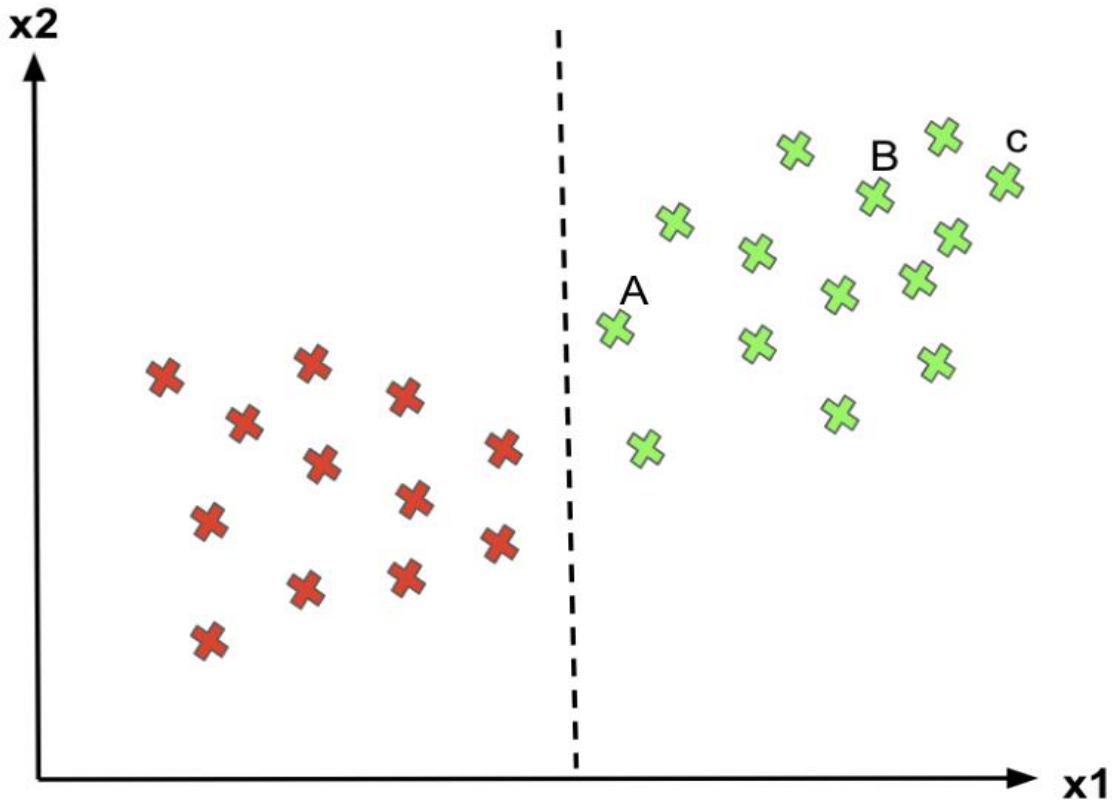
$$K(X, Y) = \tanh(\gamma \cdot X^T Y + r)$$

Polynomial Kernel

$$K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

6. IMPORTANCE OF SVM OVER LOGISTIC REGRESSION:

It helps solve classification problems separating the instances into two classes. However, there is an infinite number of decision boundaries, and Logistic Regression only picks an arbitrary one.



- For point C, since it's far away from the decision boundary, we are quite certain to classify it as 1.
- For point A, even though we classify it as 1 for now, since it is pretty close to the decision boundary, if the boundary moves a little to the right, we would mark point C as "0" instead. Hence, we're much more confident about our prediction at A than at C

Logistic Regression doesn't care whether the instances are close to the decision boundary. Therefore, the decision boundary it picks may not be optimal. As we can see from the above graph, if a point is far from the decision boundary, we may be more confident in our predictions. Therefore, the optimal decision boundary should be able to maximize the distance between the decision boundary and all instances. i.e., maximize the margins. That's why the SVM algorithm is important!

7. ADVANTAGES:

- It works really well with a clear margin of separation
- It is effective in high dimensional spaces.
- It is effective in cases where the number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

8. DISADVANTAGES:

- It doesn't perform well when we have large data set because the required training time is higher
- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of Python scikit-learn library.

9. Why SVM is an application of linear algebra?

Through out the above journey we realised that SVM model uses hyperplane/vector to distinguish the arbitrary non separable vector spaces.

Here vector spaces are defined as any kind of arbitrary objects placed in a certain group and have different categories. Simply different vector spaces means different categories.

10. APPLICATIONS:

1. Face detection – SVM classify parts of the image as a face and non-face and create a square boundary around the face.

2.Text and hypertext categorization – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.

3. Classification of images – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.

4. Bioinformatics – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.

5. Protein fold and remote homology detection – Apply SVM algorithms for protein remote homology detection.

6.Handwriting recognition – We use SVMs to recognize handwritten characters used widely.

7. Generalized predictive control (GPC) – Use SVM based GPC to control chaotic dynamics with useful parameters.

11. CONCLUSION:

It's not a wonder how a small concept of vector spaces in linear algebra can become the basis of a whole new machine learning algorithm called the Support Vector Machine Classification.

This very elegant algorithm has enabled so much of advancement in machine learning supporting the backbone of Artificial Intelligence.

12. BIBLIOGRAPHY:

1. https://www.kdnuggets.com/2020/03/machine-learning-algorithm-svm-explained.html

2. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

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

4. https://www.analyticsvidhya.com/blog/2019/07/10-applications-linear-algebra-data-science/

5. Howard Anton Chris Rorres elementary linear algebra applications version 11th edition