educlash

Just Another Way To Learn

Motivation for Support Vector Machines

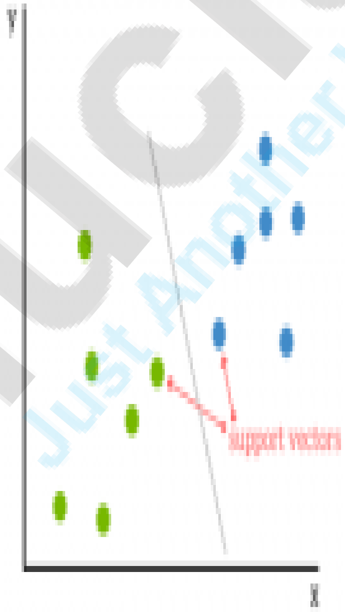
- The problem to be solved is one of the **supervised binary classification**. That is, we wish to categorize new unseen objects into two separate groups based on their properties and a set of known examples, which are already categorized.
- A good example of such a system is classifying a set of new *documents* into positive or negative sentiment groups, based on other documents which have already been classified as positive or negative.
- Similarly, we could classify new emails into spam or non-spam, based on a large corpus of documents that have already been marked as spam or non-spam by humans. SVMs are highly applicable to such situations.

Motivation for Support Vector Machines

- A Support Vector Machine models the situation by creating a *feature space*, which is a finite-dimensional vector space, each dimension of which represents a "feature" of a particular object. In the context of spam or document classification, each "feature" is the prevalence or importance of a particular word.
- The **goal of the SVM** is to train a model that assigns new unseen objects into a particular category.
- It achieves this by creating a linear partition of the feature space into two categories.
- Based on the features in the new unseen objects (e.g. documents/emails), it places an object "above" or "below" the separation plane, leading to a categorization (e.g. spam or non-spam). This makes it an example of a non-probabilistic linear classifier. It is non-probabilistic, because the features in the new objects fully determine its location in feature space and there is no stochastic element involved.

OBJECTIVES

- **Support vector machines (SVM)** are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
- It is a **machine learning** approach.
- They analyze the large amount of data to identify patterns from them.
- SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes, as shown in the image below.


y
support vectors
x

Support Vectors

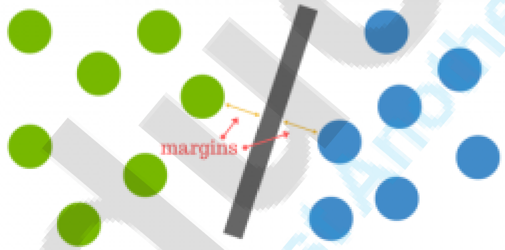
- Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).
- Support vectors are the data points that lie closest to the decision surface (or hyperplane)
- They are the data points most difficult to classify
- They have direct bearing on the optimum location of the decision surface
- We can show that the optimal hyperplane stems from the function class with the lowest “capacity” (VC dimension).
- Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

What is a hyperplane?

- As a simple example, for a classification task with only two features, you can think of a hyperplane as a line that linearly separates and classifies a set of data.
- Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.
- So when new testing data are added, whatever side of the hyperplane it lands will decide the class that we assign to it.

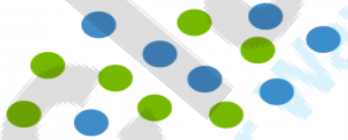
How do we find the right hyperplane?

- How do we best segregate the two classes within the data?
- The distance between the hyperplane and the nearest data point from either set is known as the **margin**. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly. **There will never be any data point inside the margin.**


margins

But what happens when there is no clear hyperplane?

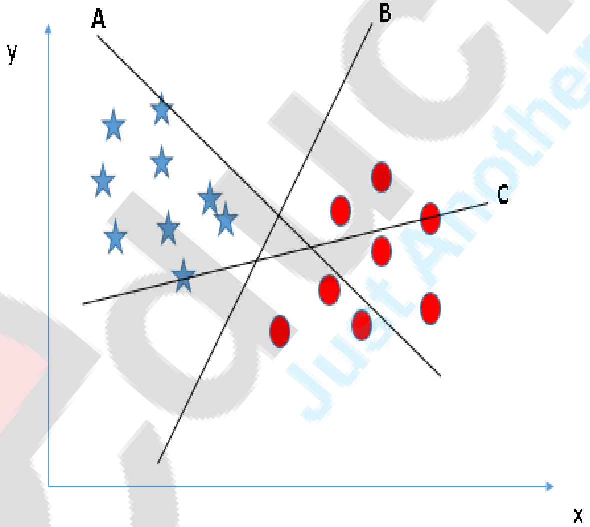
- Data are rarely ever as clean as our simple example above. A dataset will often look more like the jumbled balls below which represent a linearly non separable dataset.



- In order to classify a dataset like the one above it’s necessary to move away from a 2d view of the data to a 3d view. Explaining this is easiest with another simplified example. Imagine that our two sets of colored balls above are sitting on a sheet and this sheet is lifted suddenly, launching the balls into the air. While the balls are up in the air, you use the sheet to separate them. This ‘lifting’ of the balls represents the mapping of data into a higher dimension. This is known as **kernelling**.

Identify the right hyperplane (Scenario-1):

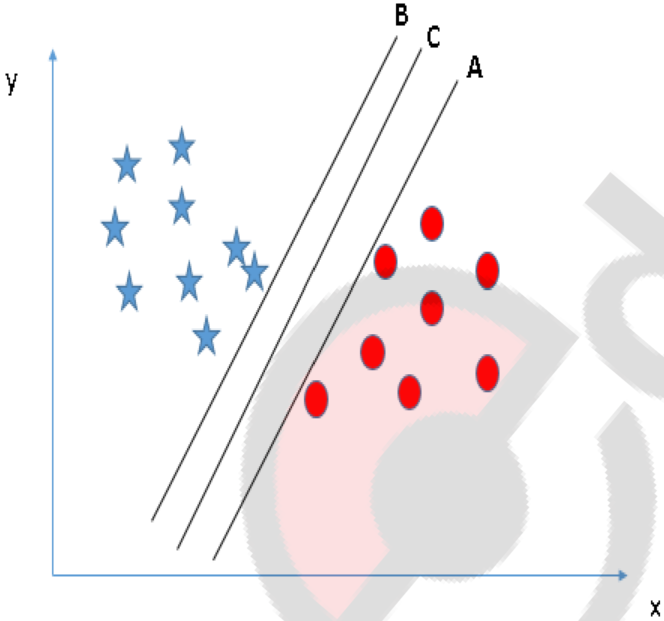
- Here, we have three hyperplanes (A, B and C). Now, identify the right hyperplane to classify star and circle.



- Hyperplane “B” has excellently performed this job.

Identify the right hyperplane (Scenario-2):

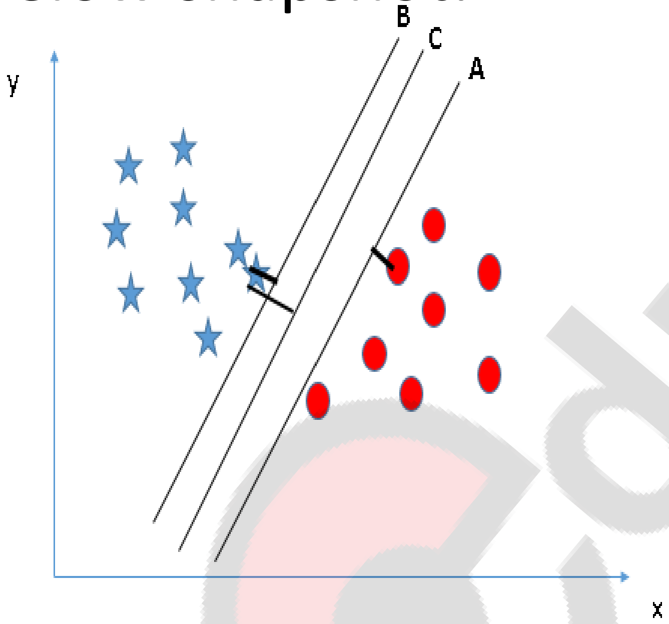
- Here, we have three hyperplanes (A, B and C) and all are segregating the classes well. Now, ify the right hyperplane?



Here, maximizing the distances between nearest data point (either class) and hyperplane will help us to decide the right hyperplane.

Scenario-2

This distance is called as **Margin**. Let’s look at the below snapshot:



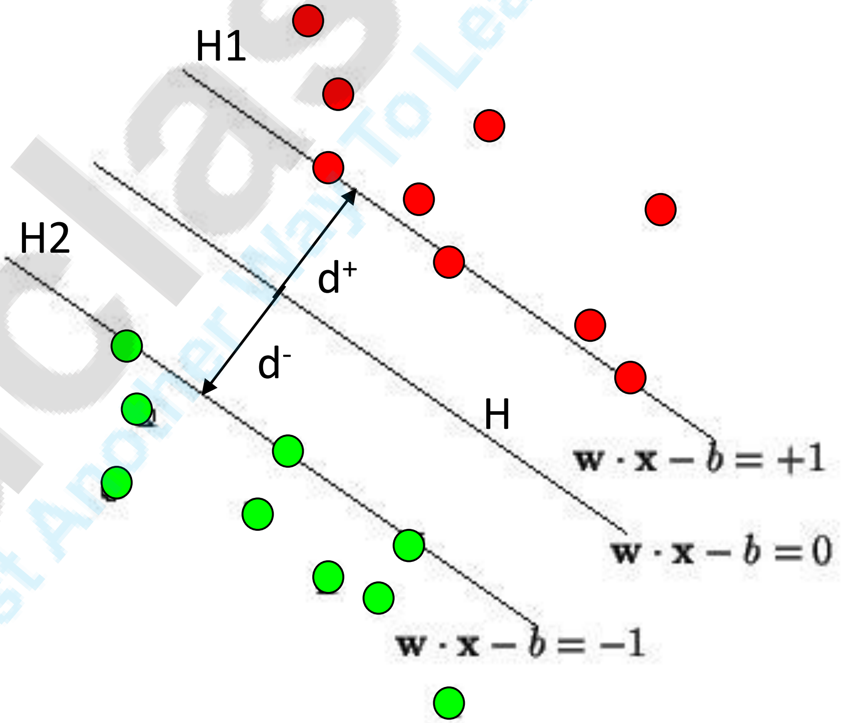
We can see that the margin for hyperplane C is high as compared to both A and B. Hence, we name the right hyperplane as C. Another lightning reason for selecting the hyperplane with higher margin is robustness. If we select a hyperplane having low margin then there is high chance of missclassification

Definitions

Define the hyperplane H such that:

$\mathbf{x}_i \bullet \mathbf{w} + b \geq +1$ when $y_i = +1$

$\mathbf{x}_i \bullet \mathbf{w} + b \leq -1$ when $y_i = -1$


H1
H2
d+
d-
H
$\mathbf{w} \cdot \mathbf{x} - b = +1$
$\mathbf{w} \cdot \mathbf{x} - b = 0$
$\mathbf{w} \cdot \mathbf{x} - b = -1$

H1 and H2 are the planes:

H1: $\mathbf{x}_i \bullet \mathbf{w} + b = +1$

H2: $\mathbf{x}_i \bullet \mathbf{w} + b = -1$

The points on the planes H1 and H2 are the Support Vectors

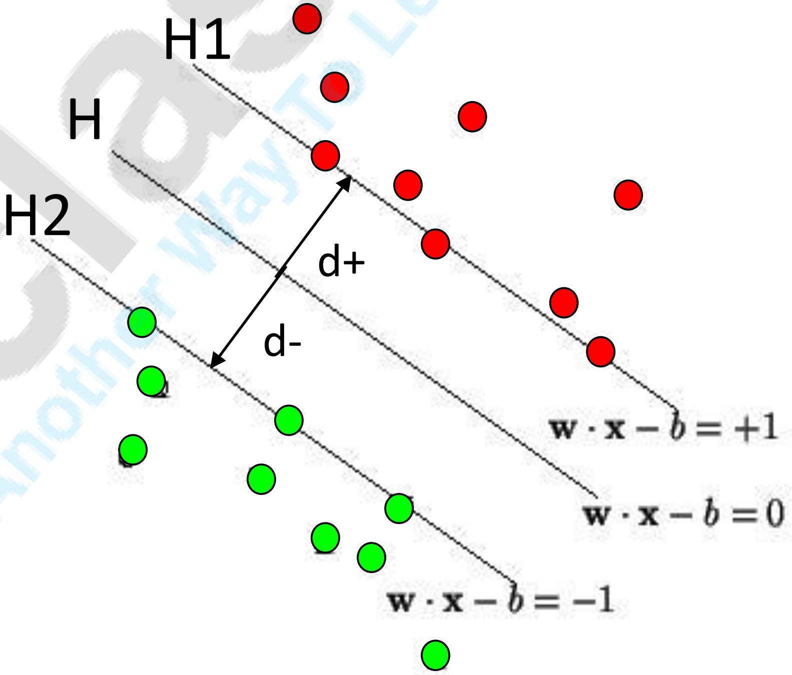
d+ = the shortest distance to the closest positive point

d- = the shortest distance to the closest negative point

The margin of a separating hyperplane is $d^+ + d^-$.

Maximizing the margin

We want a classifier with as big margin as possible.


H1
H
H2
d+
d-
$\mathbf{w}\cdot\mathbf{x}-b=+1$
$\mathbf{w}\cdot\mathbf{x}-b=0$
$\mathbf{w}\cdot\mathbf{x}-b=-1$

Recall the distance from a point(x_0,y_0) to a line: Ax+By+c = 0 is|A x_0 +B y_0 +c|/sqrt(A^2+B^2)

The distance between H and H1 is: |**w•x**+b|/||w||=1/||w||

The distance between H1 and H2 is: 2/||w||

In order to maximize the margin, we need to minimize ||w||. With the condition that there are no datapoints between H1 and H2:

$\mathbf{x}_i$•**w**+b ≥ +1 when y_i =+1
$\mathbf{x}_i$•**w**+b ≤ -1 when y_i =-1

Maximum Margin: Formalization

- **w**: decision hyperplane normal vector
- $\mathbf{x}_i$: data point *i*
- y_i: class of data point *i* (+1 or -1) NB: Not 1/0
- Classifier is: $f(\mathbf{x}_i) = \text{sign}(\mathbf{w}^T\mathbf{x}_i + b)$
- Functional margin of $\mathbf{x}_i$ is: $y_i\,(\mathbf{w}^T\mathbf{x}_i + b)$
  - But note that we can increase this margin simply by scaling **w**, **b**….
- Functional margin of dataset is twice the minimum functional margin for any point
  - The factor of 2 comes from measuring the whole width of the margin