MLT-02 Summary Document

Overview

This document summarizes the lecture MLT-02 on Machine Learning. This lecture will help you understand support vector machines and the K-means clustering algorithm.

The lecture covers the following topics:

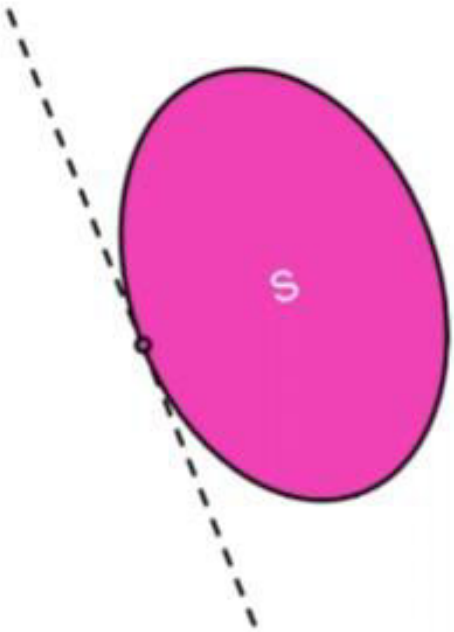
- Support vector machines: Introduction
- Supporting and Separating Hyperplane
- Choosing the Best Hyperplane Linear and Non-Linear SVM
- Kernels
- Soft Margin vs Hard Margin
- Clustering
- Similarity Measure
- Clustering approaches
- K-Means Clustering

Support vector machines: Introduction

The regression purpose is to predict the target variable based on the fitting of an equation to the target variable. The support vector machine is a supervised machine learning technique that makes a classification or a regression analysis. It takes the attributes' values and tries to separate them as per the target variable. This separation is given by a line (or plane) or vector that supports this separation, hence its name.

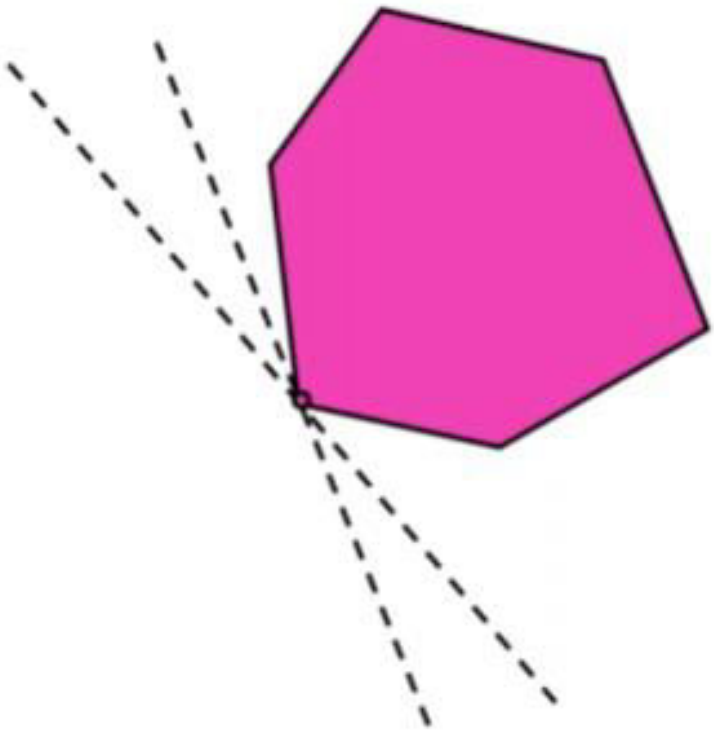
Choosing the Best Hyperplane Linear and Non-Linear SVM

Imagine S is a set, as in the following graph:

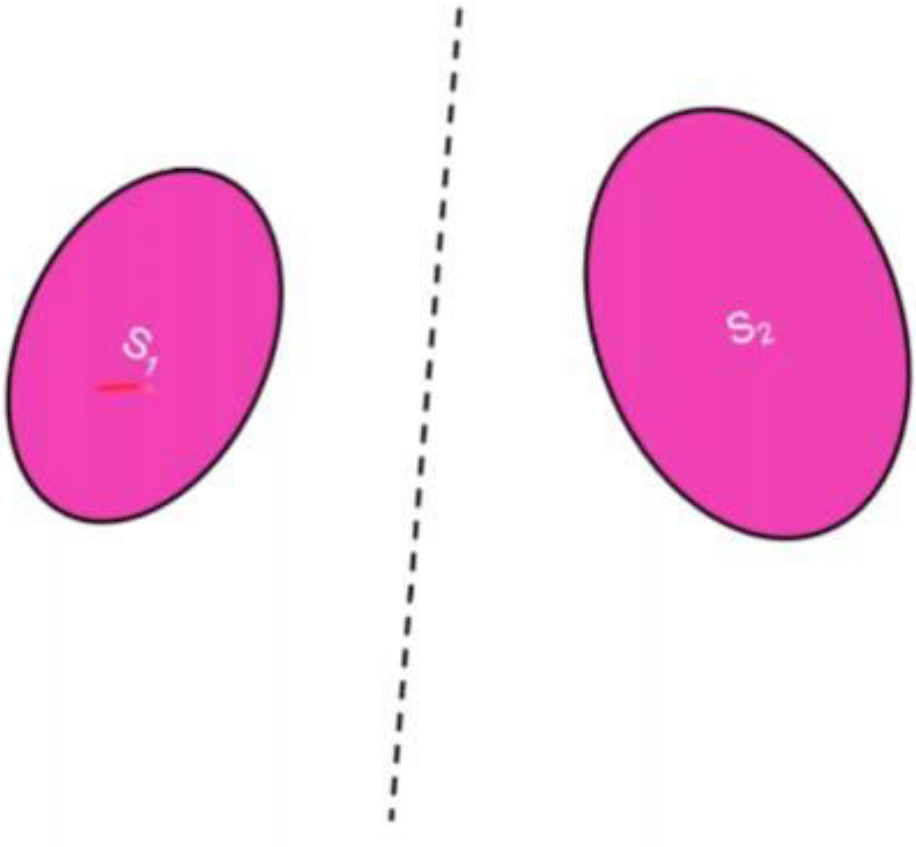


A supporting hyperplane of a set S that has both of the following two properties:

- S is entirely contained in one of the half-spaces.
- S has at least one boundary point on the hyperplane.



In the new graph above, more than one hyperplane may be possible that meets the supporting hyperplane criteria.



In this new example, we have 2 sets, set 1 and set 2. You put a plane between them. But which plane will be the best to separate both sets? We can decide it using the hyperplane equation given below:

$$a_1x_1 + a_2x_2 + \ldots + a_3x_3 + a_4x_4 + b = 0$$

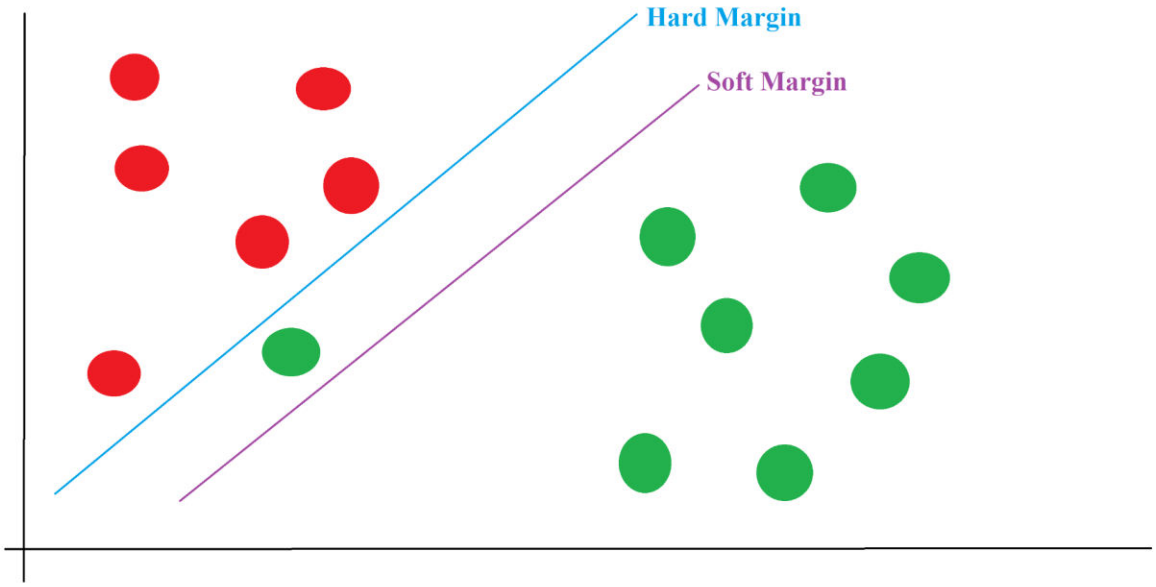
Kernels

Kernels are the support vector machine hyperplane equation that will be used to fit the classification onto the data. A kernel can be Polynomial, Gaussian or Non linear.

Be careful with non-linear kernels. Even though you might get a perfect fit for you train dataset, it may perform worse than a linear kernel on the test data.

Soft Margin vs Hard Margin

In case you use a linear kernel and allow some errors while fitting the model, then you say you have a soft margin. When you don't allow any error in case it's possible to do it, then you fit a hard margin. Hard margin can lead to overfitting, so do not be so strict while fitting the model.



Clustering

In any stock market, you can group or cluster companies based on features' similarities. Some of these features can be their betas, their company size, their sector, their price to book, etc. You can group the companies by more than one feature.

This is what is called clustering and you can do this for any asset class. Clustering is done without any target variable. Actually, you create the groups while fitting the clustering model. Thus, clustering is an unsupervised machine learning algorithm.

Similarity Measure

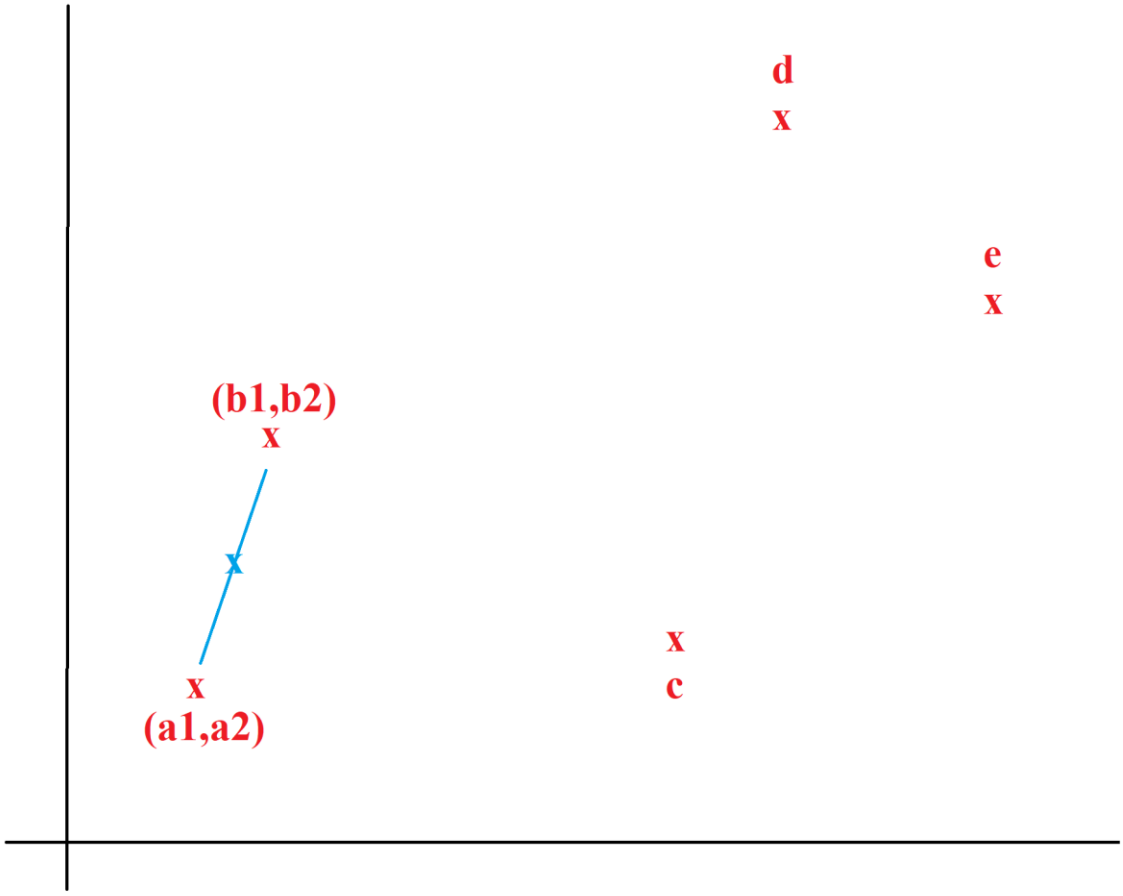
In order to cluster assets per their similarities, you need a similarity measure to compare features' values. You can measure this similarity between two assets by:

- Euclidean distance
- Manhattan distance: It is the sum of the absolute distance between the values in all the dimensions of 2 points.
- Mahalanobis distance: It is a weighted sum of the absolute distance between the values in all the dimensions between 2 points.

Clustering approaches

Hierarchical clustering: Find objects that are the closest between them and group the objects as per this closeness.

Let's see an example: Imagine we have 5 objects, a, b, c, d and e.



As you see in the above graph, points a and b have the closest distance. This distance (highlighted in blue collour) is calculated as:

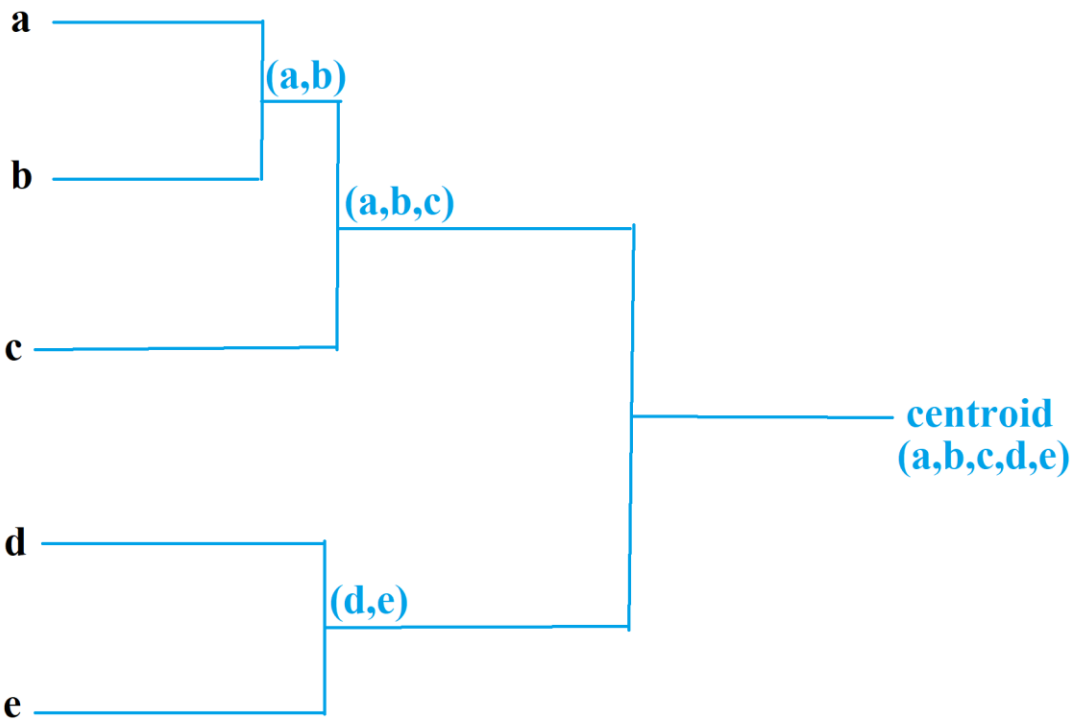
$$\left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}\right)$$

Then you can cluster the a-b distance with the c point, since this object has the closest distance with a and b, compared to d and e. Thus, the three-object distance is given by:

$$\left(\frac{a_1 + b_1 + c_1}{3}, \frac{a_2 + b_2 + c_2}{3}\right)$$

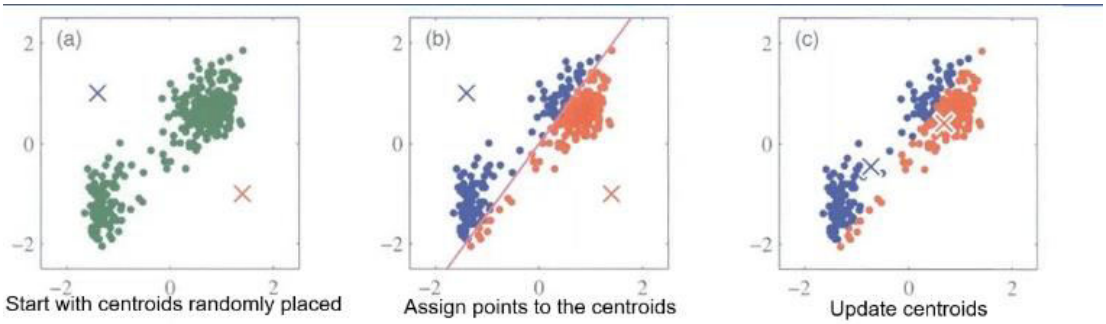
Besides, d and e are really close, so we can cluster them. Finally, we cluster the a-b-c cluster with the d-e cluster. This high hierarchical cluster will be called the centroid.

So the hierarchical clustering figure might look like this:



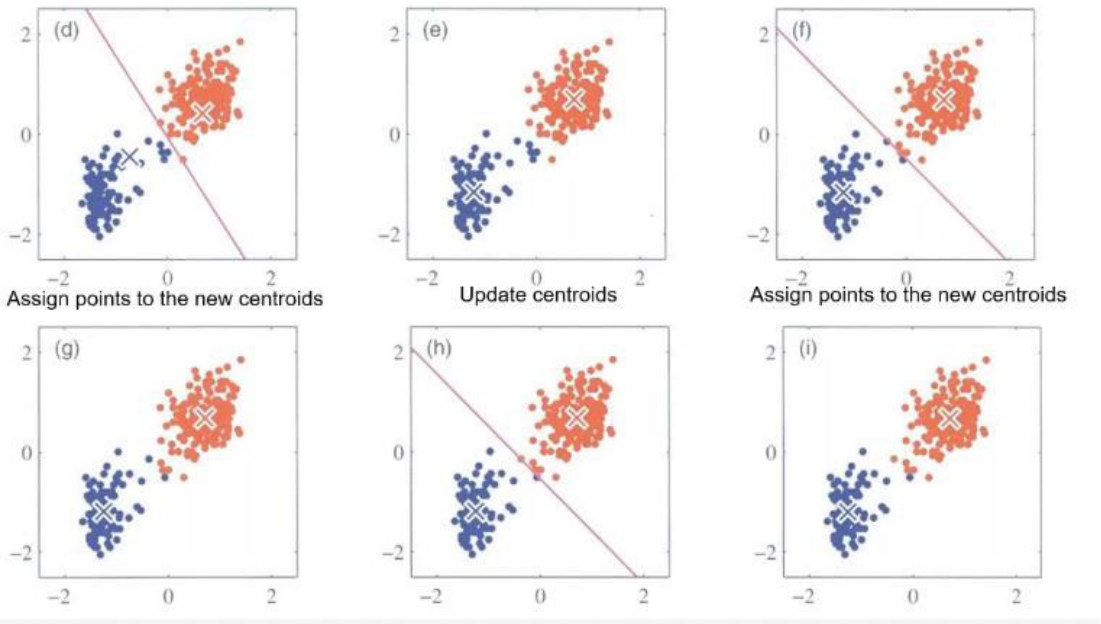
K-Means Clustering

This unsupervised machine learning algorithm tries to cluster the objects through an iteration of consecutive random generated centroids. These centroids are updated through each iteration based on minimising the distance between the points and the objects.



You start with centroids that are chosen randomly (blue and red x). Then you assign the points to each x as per the closest distance to them. Next, you update the centroids to make them match better with the data. This is one

iteration. The loop or process is repeated until you minimize the distance between the points and the centroids. As you can see below:



Whenever you want to classify assets and you don't have a specified classification, you can use this algorithm. For example, stocks can be classified by their industry sector. However, this algorithm can be used to cluster stocks as per some financial ratios or financial indicators of the companies, etc.