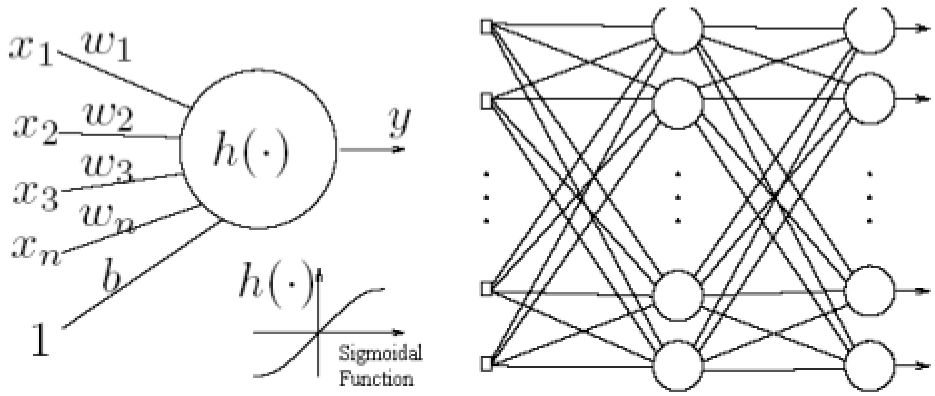
SVM

Supervised learning methods such as support vector machines (SVMs) are utilised for classification and regression analysis. They're a member of the generalised linear classifiers family of algorithms. Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to enhance predicted accuracy while automatically avoiding overfitting to the data.

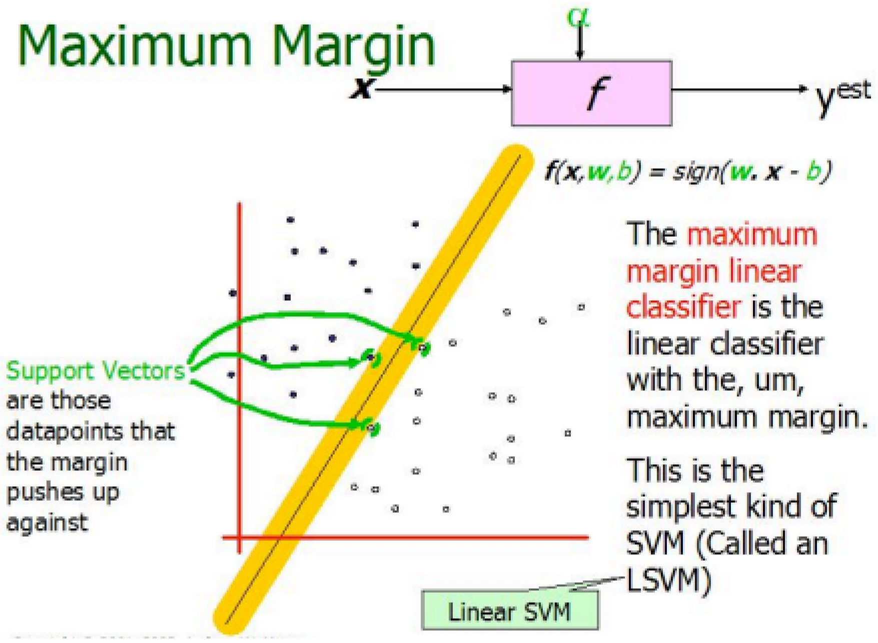
Introduction to Support Vector Machines: Why SVM?

To begin, experiments using neural networks for supervised and unsupervised learning yielded promising results. There are two types of networks used by MLP's(Multilayer perceptron): ***feed forward and recurrent***. It includes universal approximation of continuous nonlinear functions, learning with input-output patterns, and multilayer network designs with numerous inputs and outputs in the MLP characteristics.



Simple Neural Network(Left), Multilayer perceptron(Right)

Classifying the data is done using a large number of linear classifiers (hyper planes). There's just one way to get the most distance apart from the other two. A hyperplane classifier may wind up classifying more closely with one set of datasets than with others, which is something we do not want to happen. This leads us to see the concept of a maximum margin classifier or hyper plane as an obvious answer to our problem.



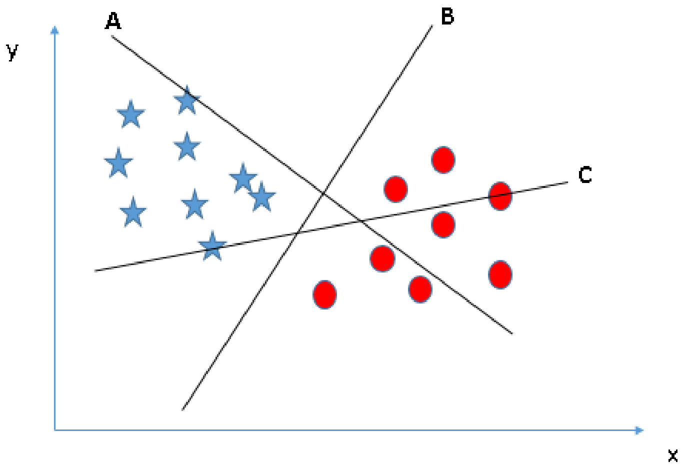
How does it work?

We became acquainted with the procedure of segregating the two classes using a hyper-plane in the preceding section. We now have to figure out how to find the correct hyper-plane. Don't be alarmed; it's not as difficult as you may expect.

Let's see if we can grasp this:

Scenario 1: Find the most appropriate hyper-plane:

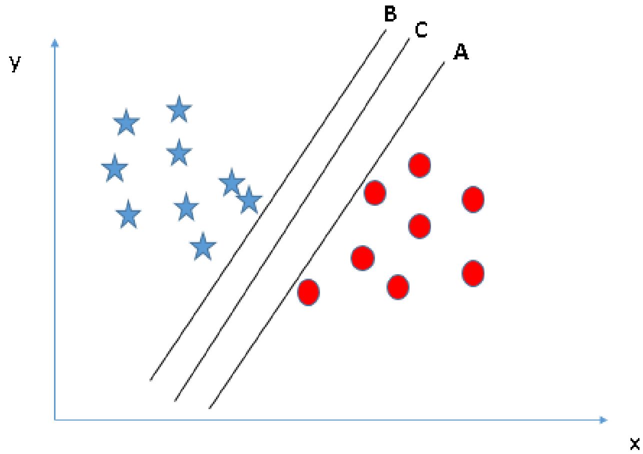
Three hyper-planes can be seen here (A, B, and C). Find the correct hyper-plane to categorise the stars and circles now.



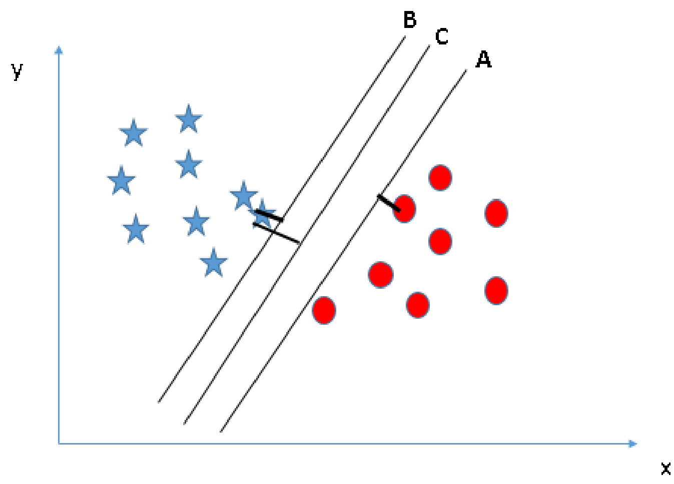
To choose the appropriate hyper-plane, you must remember the following rule:
"Select the hyper-plane that best segregates the two classes." In this instance, hyper-plane "B" functioned wonderfully.

Scenario 2: Determine the appropriate hyper-plane:

Here, we have three hyper-planes (A, B, and C), each of which effectively segregates the classes. Now, how do we find the correct hyperplane?

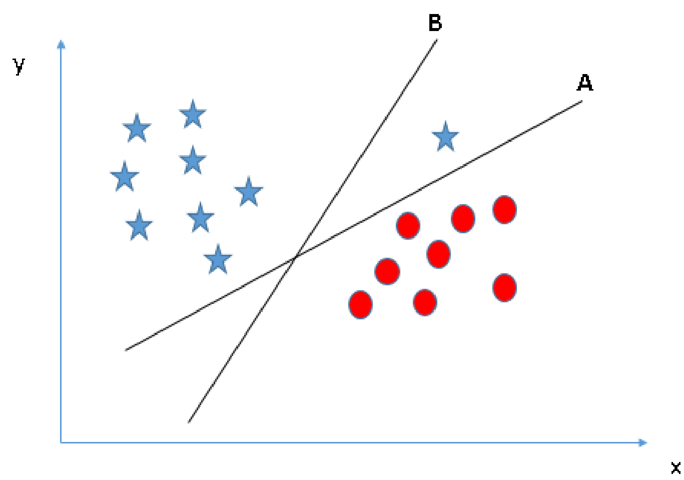


In this case, the optimal hyper-plane will be determined by maximising the distances between the nearest data points (of either class). Margin is the name given to this distance. Take a look at the image below:



Hyper-plane C has a large margin over hyper-planes A and B, as seen above. As a result, we'll refer to the correct hyper-plane as C. A hyper-plane with a larger margin is preferred because of its increased resilience. The likelihood of misclassification increases if we choose a hyperplane with a small margin.

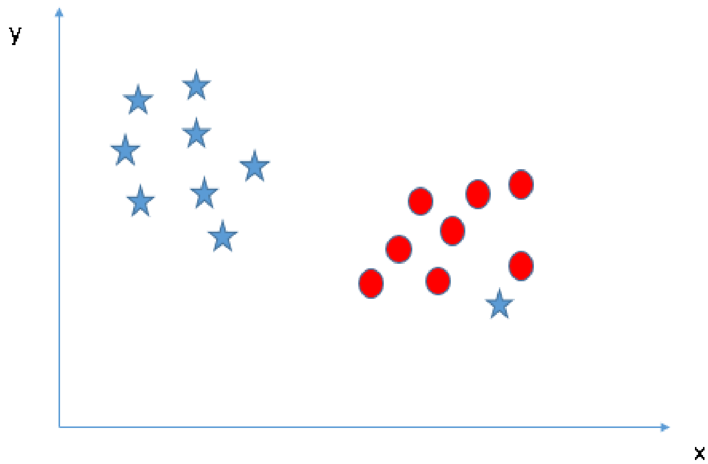
Scenario 3: Select the appropriate hyperplane:



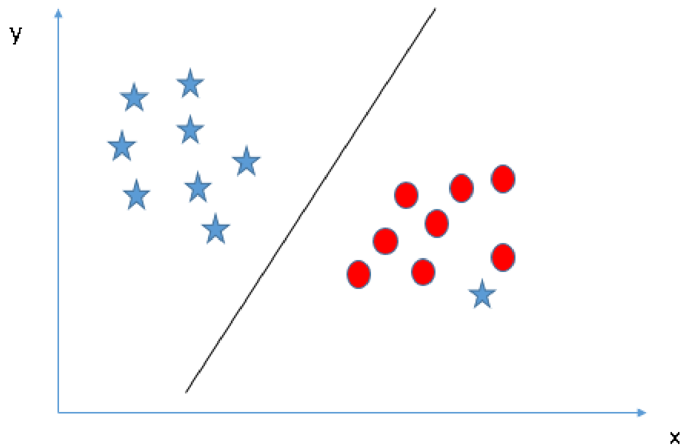
Some of you may have gone for hyper-plane B instead of A because of the higher margin. However, and this is the catch, SVM chooses the hyper-plane that accurately classifies the classes before maximising the margin. Hyper-plane B makes a faulty categorization here, while hyper-plane A gets it right every time. As a result, A is the appropriate hyper-plane.

Scenario 4: Is it possible to divide the classes into two groups?

Below, I'm unable to draw a straight line to separate the two groups because an outlier star lies in the region of the other(circle) group.

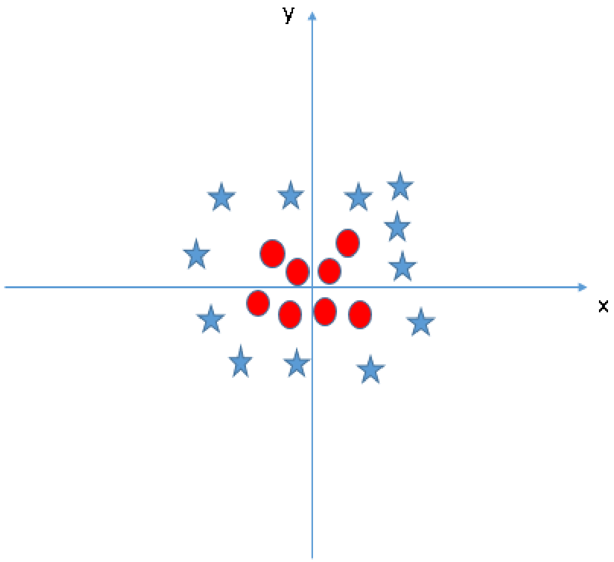


As I previously stated, the star at the opposite end acts as an outlier for its star class. The SVM method includes a feature that allows it to ignore outliers and identify the hyperplane with the greatest margin. As a result, we can assert that SVM classification is resistant to outliers.

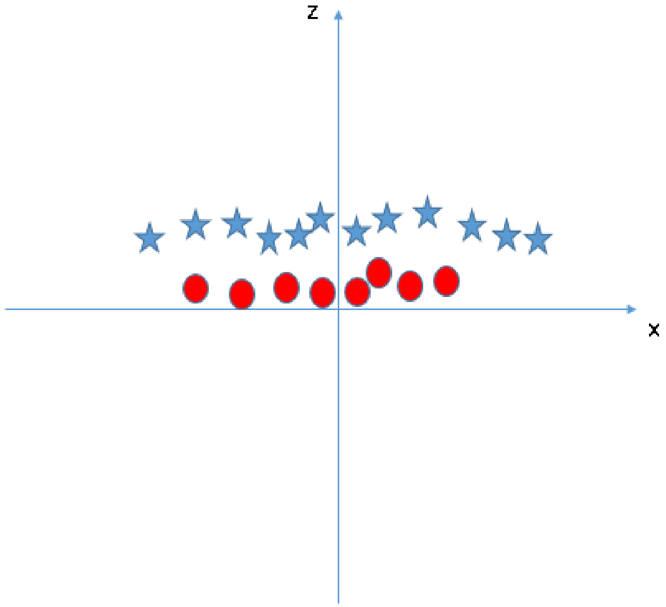


Scenario 5: Segregate classes by finding the hyper-plane:

Because there is no linear hyper-plane between the two classes in the example below, how can SVM classify them? We've just studied the linear hyper-plane up to this point.



This issue can be resolved with SVM. Easily! This issue is addressed by adding new capabilities. New functionality z=x2+y2 will be added here. Once you've done that, you're ready to plot the points on the axis.

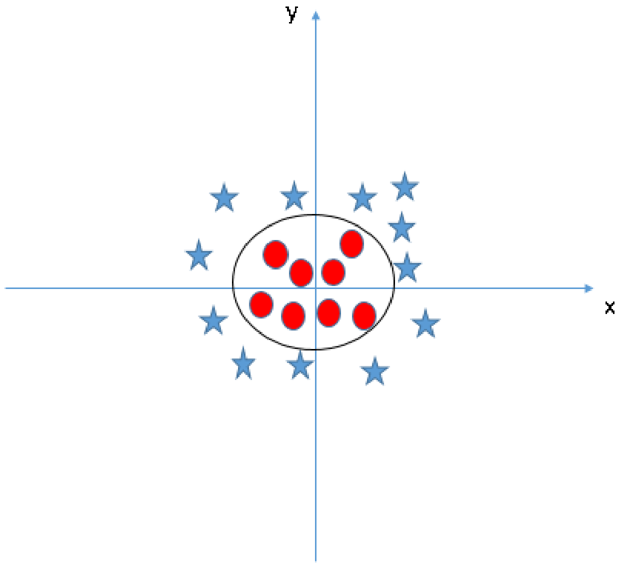


Points to keep in mind when looking at the above plot include:

- Because z equals the sum of x and y, all z values will be positive.
- The lower the z value, the closer the red circles are to the x and y axis origins, and the higher the z value, the further the stars are from the origins.

It is simple to have a linear hyper-plane between these two classes in the SVM classifier. However, another important topic that needs to be addressed is whether or not a hyper-plane can be created automatically. The kernel trick is a technique used by the SVM algorithm. For example, it takes non-separable problems and turns them into separate problems by using an SVM kernel to raise the spatial dimension of the input. It is highly effective in cases with non-linear separation. Just to be clear: It does some fairly complicated data transformations before determining how to split data depending on labels or outputs that you specify.

When viewed in original input space, the hyper-plane seems to be a circle:



Classification Using Support Vector Machines

SVM is an effective tool for categorising data. Despite the fact that Neural Networks are thought to be more user-friendly, undesirable results might still occur. Classification tasks often need the use of training and testing data, both of which are made up of many types of data. There are several target values and attributes in every instance of the training set. SVM's objective is to create a model that accurately predicts the target value of data instances in a testing set given just their properties.

Supervised learning is used in SVM classification. A system's performance can be gauged by looking at whether or not certain labels are being used. Using this data, the system can be taught to respond correctly or it can be used to validate its correctness. SVM categorization begins with identifying entities that are closely linked to previously identified classes. Features are selected or extracted in this manner. Even when predicting unknown samples isn't necessary, feature selection and SVM classification work well together. They can be used to identify key sets that are engaged in the processes that separate the various groups of people.

Regression Analysis Using SVM

By using a different loss function, SVMs can be used to solve regression issues. A distance measure must be incorporated into the loss function.

You have the option of having a linear or non-linear regression model. e-intensive loss functions, quadratic and Huber loss functions are the most common loss functions used in linear models, as are linear and quadratic loss functions.

A non-linear model is often required to adequately model data while dealing with categorization issues. A non-linear mapping can be used to map the data into a high-dimensional feature space where linear regression can be conducted in the same way as the nonlinear SVC technique. The curse of dimensionality is once again addressed using the kernel technique. The regression method takes into account the problem's prior knowledge as well as the noise's distribution. It's been proven that Huber's robust loss function is a good option when such information isn't available.

Uses for Support Vector Machines

Using SVM for pattern classification tasks has shown to be a huge success so far. There are several problems that must be answered before the Support Vector technique can be applied to a particular practical issue. The selection of an appropriate kernel for a specific application poses a significant difficulty. While Gaussian and polynomial kernels are the default alternatives, more complex kernels will be required if they prove ineffective or if the inputs are discrete structures. The kernel supplies the descriptive language used by the machine to examine the data by automatically establishing a feature space.

Once the kernel and optimization criterion have been selected, the system's key components are in place. Let's take a look at a few real-world instances.

Language categorization is the task of classifying natural text documents according to their content into a set number of predetermined categories. Given that a single document might be allocated to more than one category, we can think of this as a collection of binary classification problems, one for every single category. An appropriate feature mapping for the construction of a Mercer kernel is provided by one of the common text representations for information retrieval.

Kernels do include a measure of similarity across instances, and specialists in the given application domain have likely already developed reliable similarity measures, notably in information retrieval and generative models.

When dealing with large datasets, traditional classification algorithms struggle due to the enormous dimensionality of the data. Support Vector Machines, on the other hand, excel at dealing with small datasets. Image classification can be approached in a way that is quite similar to the text categorization techniques discussed above, and linear hard margin machines are frequently able to generalise effectively in this instance as well. Hand-written character recognition was the first real-world assignment to put Support Vector Machines to the test. These data have also been used to evaluate multi-class SVMs. Comparing SVMs with other classifiers is intriguing, but so is comparing various SVMs with one another. Regardless of the kernel, they have about the same performance and share the majority of their support vectors. SVM's ability to outperform these systems despite the lack of extensive prior knowledge is quite impressive.

SVM's Strengths and Weaknesses:

In particular, SVM has the advantage of being simple to learn. In contrast to neural networks, there is no such thing as a local optimum. A classifier's complexity and error may be traded off directly, thus it scales well to large datasets. The flaw is that a good kernel function is required.

Conclusion:

SVMs are based on the statistical learning theory they're derived from. They can be employed in the process of learning to anticipate future outcomes based on historical data. Solving a confined quadratic optimization problem trains a support vector machine (SVM). SVM uses a set of nonlinear basis functions to map inputs into a high-dimensional space. There is a unique optimal solution for each decision in the SVM parameters because SVM may be used to learn a number of representations, including neural nets, splines, and polynomial estimators. Standard Neural Networks trained through back propagation are different in that they do not use back propagation. SVM's development differs significantly from those of other learning algorithms, and it offers new insights into how humans learn. Duality, kernels, convexity, and sparseness are the four most important SVM characteristics.