Machine Learning & Deep Learning

Week-9

Instructor: *Najam Aziz*

Classification Models/Algorithms

Support Vector Machines

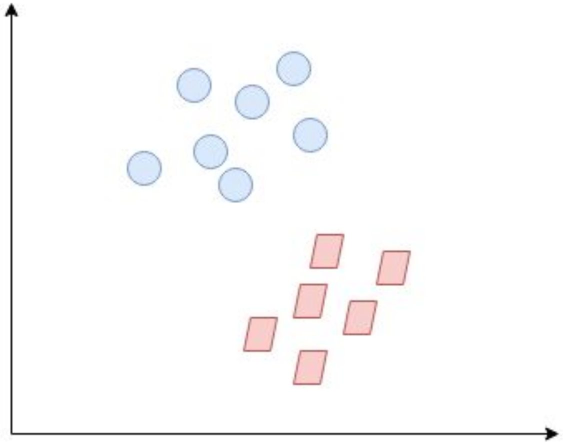
Use Case: what Kind of problem it can solve?

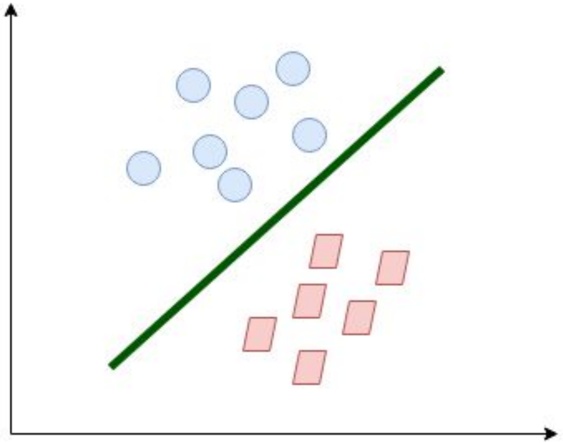
- **Regression Problems (SVR)**
  - predict the salary of an employee given a few independent variables.
- **Classification Problems (SVM)**
  - handwriting recognition, intrusion detection, face detection, email classification, gene classification
- **Outliers Detection**

-Can handle both classification and regression on linear and non-linear data. But generally use for classification problems

What exactly SVM does?

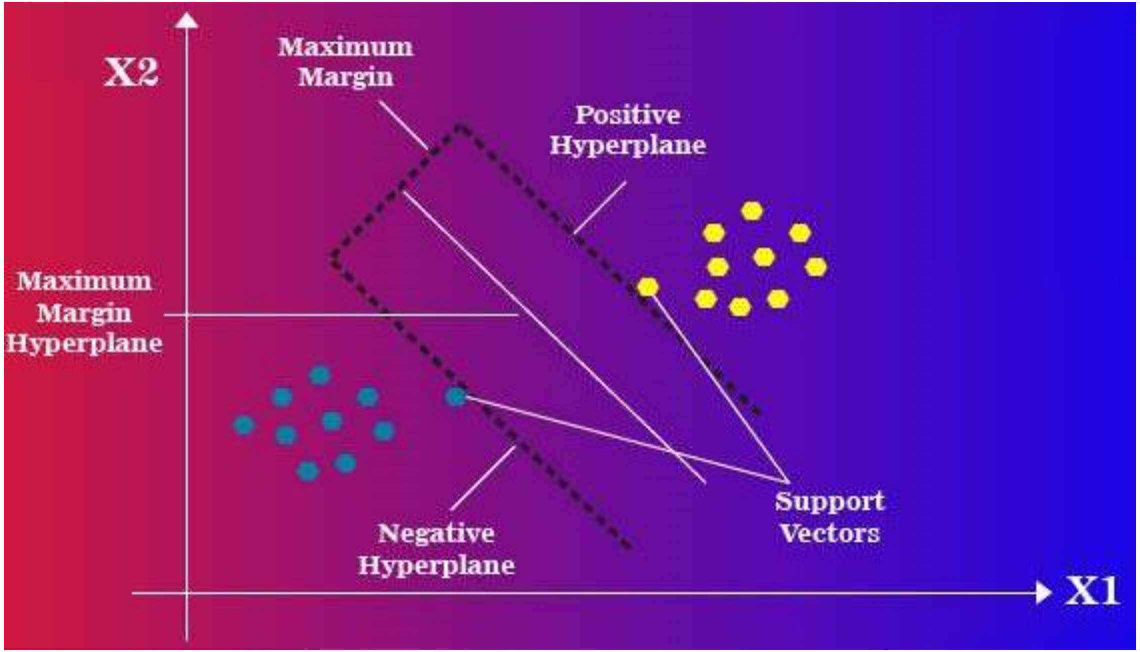
- Find a line/hyperplane (in multidimensional space) that separates these two classes. Then it classifies the new point depending on whether it lies on the positive or negative side of the hyperplane depending on the classes to predict.

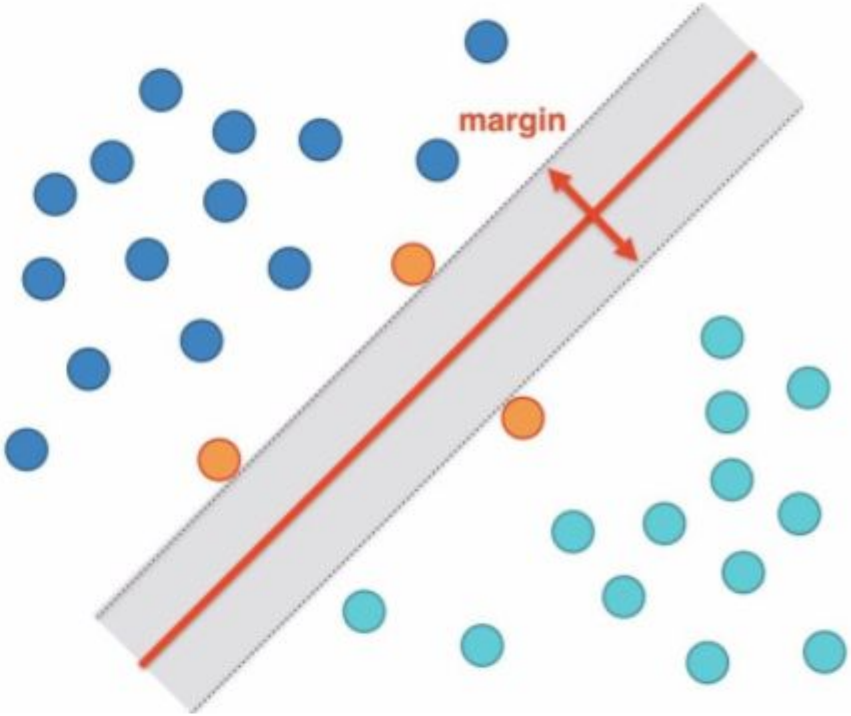




Basic Concepts

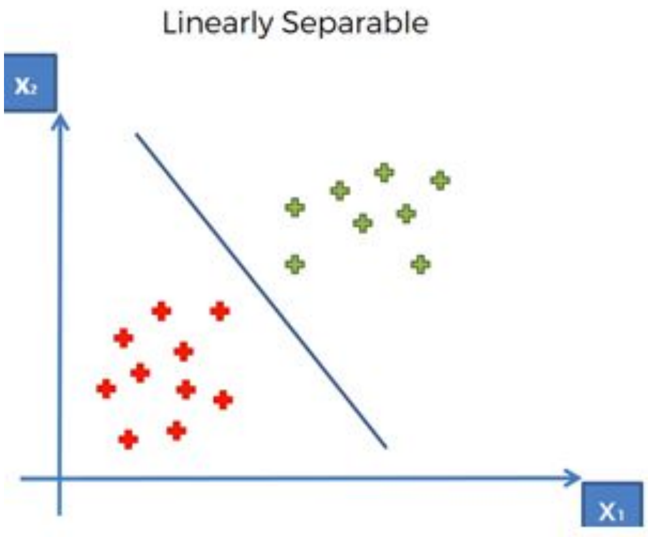
- Support Vector
- Hyperplane
- Margin Distance
- Linear Separable
- Non-Linear Separable

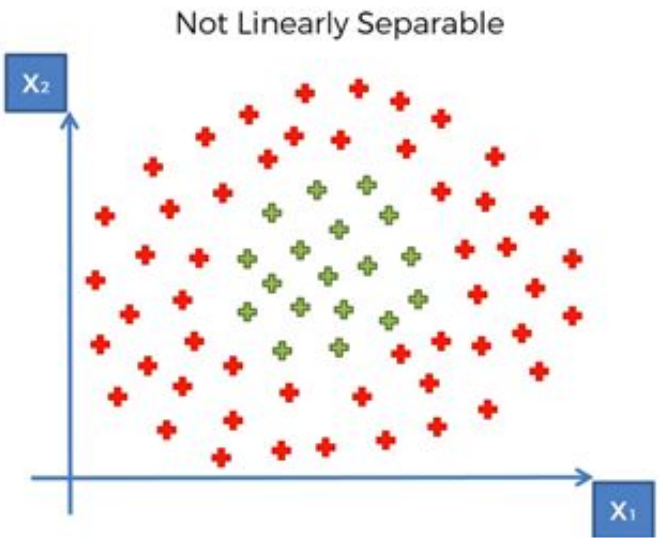



margin

Linear vs Non-Linear Separable

- **Linear**: Separated by straight line
- **Non-Linear:** Can’t separate by a straight line





Types of SVM

SVM can be of two types:

Linear SVM: SVM used for linearly separable data

- which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: SVM used for non-linearly separated data

- which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

SVM Hyperparameter

Kernel: Linear, RBF, Poly(default value is “rbf”).

- “rbf” and “poly” are useful for non-linear hyper-plane.
- Go for linear SVM kernel if you have a large number of features (>1000) because it is more likely that the data is linearly separable in high dimensional space.

Gamma: Used with Non-linear SVM

- Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.
- Higher the value of gamma, will try to exact fit as per training data set i.e. generalization error and cause over-fitting problem.

C: SVM regularization parameter-

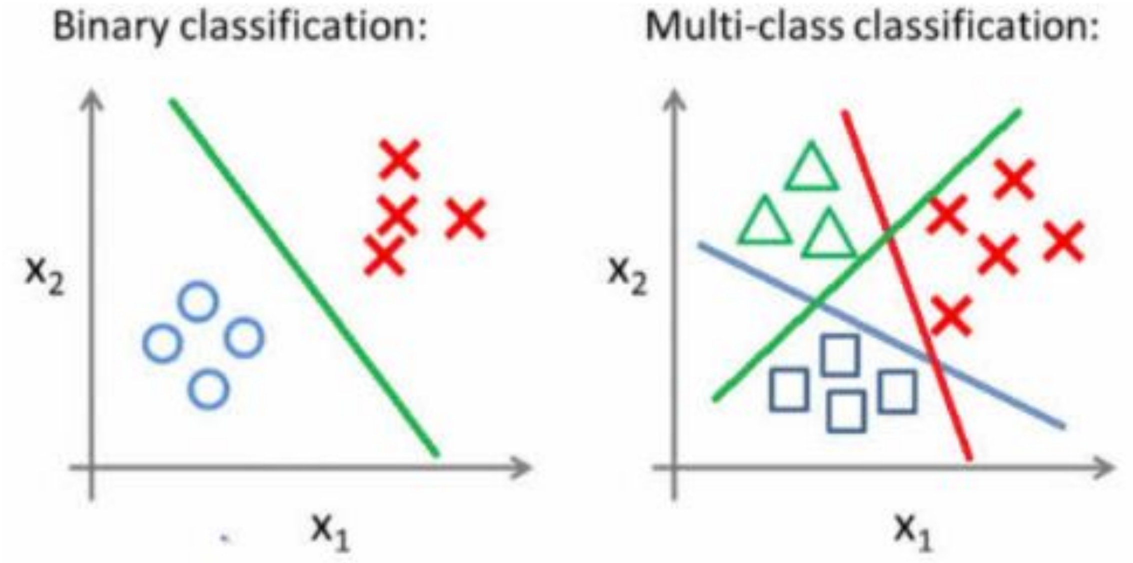
- Controls the trade-off between smooth decision boundaries and classifying the training points correctly.
- The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example

Multiclass Classification Using SVM

- SVM were originally designed to only support two-class-problems.
- SVM doesn't support multiclass classification. For multiclass classification, the same principle is utilized after breaking down the multi-classification problem into smaller subproblems, all of which are binary classification problems.

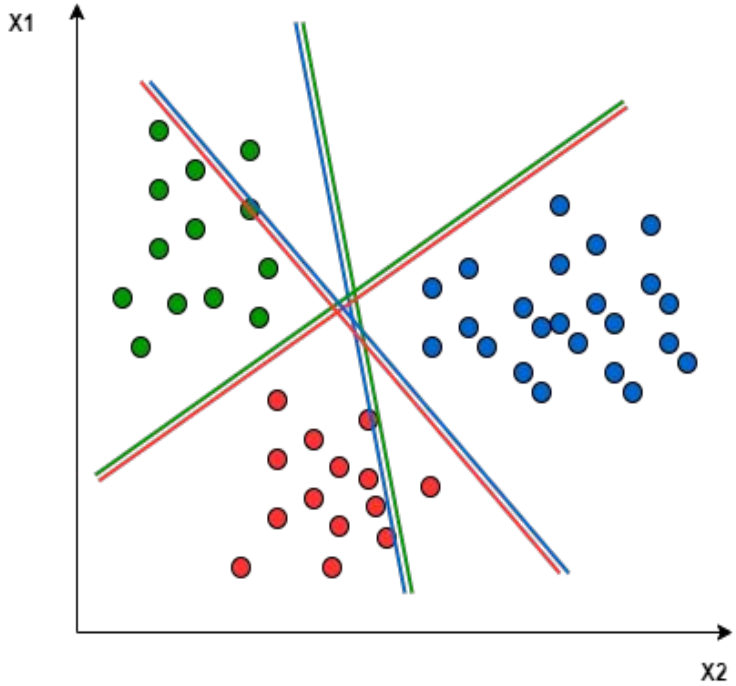
- The popular methods which are used to perform multi-classification on the problem statements using SVM are as follows:
  - One vs One (OVO) approach
  - One vs All (OVA) approach
  - Directed Acyclic Graph (DAG) approach

Binary vs Multiclass Classification



One vs One (OVO) approach

- In the One-to-One approach, we try to find the hyperplane that separates between every two classes, neglecting the points of the third class.
- The major problem with this approach is that we have to train too many SVMs.
- Each binary classifier predicts one class label. Classification decision is based on majority voting



One vs All (OVA)

In this technique, if we have N class problem, then we learn N SVMs:

SVM number -1 learns “class_output = 1” vs “class_output ≠ 1″

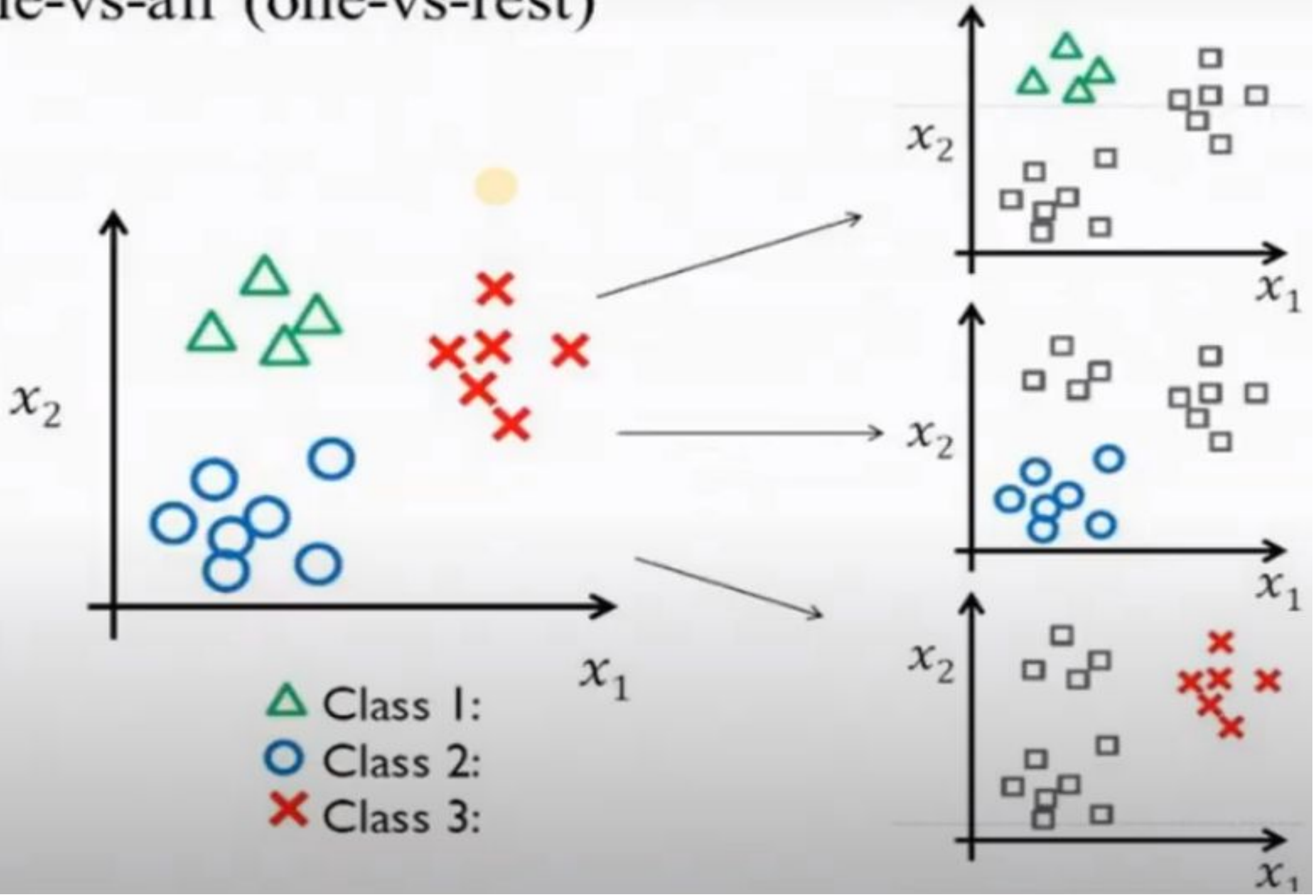
SVM number -2 learns “class_output = 2” vs “class_output ≠ 2″

:

SVM number -N learns “class_output = N” vs “class_output ≠ N”

- Classification decision is based on the probability of the new data sample belonging to a specific class

One-vs-all (one-vs-rest)



Challenges with SVM Multiclass classification

Yes, there are some challenges to train these N SVMs, which are:

1. Too much Computation: To implement the OVA strategy, we require more training points which increases our computation.

2. **N-class instances** then

- **One vs. One:** N* (N-1)/2 binary classifier models
- **One vs. All:** N binary classifier models

3. Problems becomes Unbalanced: Let's you are working on an MNIST dataset, in which there are 10 classes from 0 to 9 and if we have 1000 points per class, then for any one of the SVM having two classes, one class will have 9000 points and other will have only 1000 data points, so our problem becomes unbalanced.

SVM Pros & Cons

Pros:

- 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/features is greater than the number of samples. Works better on smaller and complex dataset (large features and lesser training data)

Cons:

- It doesn’t perform well when we have large data set(more samples, less features) 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