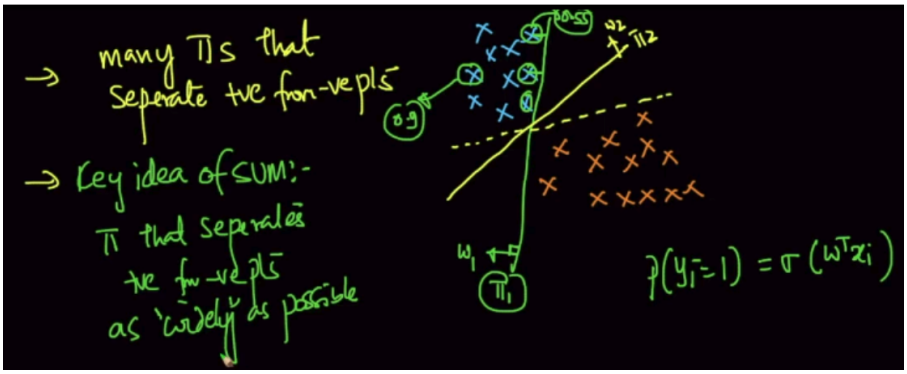
Support Vector Machine (SVM) -

Introduction -

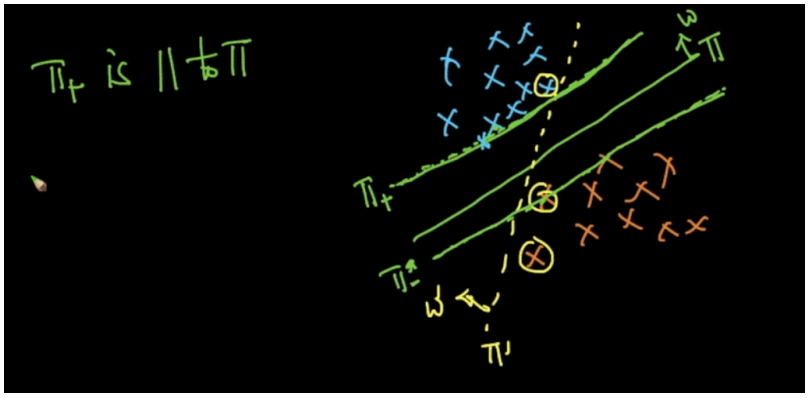
- SVM is a supervised machine learning algorithm
- It is used for both classification or regression challenges. Mostly used in classification problems.

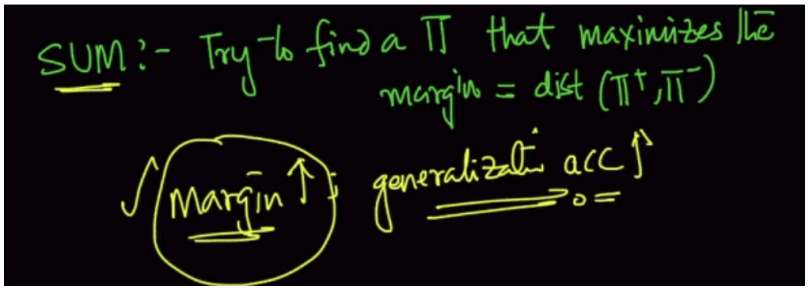
Define-

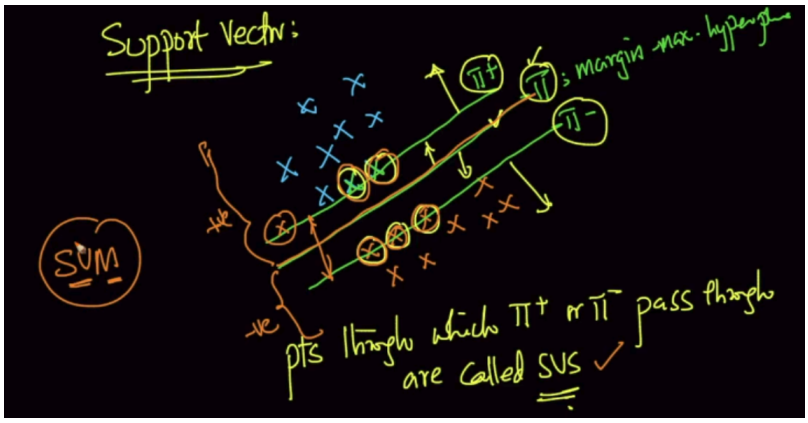
- **Key idea of SVM - π that separate +ve and -ve points as widely as possible.**
- If there are positive and negative clusters of points, the key idea of SVM is to find the plane that separates the two clusters as widely as possible. That hyperplane is called margin maximizing hyperplane.



- If we draw hyperplanes parallel to π which touches +ve and -ve points respectively, then the distance between the two parallel hyperplanes || to pi is called **margin.**
- Distance b/w π and π+ is equal to the distance b/w π and π-.

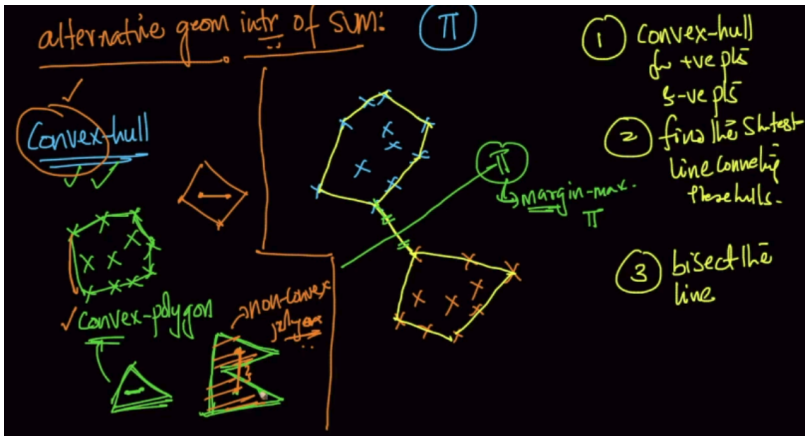






Alternate geometric intuition:

1. Create a convex hull for each class. Convex-hull is a polygon which is the smallest polygon for a set of points such that all the points lie either inside or on the edges of the polygon and the shortest path between any two points in the polygon always lie within the area of the polygon.
2. Find the shortest line connecting both the hulls.
3. The plane bisecting the hulls is the margin maximizing hyperplane.



Hard SVM -

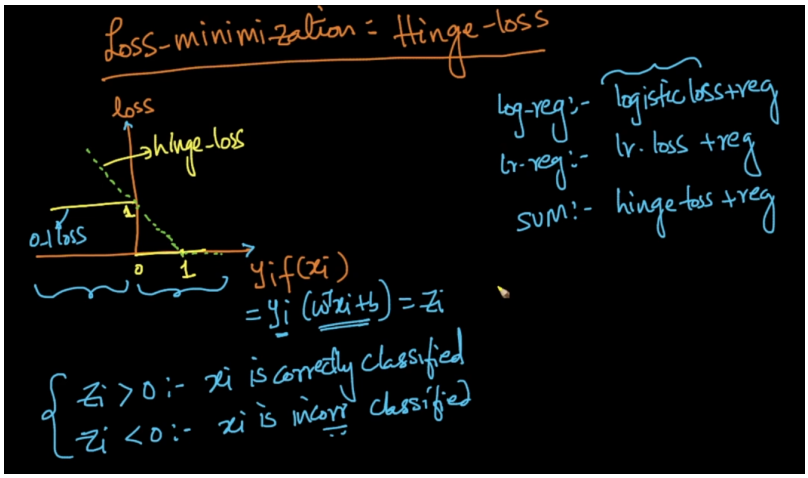
When positive and negative points are well separated.

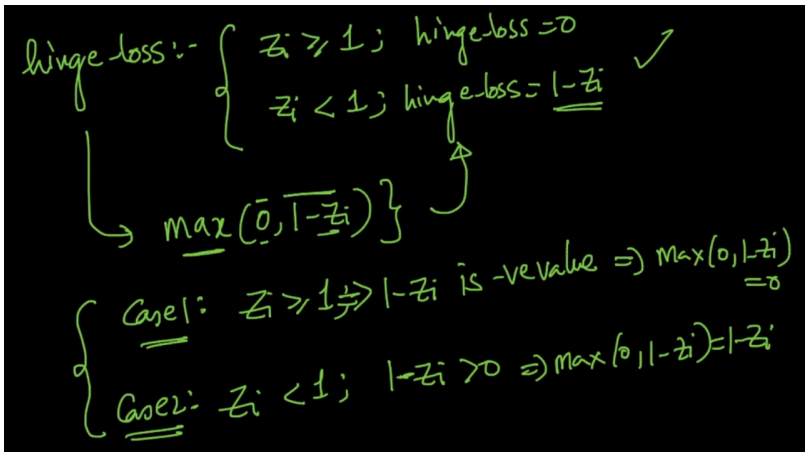
Soft SVM -

When positive and negative points are not well separated.

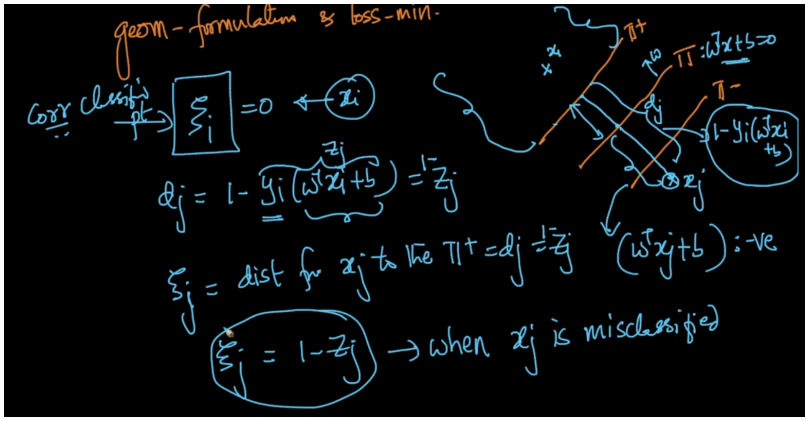
Loss Minimization- Hinge Loss

Hinge Loss- It is the approximation of the 0–1 loss minimization function which is not differentiable at 1.

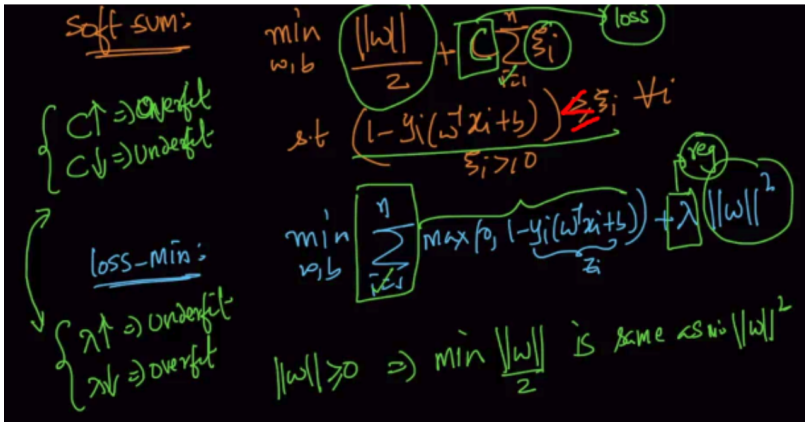




- When +ve points are present under pie- section.

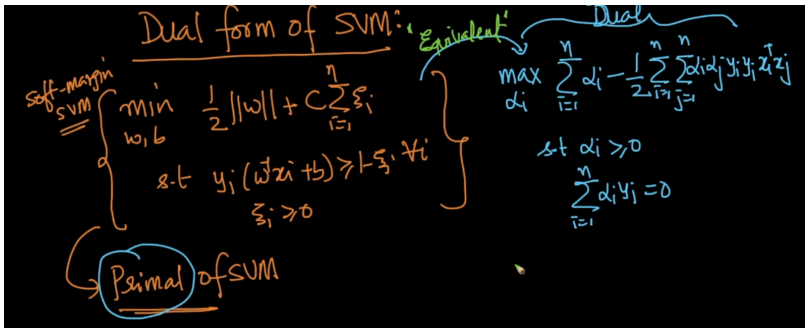


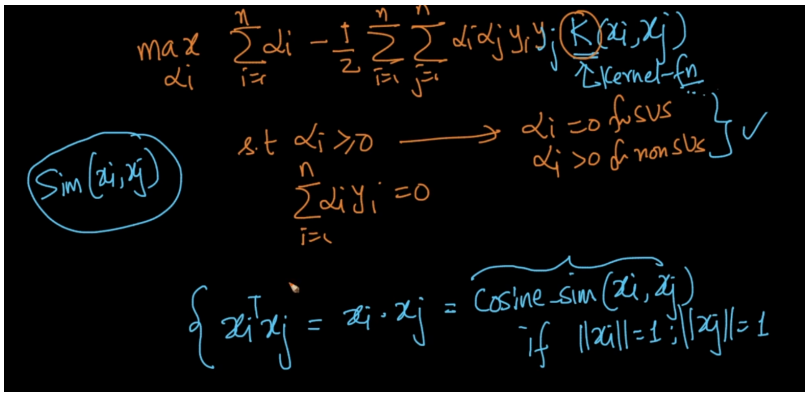
- Generally we multiply c with loss func.
- And λ with regularization.
- c = 1/λ or λ = 1/c
- Loss-minimization SVM = Soft SVM



Refer 9th point of this article- https://medium.com/@ashwanibhardwajcodevita16/from-zero-to-hero-in-depth-support-vector-machine-264931a1e135 (https://medium.com/@ashwanibhardwajcodevita16/from-zero-to-hero-in-depth-support-vector-machine-264931a1e135)

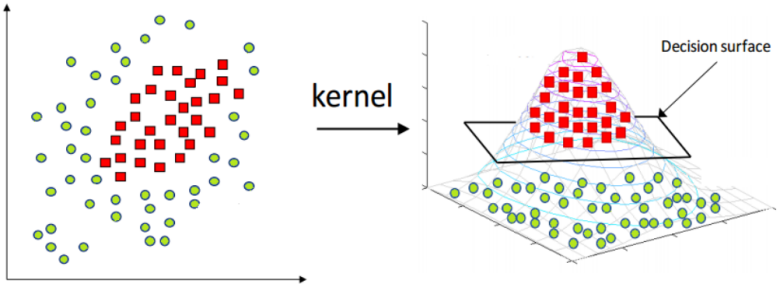
Dual Form of SVM





Kernel Trick or Kernelization -

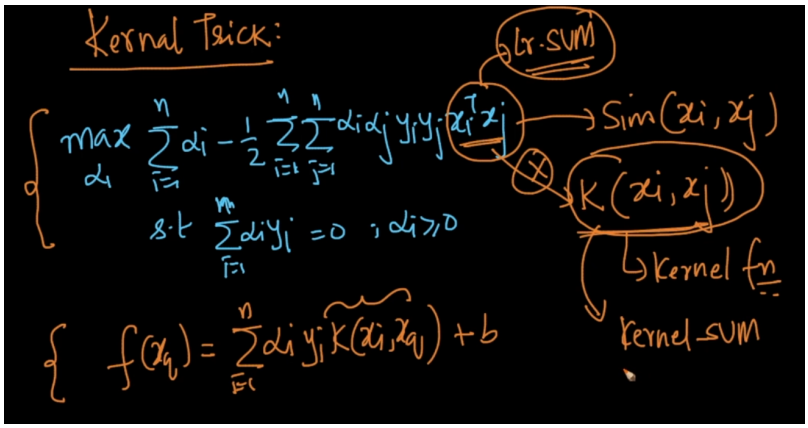
The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. **it converts not separable problem to separable problem.**

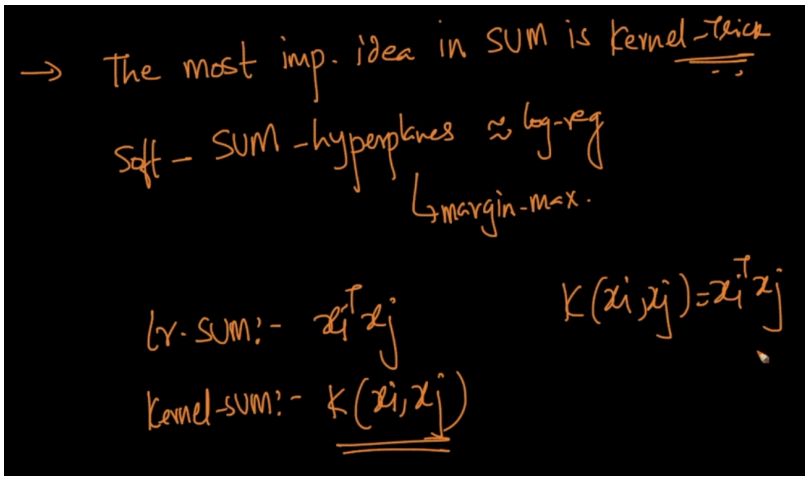


So, we are projecting the data with some extra features so that it can convert to a higher dimension space.

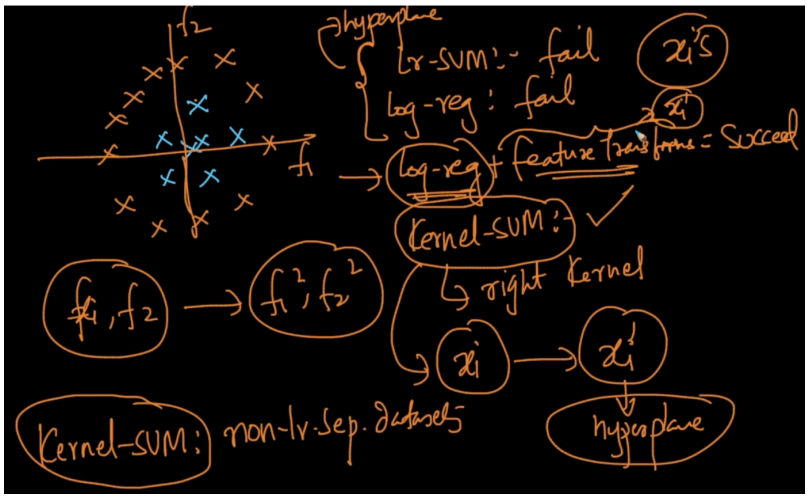
It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

Function kernel take data as input and transform it into the required form.





- For +ve and -ve two concentric datapoints, Linear SVM, Logistic regression Fail to separate them.
- Whereas Logistic regression with Feature Tranform and Kernel-SVM can able to classify them.
- Kernel-SVM can solve Non-linearly separable datasets also.



- Some famous kernels like polynomial kernel, Radial basis Kernel etc.

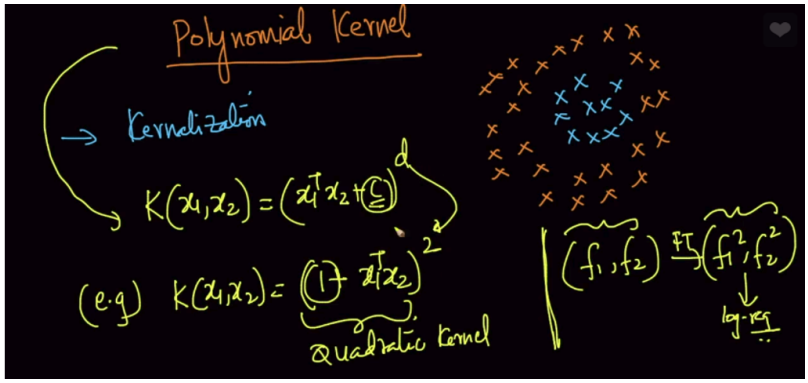
Polynomial Kernel

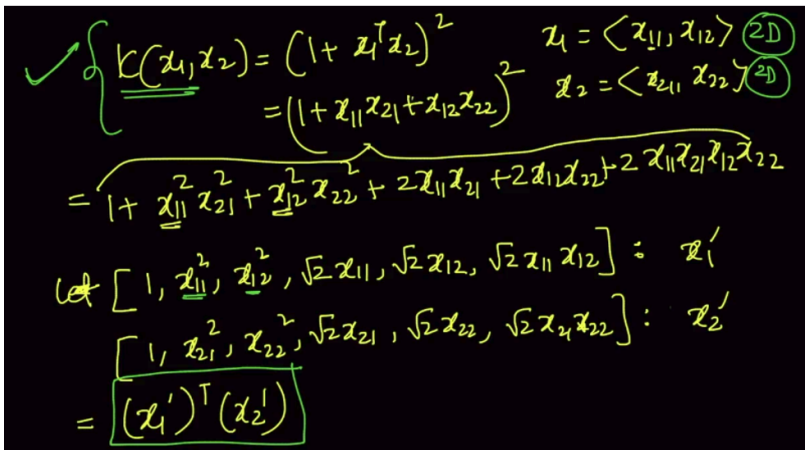
In general, the polynomial kernel is defined as -

$$K(X_1, X_2) = (a + X_1^T X_2)^b$$

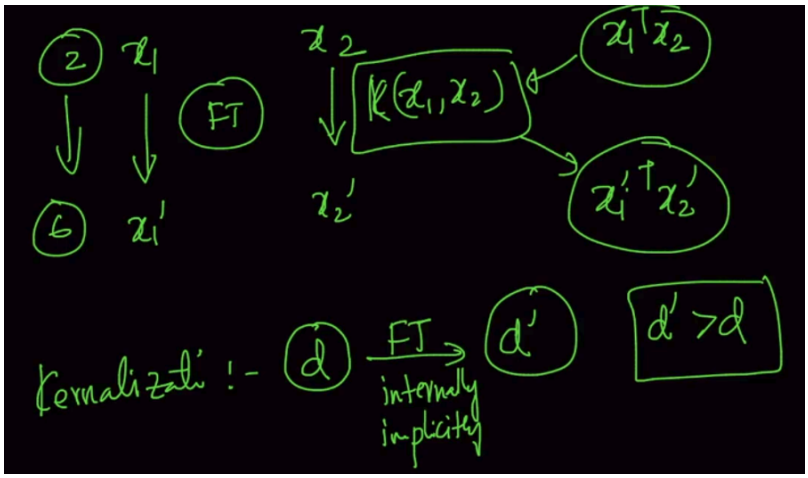
where a = constant term and b = degree of kernel.

in the polynomial kernel, we simply calculate the dot product by increasing the power of the kernel.

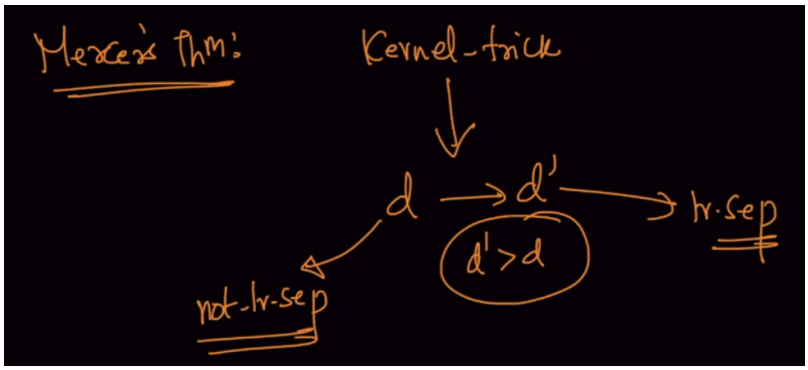




- Kernelization to the feature transform internally using the kernel trick.



- The data will become separable after kernel trick. After applying mercers theorem the data will be more linearly separable transforming to d' dimensional space.



- Kernel trick does the transformation internally than that of the feature transform that is done externally in logistic regression.

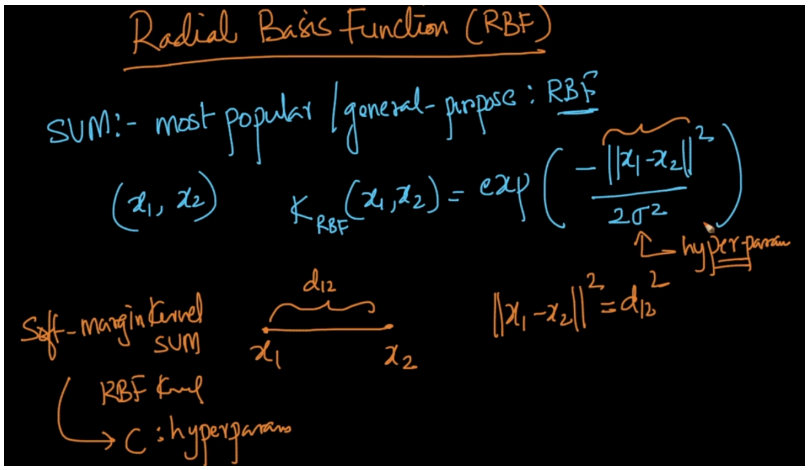
Ques-What is a right Kernel to apply?

Ans-The problem of explicit feature transform is changed to find the right kernel in SVM than that of Logistic regression.

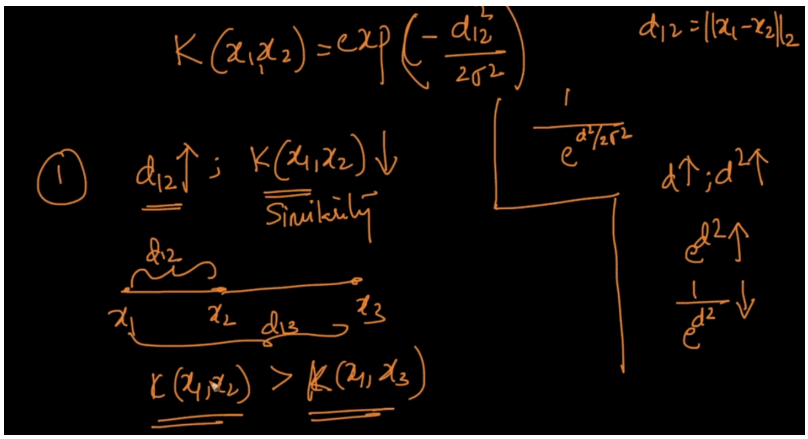
Radial Basis Function(RBF) Kernel

The RBF kernel function for two points X_1 and X_2 computes the similarity or how close they are to each other.

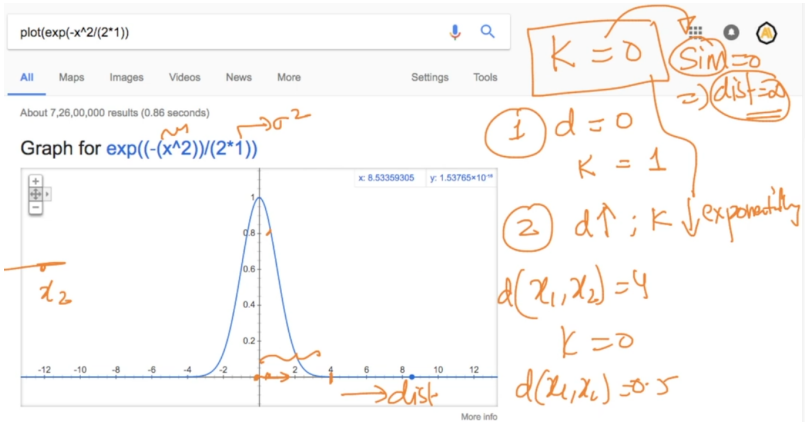
- In SVM the most popular and general purpose kernel in RBF Kernel.
- As distance between the points increases then the kernel decreases.
- Here sigma is the hyper parameter in RBF



- Kernel value of k(x1,x2) is greater than kernel value of k(x1,x3).



- Here d = distance b/w two points, k = kernel.
- If points are farther away then kernel value is 0, and Kernel value is same as similarity so as kernel = 0 which means similarity = 0 it means distance equal to infinity.



- If we dont know the Best kernel to use then simply use RBF kernel.
- Kernerlization = Feature Tranformation.

Domain Specific Kernels-

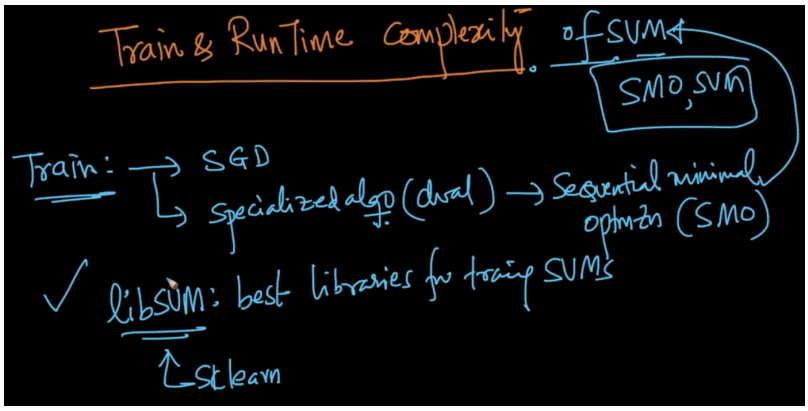
There area lot of specify kernels,

- String kernels – It is design for text classification.

- Geonmic kernels
- Graph kernels

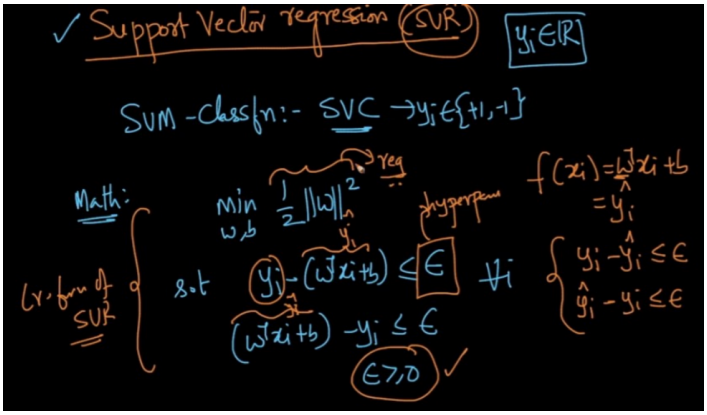
Train and Run time complexities of SVM.

- Libsvm is best library for training SVMs other one is sklearn.



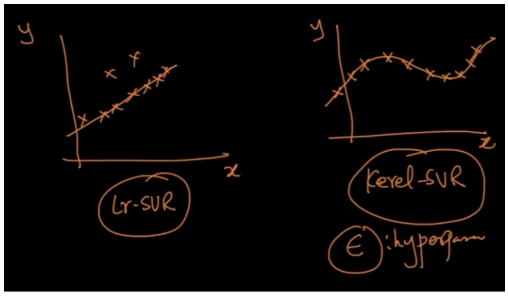
SVM Regression

- When SVM is used for classification then its also called Support Vector Classification(SVC).
- When SVM is used for regression then its also called as Support Vector Regression(SVR).



Here, we obtain the difference between the yi and yi^ and made to less than or equal to epsilon, where epsilon is a hyper parameter.

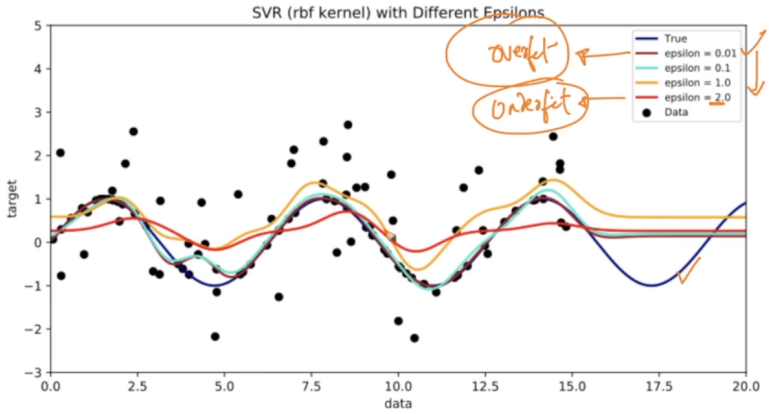
Where the ||w||^2 can be kernalized.



- If epsilon is low then the error are low on the training data, over fitting will increase.

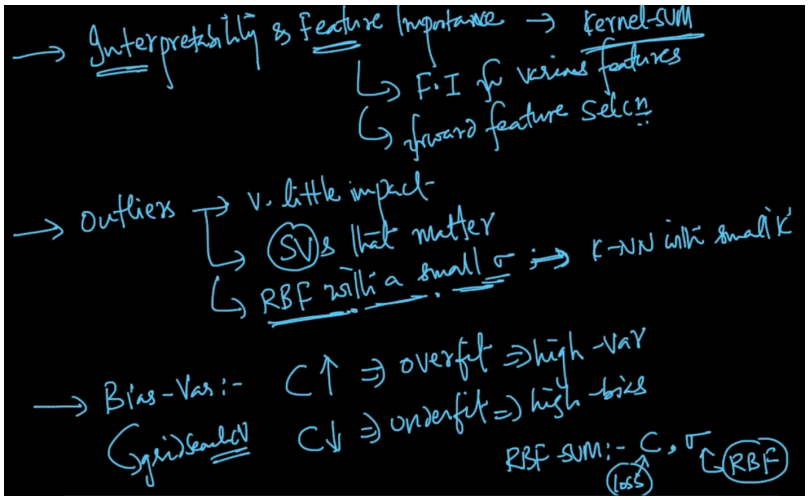
- If epsilon in high then errors on training data will increase, that means under fit the data.

RBS represents the KNN-reg roughly.

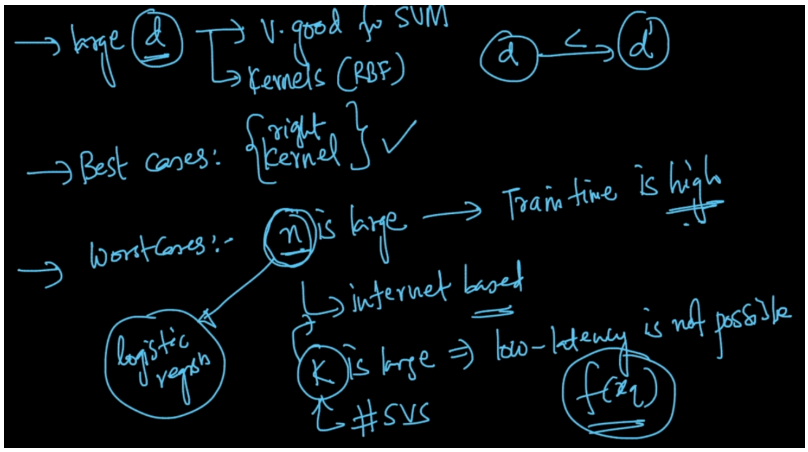


Cases in SVM-

- SVM's can work with similarity matrix.
- Interpretability & Feature Importance: For kernel SVM's we cannot get the feature importance, but we have the Forward Feature Selection process.



- On large dimension SVM work fine with kernel.
- Best Case for SVM when its have right kernel.
- Worst Case is -
  - When training size data(n) is large bcz it will have high training time.
  - When no. of Support Vector are large it means we can not have low latency.
- When we have large amount of data then we use Logistic Regression with Feature Transform and not use SVM.

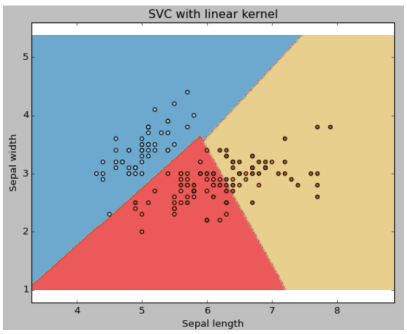


How to tune Parameters of SVM?

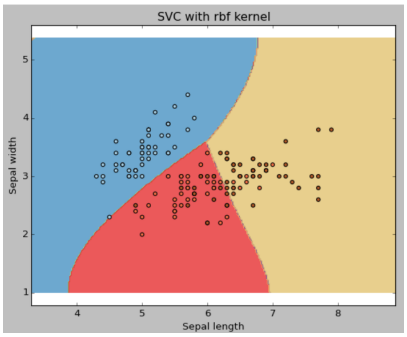
Tuning the parameters values for machine learning algorithms effectively improves model performance. Let's look at the list of parameters available with SVM.

We are going to discuss about some important parameters having higher impact on model performance, "kernel", "gamma" and "C".

Kernel- We have already discussed about it. Here, we have various options available with kernel like, "linear", "rbf","poly" and others (default value is "rbf"). Here "rbf" and "poly" are useful for non-linear hyper-plane.

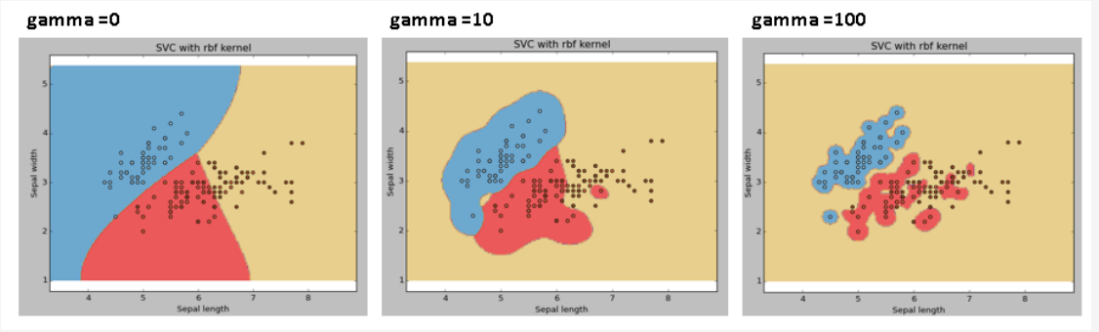


We should use linear SVM kernel if we have a large number of features (>1000) because it is more likely that the data is linearly separable in high dimensional space. Also, we can use RBF but do not forget to cross-validate for its parameters to avoid over-fitting.

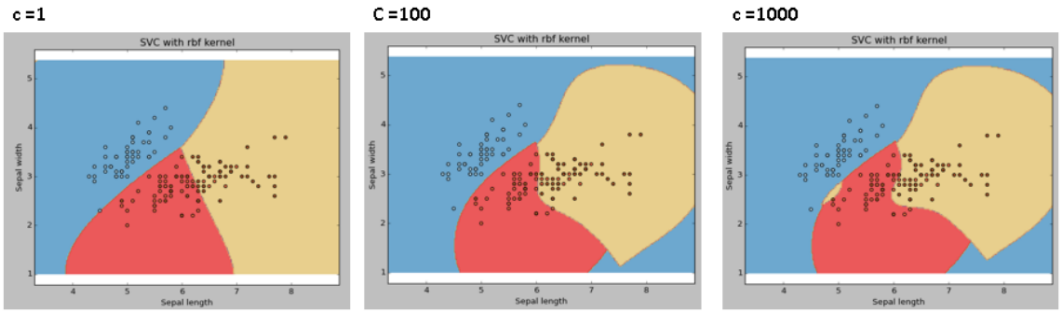


gamma- Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. Higher the value of gamma, will try to exact fit the as per training data set i.e. generalization error and cause over-fitting problem.

Eg svc = svm.SVC(kernel='rbf', C=1,gamma=0).fit(X, y)



C- Penalty parameter C of the error term. It also controls the trade-off between smooth decision boundaries and classifying the training points correctly.



Pros and Cons associated with SVM

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 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.

Cons-

- 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.