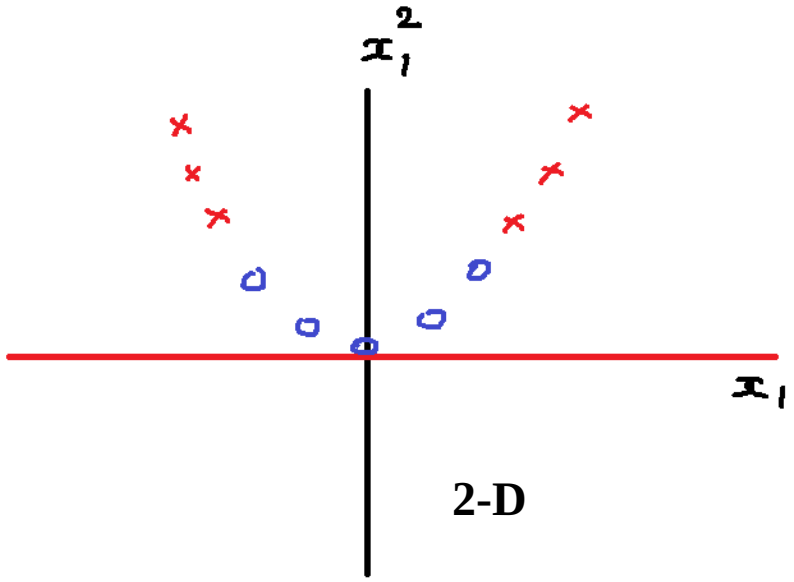
SVM Kernel Function, Multiclass Classification, Support Vector Regression

Dr. Kuppusamy .P
Associate Professor / SCOPE

Kernel Trick

- If data is One dimension and linearly arranged, then data can be separated using a point.
- If data is two dimension and linearly arranged, then data can be separated using a straight line.
- If data is three dimension and linearly arranged, then data can be separated using a hyperplane.
- If dataset is two dimension and non-linear, cannot separate using a single straight line or plane.
- Need to transform the lower dimension (e.g., 2-D) features into higher dimension (e.g., 3-D) features by applying suitable function to the dataset.
- Kernel trick is non-linear function that transforms the data from lower dimension into higher dimension.
- The transformed datapoints can be linearly separable.

1-D



2-D

Kernel Trick- Feature Transformation from Lower Dimension to Higher Dimension

Example 1

- Given dataset is

Feature x_1
4
1
-1
0
2
3
-3
-2

Transformation

$$f(x_1) = \begin{pmatrix} x_1 \\ x_1^2 \end{pmatrix}$$

Feature x_1	**Feature** x_1^2
4	16
1	1
-1	1
0	0
2	4
3	9
-3	9
-2	4

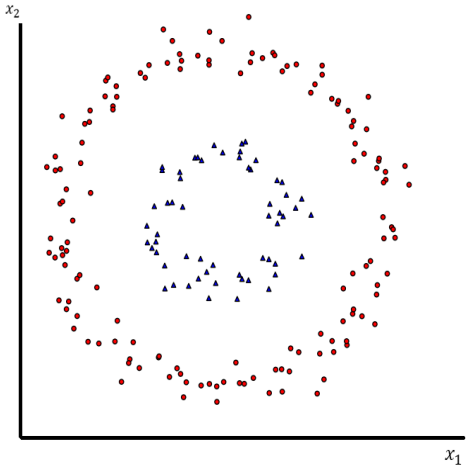
- Apply the suitable 2-D non-linear function is applied to transform this 1-D dataset into 2-D dataset

$$f(x_1) = \begin{pmatrix} x_1 \\ x_1^2 \end{pmatrix}$$

- This can be classified using the straight line.

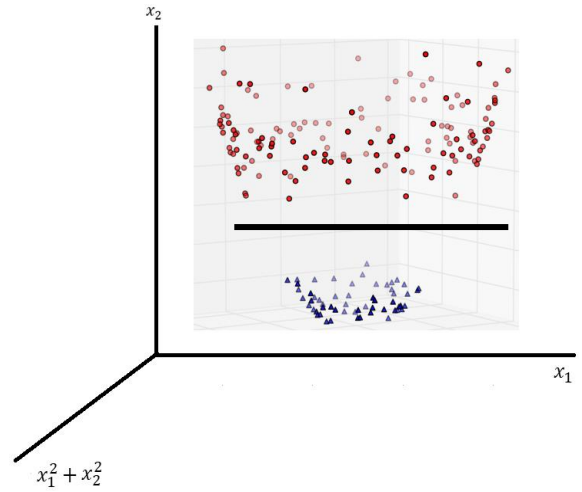
Kernel Trick- Example 2

- If dataset is two dimension and non-linear, cannot separate using a single straight line or plane.
- Need to transform the lower dimension (e.g., 2-D) features into higher dimension (e.g., 3-D) features by applying suitable function to the dataset.

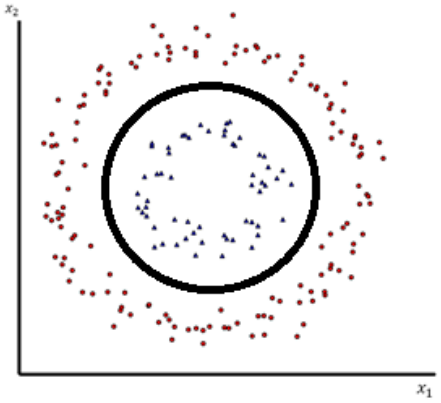


- Apply the suitable function to transform this 2-D dataset into 3-D dataset.
- The following 3-D function is applied on 2-D data for transformation

- $f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1^2 + x_2^2 \\ x_2^2 \end{pmatrix}$



- Since plot contains 3-d Space, it appears like a plane parallel to the x-axis.
- If convert 3-d space into 2d space with $x_1^2 + x_2^2$=1, then it appears as in the given plot.
- Hence, we get a circumference of radius 1 in case of non-linear data.



Kernel Function – Math Intuition

- Dual Problem $D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} \alpha_i \alpha_k \left[y^{(i)} y^{(k)} * x^{(i)^T} x^{(k)} \right]$
- $x^{(i)^T} x^{(k)}$ is dot product of features

Example

- Assume first sample is ‘a’ and second sample is ‘b’.

Feature x_1	**Feature** x_2
1	5
2	6

- The non-linear function $f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{pmatrix}$
- $f(a) = \begin{pmatrix} a_1^2 \\ \sqrt{2} a_1 a_2 \\ a_2^2 \end{pmatrix}$ and $f(b) = \begin{pmatrix} b_1^2 \\ \sqrt{2} b_1 b_2 \\ b_2^2 \end{pmatrix}$

Kernel Function – Math Intuition

- $f(a).f(b) = \begin{pmatrix} a_1^2 \\ \sqrt{2}a_1a_2 \\ a_2^2 \end{pmatrix} . \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_2^2 \end{pmatrix}$

$= a_1^2\ b_1^2 + 2\ a_1b_1a_2b_2 + a_2^2\ b_2^2$

$= (a_1b_1 + a_2\ b_2)^2$

$= \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} . \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2$

$f(a).f(b) = (a.b)^2$

- Kernel function takes input as vector in original space and returns as dot product of vectors in transformed space.
- $K(a,b) = (a.b)^2$

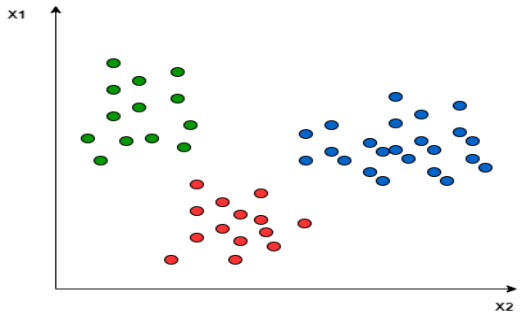
Kernel function Property:

Transformed data Dot product values can be written in the form of actual data dot product values

Polynomial Kernel $K(x_1x_2) = (x_1\ .x_2 + c)^d$ c –constant d- degree (any value)

RBF Kernel function : $K(x_1x_2) = \exp\left(\frac{-\|x_1 - x_2\|^2}{2\sigma^2}\right)$

Multiclass Classification Using SVM



- Initially, the multi-classification problem is breaking down into smaller subproblems.
- Now, subproblems appear as binary classification problems.
- Then, apply binary classification principle for classifying all smaller subproblems.
- The majority voting or highest value (farthest into the positive region) of subproblems output will be the predicted result.

Two approaches are considered in SVM multi-class classification:

1. One vs One (OVO) approach
2. One vs Rest (OVR) approach

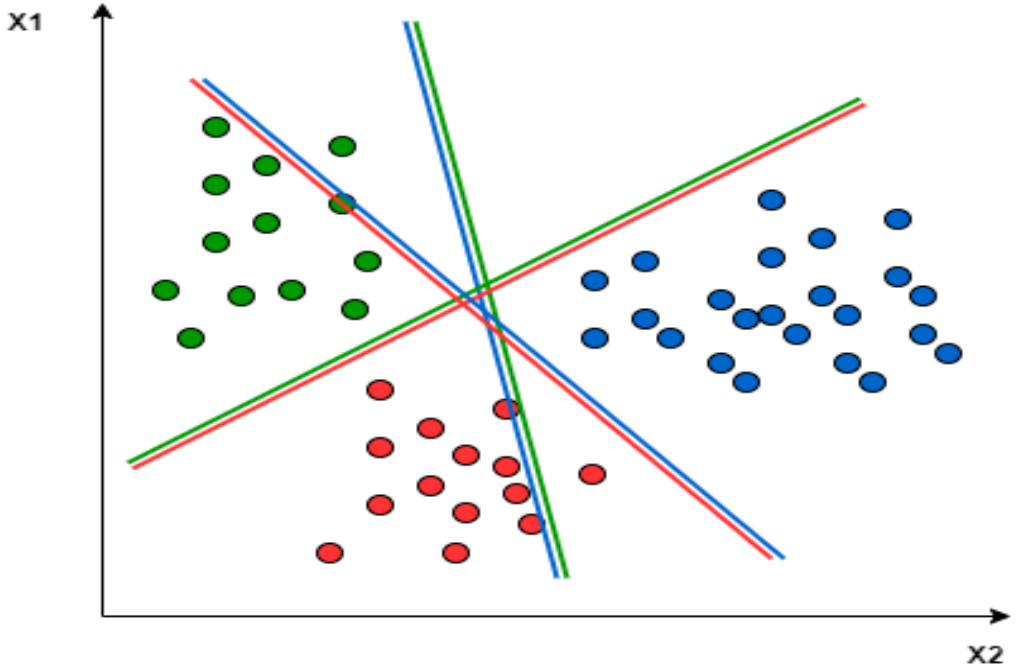
One vs One (OVO) approach:

- It uses **majority voting** of all the subproblem classifiers along with the distance from the margin as its confidence criterion.

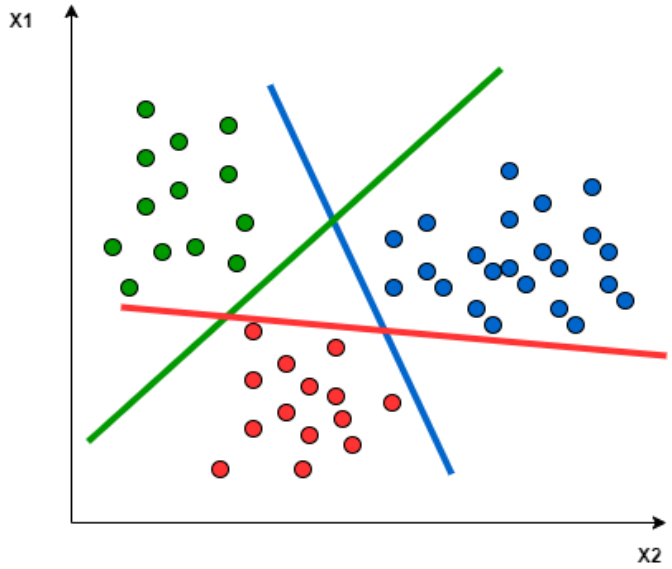
clf = svm.SVC(decision_function_shape='ovo')

OVO Limitations: Need to train too many SVM classifiers.

- Let's consider three class classification problem: Green, Red, and Blue.
- In One-to-One approach, computes the hyperplane that separates between every two classes, neglecting the points of the third class.
- E.g., Red-Blue line tries to maximize the separation only between blue and red points while it doesn't do anything with the green points.



Multiclass Classification Using SVM



One vs Rest (OVR)

Clf = svm.SVC(decision_function_shape= "ovr“)

- In OVR, assume N class problem, then need of N SVM classifiers as follows:

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

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

  :

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

Prediction for new test input

- Predict with each of the SVM classifiers and then find highest (farthest into the positive region) value that behaves as a confidence criterion for a particular SVM.

Challenges to train N SVM classifiers:

1. High Computation: Need of more training points that increases the computation.
2. Problems becomes Unbalanced: Assume dataset contains 4 classes. In this, Each class contains 250 data points, then any one of the SVM having two classes. One class will contain 750 points and other class will have only 250 data points. So, this dataset for this problem becomes unbalanced.

Multiclass Classification Using SVM

Address the unbalanced problem:

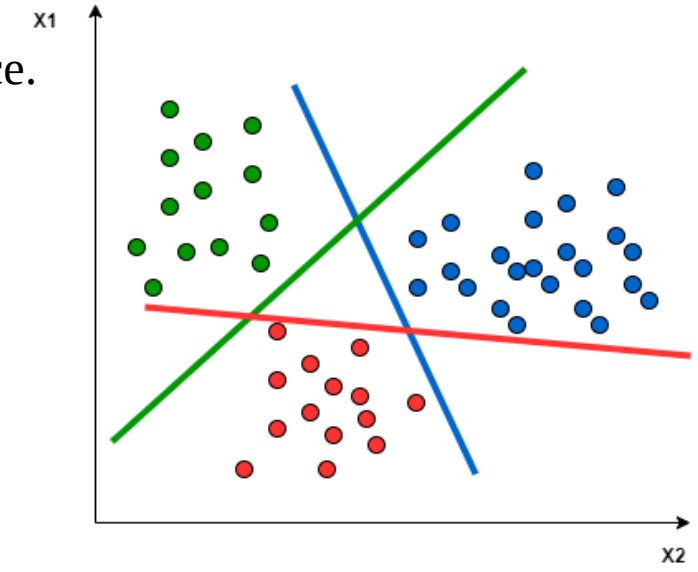
- Collect some representative (subsample) from the class that contains more training samples i.e., majority class.

Its processed as follows:

- *Use the 3-sigma rule of the normal distribution*: Fit data to a normal distribution and then subsampled accordingly so that class distribution is maintained. Pick some data points randomly from the majority class.
- *Apply SMOTE subsampling technique*.

- **One vs Rest** approach computes a hyperplane to separate the classes i.e., the separation takes all data points into account and then segregates into two groups. One group for the one class of data points and the other group for all other points.

E.g., Greenline tries to maximize the gap between green points and all other points at once.



Difference OVO and OVR

- One vs Rest approach, the classifier can use n SVMs. n – no of classes
- One vs One approach, the classifier can use n(n-1)/2 SVMs.

- E.g., Dataset contains 4 Classes (Red, Green, Blue, Orange)
- It needs 6 Classifiers such as (Red, Blue), (Red, Green), (Red, Orange), (Blue, Green), (Blue, Orange), (Green, Orange)

Types of parameters

Kernel Parameters

- The kernel parameters determine the shape and structure of the hyperplane. Four types are **linear, RBF, polynomial and sigmoid**. By tuning, can set the kernel type and the model adapts to it.

Gamma Parameters

- The gamma parameter is for **non-linear** hyperplanes. Here, the kernel value is RBF, polynomial or sigmoid. This parameter tries to fit in all the data.

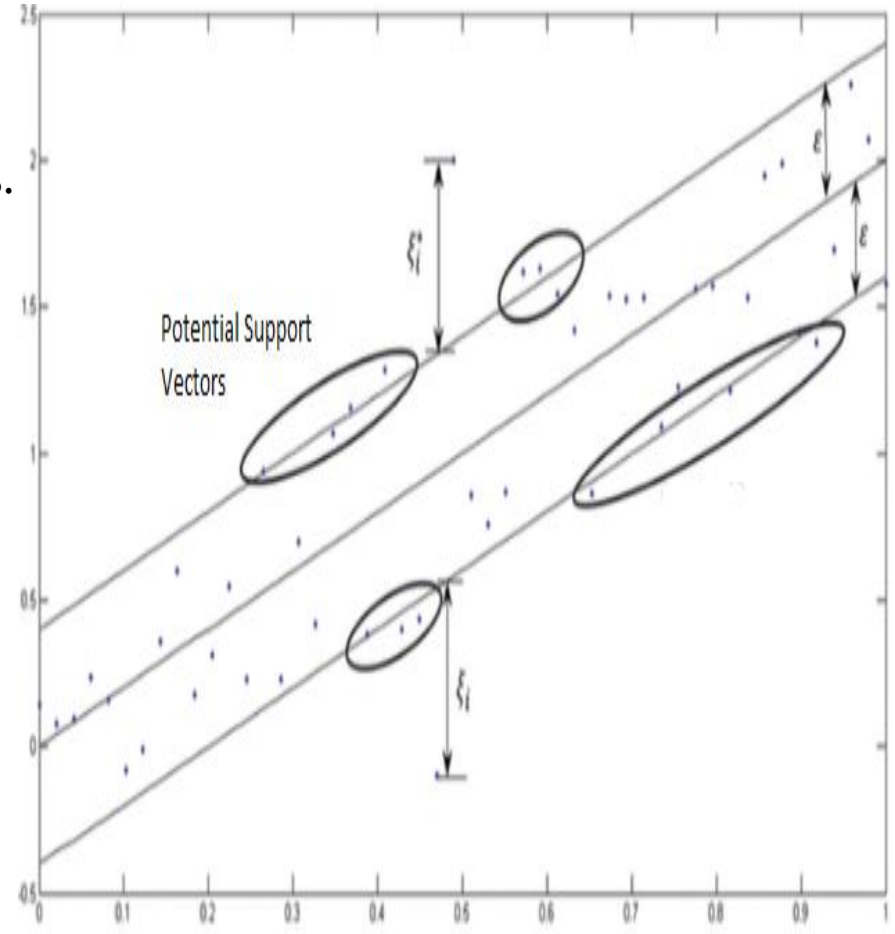
C Parameters

- The C parameter is more like a **penalty parameter**. If the value of C is high then, the datapoints further away from the plane. This makes us include them also and this leads to overfitting.

Degree Parameters

- The degree parameter uses **‘poly’** as the kernel value. The degree parameter helps to find the hyperplane for splitting data. The more the degree of polynomial increases, the more training time it takes. By tuning all these values, one can control the hyperplane as well as the model of SVM.

Support Vector Regression (SVR)



- Support Vector Regression is used for linear and non-linear regression problems.
- SVR finds a function that approximates mapping from an input domain to real numbers based on a training sample.
- SVR fits as many samples as possible between the lines while limiting the margin violations denoted as epsilon.

  clf = SVR(kernel='rbf')
- SVR uses linear, polynomial and RBF kernels.

Steps of SVR for Practice:

- Collect a training samples
- Choose a kernel and parameter and regularization if needed.
- Apply feature scaling
- Form a correlation matrix $K_{i,j} = \exp\left(\sum_k \theta_k \left|x_k^i - x_k^j\right|^2\right) + \epsilon\delta_{i,j}$
- Train the model to get contraction coefficient. $\vec{\alpha} = \bar{K}^{-1}\vec{y}$
- Use this coefficient to create an estimator.

To estimate the y^* value for a test $\vec{x^*}$ point, compute the correlation vector $\vec{k}$,

$$y^* = \vec{\alpha} \cdot \vec{k} \qquad k_i = \exp\left(\sum_k \theta_k |x_k^i - x_k^\star|^2\right)$$

- Predict a new data point

SVM Pros and Cons

Pros

- SVMs offers better results than ANNs.
- SVM handles higher dimensional and linearly inseparable data efficiently. They are quite memory efficient.
- Complex problems can be solved using kernel functions in the SVM.
- SVM works well with all three (structured, semi-structured and unstructured) types of data.
- Over-fitting is a problem avoided using regularization parameters.

Cons

- Choosing a good kernel function is not an easy task.
- SVM consumes more time for training with large datasets.
- If dataset contains more features than samples, the model will give a poor performance.

VIT-AP UNIVERSITY

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.

2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.

3. Wikipedia

4. https://www.svm-tutorial.com/2016/09/duality-lagrange-multipliers/