Data Classification using Support Vector Machine

Arun Kumar and Pankaj Kumar Saini
(MIT2020116) (MIT2020117)

M.Tech Information Technology
Indian Institute of Information Technology Allahabad

***Abstract*—Classification is one of the most important tasks for different application such as content order, tone acknowledgment, picture characterization, miniature cluster quality articulation, proteins structure forecasts etc. Most of the existing supervised classification methods are based on a large portion of the current directed grouping techniques depend on customary measurements, which can give ideal outcomes when test size is watching out for vastness. This paper contains details about Support Vector Machine (SVM). In our experiments, the support vectors, which are critical for classification, are obtained by learning from the training samples.In this paper we have shown the comparative results using different kernel functions for all data samples. from the training samples.**

***Index Terms*—Support Vector Machine,Classification,Kernel function**

I. Introduction

The Support Vector Machine (SVM) was first proposed by Vapnik and has since pulled in a high level of revenue in the AI research local area [2]. A few late investigations have revealed that the SVM (support vector machines) for the most part are equipped for conveying higher execution regarding characterization exactness than the other information characterization calculations. Sims have been utilized in a wide scope of genuine world issues like content arrangement, transcribed digit acknowledgment, tone acknowledgment, picture arrangement and item identification, miniature cluster quality articulation information examination, information order. It has been shown that Sims is reliably better than other regulated learning strategies. Some datasets, the exhibition of SVM is very delicate to how the expense boundary and bit boundaries are set.

This process is commonly referred to as model selection. One practical issue with model selection is that this process is very time consuming. We have experimented with a number of parameters associated with the use of the SVM algorithm that can impact the results. These parameters include choice of kernel functions: linear kernel, polynomial kernel and RBF kernel.

II. Dataset

In this report we are discussing about Support Vector Machine(SVM). For better understanding we are using real life dataset : Social Network Ads.

Social Network Ads impact daily life. Before the finish of 2020, 98 percent of shoppers worldwide said they'd visited an interpersonal organization in the previous month. In the interim, worldwide web-based media promotion spend has hopped 50 percent in a solitary year. With advertisements soaking our feeds, standing apart is critical.

What age group effect more, what gender, what estimated salary purchased more items two prediction (1/0) given in dataset: 1 (yes) and 0 (No).

III. SVM overview

Support Vector Machine:

SVMs are set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classification. A special property of SVM is , SVM simultaneously minimize the empirical classification error and maximize the geometric margin. So SVM called Maximum Margin Classifiers.

IV. KERNEL SELECTION OF SVM

Training vectors xi are mapped into a higher (may be infinite) dimensional space by the function phi . Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimension space . c is greater than 0 is the penality parameter of the error term.

Furthermore, is called You see kernel function[2]. There are many kernel functions in SVM, so how to select a good kernel function is also a research issue.However, for general purposes, there are some popular kernel functions

Here, gama, r and d are kernel parameters. In these popular kernel functions, RBF is the main kernel function because of following reasons [2]:

1. The RBF kernel nonlinearly maps samples into a higher dimensional space unlike to linear kernel.
2. The RBF kernel has less hyperparameters than the polynomial kernel.
3. The RBF kernel has less numerical difficulties.

k(xi,xj)=$\phi(xi)^T\phi(xj) : Kernelfunction(2)$

$*$ **Linearkernel** :
$K(xi, xj) = xi^T.xj.$
$*$ **Polynomialkernel** :
$K(xi, xj) = (\gamma.xi^T.xj + r)^d, \gamma > 0$
$*$ **RBFkernel** :
$K(xi, xj) = exp(-\gamma \left\|xi - xj\right\|^2), \gamma > 0$

V. MODEL SELECTION OF SVM

Model selection is also an important issue in SVM. Recently, SVM have shown good performance in data classification. Its success depends on the tuning of several parameters which affect the generalization error. We often call this parameter tuning procedure as the model selection. If you use the linear SVM, you only need to tune the cost parameter C. Unfortunately , linear SVM are often applied to linearly separable problems.

Many problems are non-linearly separable. For example, Satellite data and Shuttle data are not linearly separable. Therefore, we often apply nonlinear kernel to solve classification problems, so we need to select the cost parameter (C) and kernel parameters (gamma, d) We usually use the grid-search method in cross validation to select the best parameter set. Then apply this parameter set to the training dataset and then get the classifier. After that, use the classifier to classify the testing dataset to get the generalization accuracy.

ANALYSIS

SVM uses kernel to find a hyper-plane to identify the class of the data point. we have used 75% data for training and 25% for testing.

A. Linear Kernel

Linear Kernel is using optimisation problem for a linear kernel is much faster.Use linear kernel when number of features is larger than number of observations.Therefore, Linear Kernel is the easiest and can achieve decent accuracy.

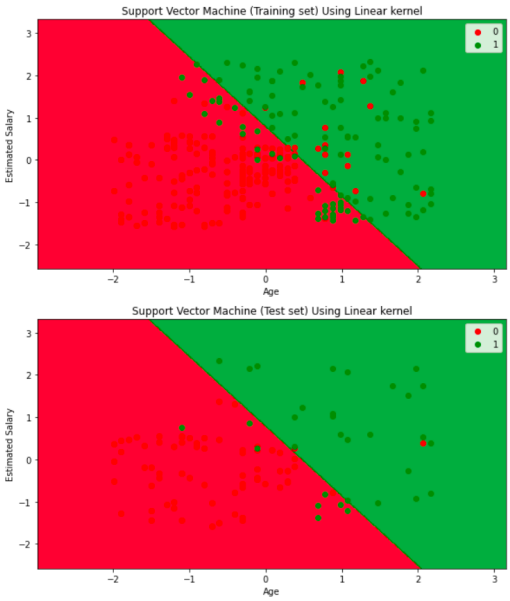

Fig. 1. figure shows performance of linear kernel with no regularizer on training and test dataset.

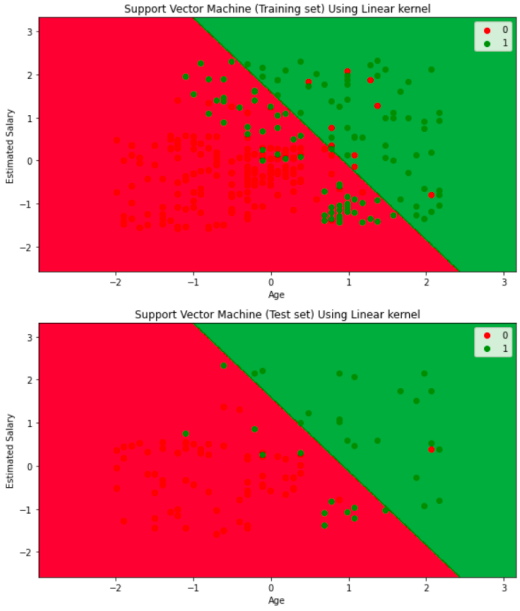

Fig. 2. figure shows performance of linear kernel with regularizer value 0.01 on training and test dataset.

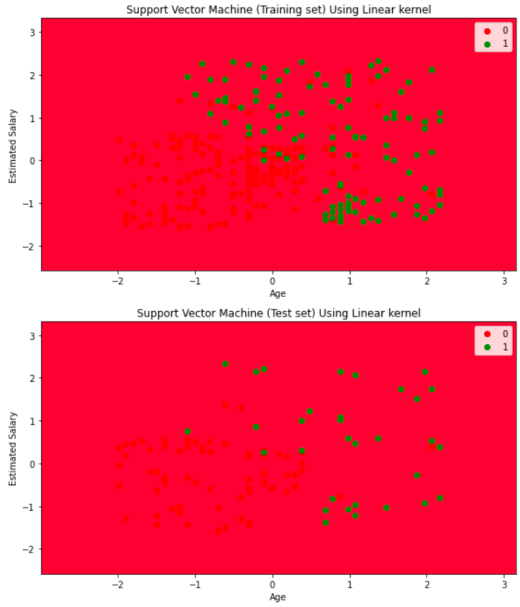

Fig. 3. figure shows performance of linear kernel with regularizer value 0.001 on training and test dataset.

after experimenting linear kernel with 3 different regularizer value we get highest accuracy with no regularizer with 90% accuracy than with 0.01 of 87% and last with 0.001 of 68% accuracy.

So we can conclude that with no regularizer we get highest accuracy in linear kernel.

B. *Polynomial Kernel*

Polynomial Kernel uses degrees to find a plane and than classify data points.Polynomial kernels are less time consuming and provides less accuracy than the rbf kernel.

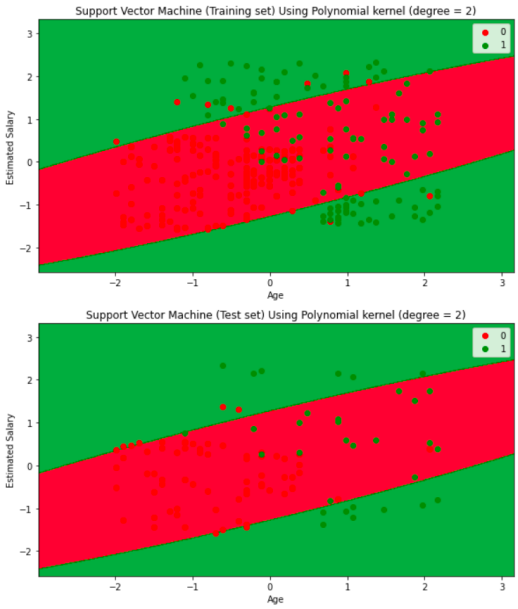

Fig. 4. figure shows performance of polynomial kernel on training and test dataset with degree 2.

using polynomial kernel with degree 2 model get an accuracy of 74%.

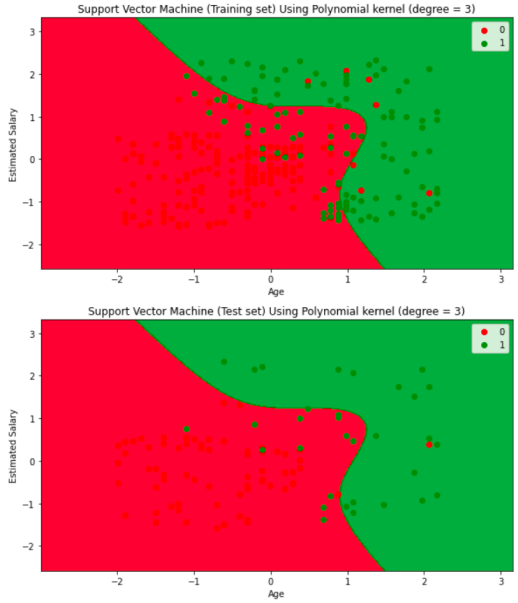

Fig. 5. figure shows performance of polynomial kernel on training and test dataset with degree 3.

using polynomial kernel with degree 3 model get an accuracy of 86%.

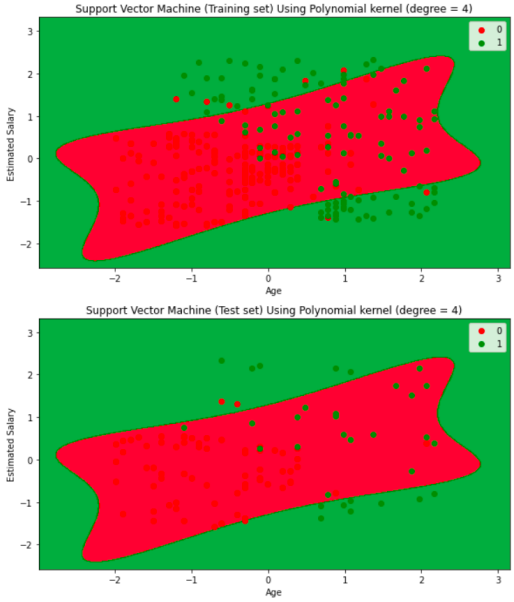

Fig. 6. figure shows performance of polynomial kernel on training and test dataset with degree 4.

using polynomial kernel with degree 4 model get an accuracy of 79%.

we have achieved highest accuracy when we have used polynomial kernel with degree 3.

C. RBF Kernel

RBF kernels are the most generalized form of kernelization and is one of the most widely used kernels due to its similarity to the Gaussian distribution. The RBF kernel function for two points X1 and X2 computes the similarity or how close they are to each other.

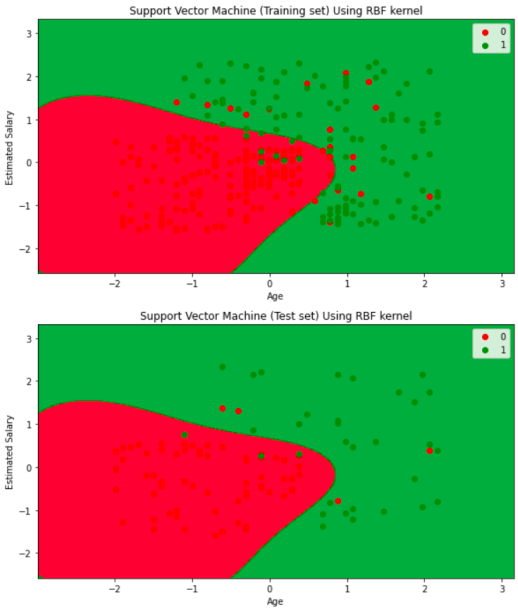

Fig. 7. figure shows performance of RBF kernel with no gamma on training and test dataset.

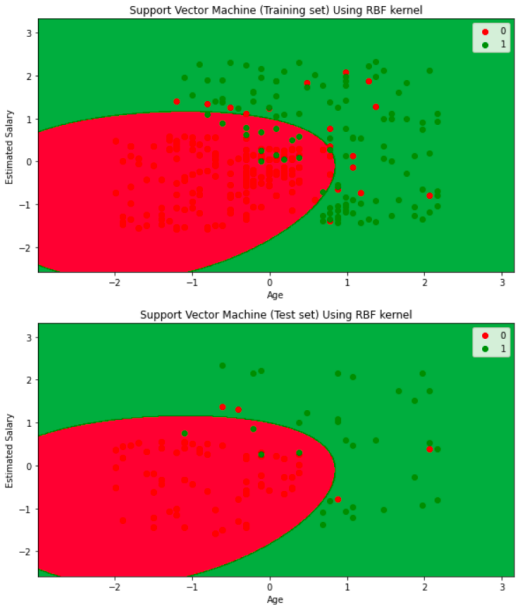

Fig. 8. figure shows performance of RBF kernel with 0.1 gamma on training and test dataset.

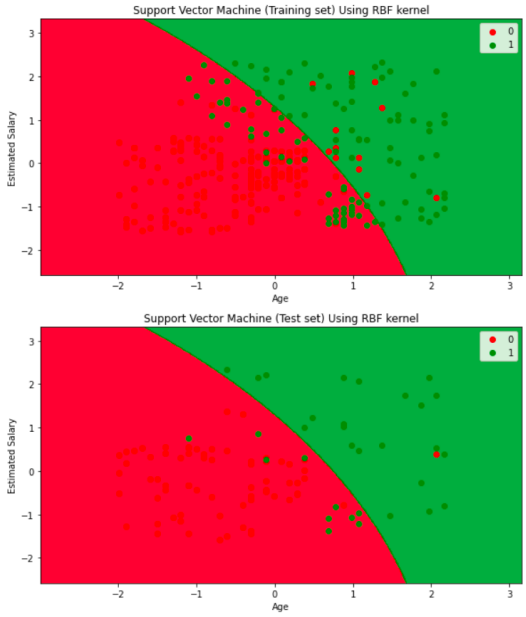

Fig. 9. figure shows performance of RBF kernel with 0.01 gamma on training and test dataset.

we have tried RBF kernel with no gamma and reveive an accuracy of 93% and gamma of 0.1 with 92% accuracy and lastly with gamma of 0.01 with 89% accuracy. so to conclude RBF kernel with no gamma will gives the best accuracy.

Conclusion

In this paper, we have shown the comparative results using different kernel functions. Fig 1 to 9 shows the comparative results of different data samples using different kernels linear, polynomial and RBF. The experiment results are encouraging .It can be seen that the choice of kernel function and best value of parameters for particular kernel is critical for a given amount of data. Fig 7,8,9 shows that the best kernel is RBF for infinite data and multi class.

References

[1] https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a
[2] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[3] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
[4] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
[5] https://www.sciencedirect.com/topics/engineering/linear-support-vector-machine
[6] https://www.scilab.org/tutorials/machine-learning-classification-svm

APPENDIX

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os
from matplotlib.colors import
    ListedColormap

file = pd.read_csv('{}/Social_Network_Ads
    .csv'.format(os.getcwd()))

from numpy import array
from numpy import argmax
from sklearn.preprocessing import
    LabelEncoder
from sklearn.preprocessing import
    OneHotEncoder

X = file.iloc[:, [2,3]].values
Y = file.iloc[:, 4].values

from sklearn.model_selection import
    train_test_split
X_Train, X_Test, Y_Train, Y_Test =
    train_test_split(X, Y, test_size =
    0.25, random_state = 0)

# Feature Scaling

from sklearn.preprocessing import
    StandardScaler
sc_X = StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)

# Linear Kernel
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear',
    random_state = 0, C = .001)
classifier.fit(X_Train, Y_Train)

# Predicting the test set results
Y_Pred = classifier.predict(X_Test)

from sklearn.metrics import
    accuracy_score
print(accuracy_score(Y_Pred,Y_Test) *
    100)

# Polynomial Kernel
from sklearn.svm import SVC
classifier = SVC(kernel = 'poly', degree
    =2, random_state = 0)
classifier.fit(X_Train, Y_Train)

# Predicting the test set results
Y_Pred = classifier.predict(X_Test)

print(accuracy_score(Y_Pred,Y_Test) *
    100)

# RBF Kernel
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf',
    random_state = 0, gamma=0.01)
classifier.fit(X_Train, Y_Train)

# Predicting the test set results
Y_Pred = classifier.predict(X_Test)

print(accuracy_score(Y_Pred,Y_Test) *
    100)
```