Support Vector Machine Optimal Kernel Selection

Lilly Priya Puppala[1]

[1]*Student, Dept. of Computer Science and Engineering, Lovely Professional University, Punjab, India*

Abstract - *Support vector machine (SVM) is capable of outcompeting every other learned model algorithm in terms of accuracy and other high-performance metrics by its high dimensional data projection for classification. Nevertheless, the performance of the Support vector machine is greatly affected by the choice of the kernel function which helps in the same. This paper discusses the working of SVM and its dependency on the kernel function, along with the explanation of the types of kernels. The focus is on choosing the optimal kernel for three different types of data that vary on volume of features and classes to conclude the optimal choice of the kernel for each type of the three datasets. For performance measures, we used metrics such as accuracy, kappa, specificity and sensitivity. This study statistically examines and compares each type of kernel against the mentioned metrics.*

Key Words: **Support Vector Machine, Kernels, Heart Disease Data, Digit Recognition, Social Network Ads Data, Classification.**

1. INTRODUCTION

Machine learning has become a central part of the world's advancing and leading companies such as Google, Facebook, YouTube, and so on. It is a domain of study which allows models and algorithms to learn from experiences or patterns from trained data to implement such acquired knowledge into predicting future outcomes without being explicitly programmed. Predictive analytics of machine learning has abundant applications in real-world scenarios except that algorithms are complex and the interpretability is limited disallowing us to be aware of which model or which parameters of the chosen model to use for a particular problem. [6]One of such high-performance models is SVM which when deployed with the right kernel function gives more accurate results compared to other models. However, there is no hard and fast rule to know the right kernel that could be used with the SVM, leaving us with the trial and error method. In this paper, we will be comparing 4 different types of kernels and evaluating their performance on 3 different types of data to conclude the optimal kernel for each type of dataset problem.

2. SUPPORT VECTOR MACHINE (SVM)

SVM is a well-pronounced machine learning algorithm that learns from input training data (supervised learning approach) and produces outcomes for both regression and classification problems [6]. Being most useful for classification labelling, it shows an uncommon property of minimising error in practicality while simultaneously increasing the accuracy of the result. Its applications in the real world range from detecting credit card fraud and heart disease prediction to handwritten digit recognition and letters recognition [1]. SVM does not require or assume the data trained for pattern recognition to be normally distributed; however, it expects trained data to resemble the distribution of the data to be tested or desired to be used [2]. Although it is a high dimensional mathematical function implemented to classify the features as desired labels, the working of SVM mainly depends on the following two concepts –

1) A separating hyperplane with maximal margin and minimal soft margin (Figure 1)

A line separating two clusters of features classifying them evidently as 2 respective labels not only in a 2-dimensional space but in a higher-dimensional space. Thus, accounting for the feature vectors being separated into two clusters by a plane is a separating hyperplane. [2] Although such hyperplane-based classification algorithms exist, SVM is unique as it follows the statistical mathematics theorem of choosing the hyperplane with maximum distance from the nearest feature vector of both the clusters leading to maximum marginal distance from either of the margins of the 2 clusters. Nonetheless, there could exist feature vectors that could infiltrate the wrong side of the hyperplane. Therefore, to deal with such errors a soft margin should be set that defines the number and distance to which the feature vectors could be allowed to be wrong [1].

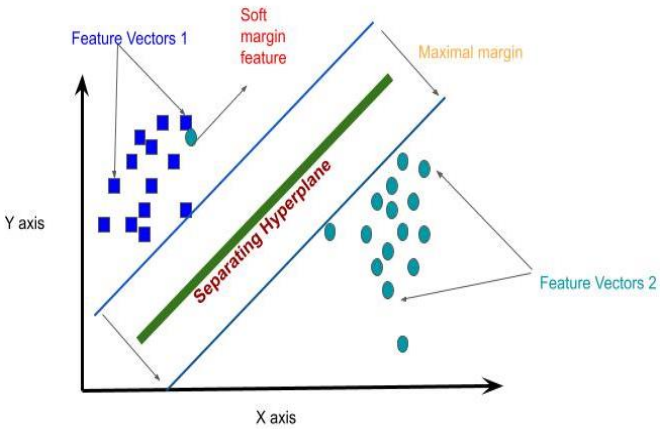

Figure 1: SVM Classification of 2 clusters with Separating Hyperplane , Maximal margin and Soft margin. [3]

2) Kernel function - A mathematical function that converts a low dimension space into higher dimensions to perform high dimension classification on low dimension data features

is a kernel function. This is important in cases in which the data is inseparable with one labelled group being near zero, while the other has large absolute values, making it impossible for a single point to separate the 2 labels. In such cases, a soft margin would be exponentially large. The essence, working and types of kernel functions will be discussed further in detail below.

2.1 Support Vector Machine Kernels

As discussed earlier, the kernel function helps to linearly separate unambiguous data by changing it into high dimensional space. To know the essence of using kernel functions, let's see the following cases. (Figure 2) Here, features are dispersed in a way that a straight line (hyperplane in terms of SVM) cannot be used to separate the 2 groups. However, employing a kernel function that squares the features' values change the space from 2 dimensional into 4-dimensional space, thus allowing linear separation using a hyperplane (Figure 3). [2] As we cannot draw features in 4 dimensions, we project the hyperplane back in 2 dimensions, thus making it a curved line (Figure 4). Based on the necessity and type chosen, respective mathematical functions would be applied to the data space. The most common issue is when and which kernel function to employ to our SVM model as projecting low dimension to high dimensional data space would also exponentially increase the possible solutions. Thereby making it difficult and slow for the algorithm to generalise and give an accurate solution. Thus, knowing which type of kernel function to use on which kind of data is important as it is crucial to classify our data without introducing irrelevant dimensions.

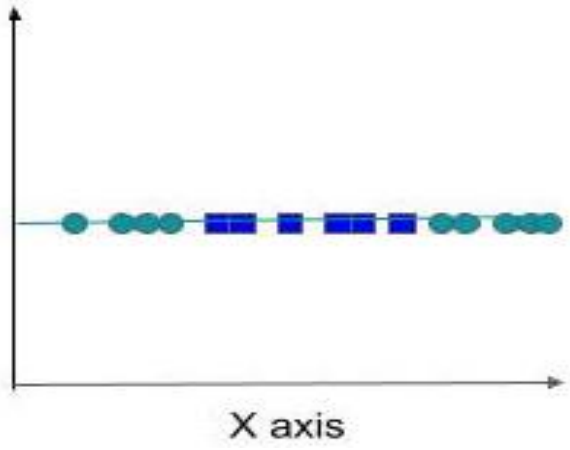

Figure 2: Case 1[2]

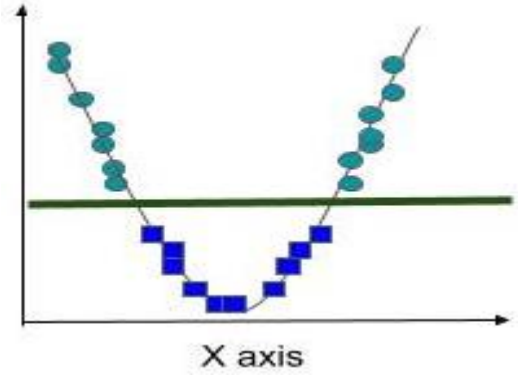

Figure 3: Case 2[2]

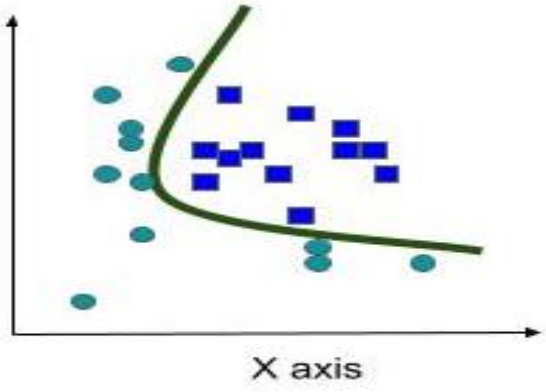

Figure 4: Case 3[2]

2.2 Types of Kernels

Unfortunately, past research dictates that the optimal way to choose the right kernel function is a trial and error method as each data would differ in distribution, features and desired classification. However, this process is time-consuming. Therefore, we are discussing each type of kernel function and their application to respective cases for better interpretability of which kernel function to deploy to our SVM model [1].

1. Linear kernel - A one-dimensional kernel, being the most simple and common kernel is much faster compared to other kernels and uses only the C parameter to optimise [1].

Formula: F(X, XJ) = SUM(X.XJ)

2. Polynomial kernel - A decision boundary that separates given data by representing the similarity of vectors in the data in a feature space over polynomials of the original variables used in the kernel. It is a more generalized representation of the linear kernel however it is known to be less efficient and accurate [3].

Formula: F(X, XJ) = (X.XJ + 1)^d

3. Gaussian Radial Basis Function (RBF) - This adds a radial basis method to improve the transformation. Optimization parameter gamma has been voluntarily added to the model

which ranges from 0 to 1. Furthermore, the most preferred value for gamma is 0.1 [1].

Formula: F(X, XJ) = exp(-gamma * ||X - XJ||^2)

4. Sigmoid kernel - this is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons. Also known as Multi-layer Perceptron Kernel or Hyperbolic Tangent Kernel as gamma, r and d are adjustable kernel functions which are adjusted based on the data [3].

Formula: F(X, XJ) = tanh(X*alpha*XJ + c)

Where, X , XJ = the data we are trying to classify.

d = adjustable power

gamma = optimization parameter

alpha = slope

c = constant

3. DATA DEFINITION AND METHODOLOGY

To give a better conclusion on which kernel type to use for which data, we will be using three different types of datasets to do three different kinds of analysis. Given is the data definition and elaborative structure of each of the datasets

1) Heart Disease Data Set- The first dataset we use is Cleveland Heart Disease from the UCI Machine Learning Repository. This dataset describes a range of conditions that are used to predict if a person is suffering from heart disease or not. This is categorical data with 2 classifications of presence or absence (1 or 0) of heart disease. This data set consists of 303 feature vectors out of which 138 absent cases and 165 are present cases of heart disease. We will be using 14 features for each feature vector.

The 14 features used are -

- Age: displays the age
- Sex: displays the gender: 1 = male; 0 = female
- 3.Chest-pain type (cp): displays the type of chest-pain: 1 = typical angina; 2 = atypical angina; 3 = non - anginal pain; 4 = asymptotic
- Resting Blood Pressure (trestbps): displays the resting blood pressure
- Serum Cholesterol (chol): displays the serum cholesterol
- Fasting Blood Sugar (fbs): fasting blood sugar value - 1 = fbs > 120mg/dl 0=fbs < 120mg/dl
- Resting ECG (restecg): displays resting electrocardiographic results: 0 = normal; 1 = having ST-T wave abnormality;2 = left ventricular hypertrophy
- Exercise-induced angina (exang):1 = yes;0 = no
- Oldpeak: ST depression induced by exercise relative to rest
- Slope: Peak exercise ST-segment :1 = upsloping;2 = flat;3 = down sloping
- Ca: Number of major vessels (0–3) colored by fluoroscopy
- Thal: displays the thalassemia
- Diagnosis of heart disease (target): Displays whether suffering from heart disease or not :0 = absence;1 = present.

2) Digit Recognition Dataset - The second dataset we used is the Handwritten Digit recognition dataset from Kaggle which contains two CSV files of train and test data, each of which contains grey-scale images of hand-written digits ranging from 0 to 9 as feature vectors. This prediction is a classification problem of many features and vectors (in comparison to the other 2 datasets we are using) with its result being classified into one of the numbers from 0 to 9. We will be using only the train CSV file which contains 42,000 feature vectors each having 785 feature attributes.

Each of the 785 feature attributes is a pixel of the image in the dataset (28 height x 28 widths) displaying a pixel value indicating the degree of lightness or darkness (0 to 255) of that pixel.

3) Social Media Ads Dataset - The third and final dataset we will be using is Social Media Ads Dataset from Kaggle which has a product's social media advertising campaign. Here, we will be analysing the dependency of the purchase of the product over a small set of features and classify whether the product is purchased or not by prediction. This dataset consists of 400 feature vectors with 257 not purchased and 143 purchased entries and 5 feature attributes, out of which we will be using only 3 feature attributes.

The 3 feature attributes are-

- Age - the age of the target audience - range(18 to 60)
- Estimated Salary - the estimated salary of the target audience - range (15000 to 150000)
- Purchased - whether the target audience has purchased the product or not: 0= not purchased; 1 = purchased

The 3 datasets differ with the volume of features and type of classification; however, we will be implementing the analysis using the following methodology for all the 3 problems.

3.1 METHODOLOGY

Each dataset is preprocessed and divided into a training set and testing set randomly in the ratio 75:25, allowing the SVM model to learn from the training set and form patterns to predict classification for the testing set. The training data is fed to the SVM model with all features using the function SVM of the e1071 package of R programming along with the particular kernel function to use as a parameter and the model prediction is tested against the actual values using the predict function. The model is implemented for all 4 kernels with their respective kernel function names and the results are compared with performance metrics [5] using the confusionMatrix function (Figure 5).

The performance metrics [4] show the following performance of the model using each kernel function

1) Accuracy - the ratio of correct predictions to total predictions. This is mainly useful when the FN and FP have similar costs[4].

Accuracy = (TP+TN)/(TP+FP+FN+TN)

2) Sensitivity - This is also known as the true positive rate and is used to measure the proportion of positive features that were correctly classified. This is used when the occurrence of false negatives (Showing purchased when not purchased) is unacceptable compared to getting low accuracy[4].

Sensitivity = TP/(TP+FN)

3) Specificity - This is also known as the true negative rate and is used to measure the proportion of negative examples that were correctly classified. This is used when one wants to not raise false alarms by maximizing the false positives (Showing the presence of heart disease instead of absent) compared to other metrics[4].

Specificity = TN/(TN+FP)

4) Kappa statistic - This adjusts accuracy by accounting for the possibility of a correct prediction. It ranges from 0 to 1[3].

Poor agreement: < 0.20

Moderate agreement: 0.20 to 0.80

Perfect agreement: 0.80 to 1.00

[4]Where, TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

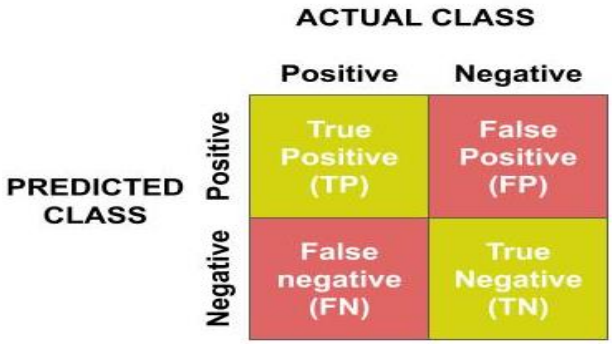

Figure 5: Confusion Matrix for Performance Evaluation

Table -1: Kernel Performance of Heart Disease and Social Network Ads Datasets [5]

	Heart Disease Data				Social Network Ads Data			
Kernel Type	Accuracy	Sensitivity	Specificity	Kappa	Accuracy	Sensitivity	Specificity	Kappa
Linear	0.76	0.7059	0.8049	0.5133	0.8	0.8906	0.6389	0.5495
Polynomial	0.7333	0.5882	0.8533	0.451	0.78	[1] [1] [1] [1]0.9375	0.5	0.4782
RBF	0.76	0.7647	0.7561	0.5182	0.9	0.9062	0.8889	0.7856
Sigmoid	0.76	0.6471	0.8537	0.5084	0.75	0.8281	0.6111	0.4474

Table-2: Kernel Performance of Digit Recognition Dataset[5]

	Digits Recognition Data			
	Accuracy	Sensitivity (mean)	Specificity (mean)	Kappa
Linear	0.9351	0.9342	0.9928	0.9278
Polynomial	0.9746	0.9714	0.9971	0.9717
RBF	0.9797	0.9927	0.9991	0.9774
Sigmoid	0.8377	0.8356	0.9819	0.8196

4. RESULTS

After evaluation of performance (Table 1)(Table 2), the results of each dataset are as follows

For the heart disease dataset - the accuracy is the same for all the kernels used; however, the polynomial kernel has shown relatively low accuracy. Since specificity and

sensitivity are also important for this type of dataset, the RBF and Sigmoid kernels have shown good sensitivity and specificity index respectively. The kappa statistic is also maximum for the RBF kernel.

Furthermore, for the Social Network Ads Dataset - While accuracy, specificity and kappa statistic is maximum for RBF kernel, the Polynomial kernel has shown the best sensitivity. On the contrary, the Sigmoid kernel has shown the least performance.

Meanwhile, for the Digit Recognition dataset problem, RBF has shown the best accuracy sensitivity specificity and the kappa statistic.

5. CONCLUSION

We have discussed and viewed the results of the comparative study of SVM kernels for 3 different types of datasets - namely Heart Disease, Digit recognition and Social-network Ads. We used 4 types of common kernels - Linear, Polynomial, Gaussian Radial Basis Function (RBF) and Sigmoid kernels of Support Vector Machine (SVM) model to label the required variable for classifying each of the datasets using all the features. We evaluated the Support Vector Machine (SVM) model performance with each of the kernel types using performance metrics - Accuracy, Sensitivity, Specificity, Kappa statistic. After evaluation, we can conclude that the Gaussian Radial Basis Function (RBF) kernel is optimal for datasets having no prior knowledge. However contrary to the popular belief that linear kernel works well with large datasets, it has shown less performance compared to Gaussian Radial Basis Function (RBF) kernel. Sigmoid Kernel has shown relatively low accuracy for all the three types of dataset problems; however, it has shown optimal performance for modicum volume datasets that need high sensitivity. For future work, investigation can be done on combining different kernels for better performance.

REFERENCES

[1] L. Bhambhu And D. K. Srivastava, “Data Classification using Support Vector Machine,” Journal of Theoretical and Applied Information Technology.

[2] W. S. Noble, “What is a Support Vector Machine?,” Nature Publishing Group, vol. 24, 2006.

[3] I. S. Al-Mejibli, D. H. Abd, J. K.Alwan and A. J. Rabash, “Performance Evaluation of Kernels in Support Vector Machine,” in 1st Annual International Conference on Information and Sciences (AICIS), 2018.

[4] M. M. Islam, H. Iqbal, M. R. Haque and M. K. Hasan, “Prediction of Breast Cancer using Support Vector Machine and K-Nearest Neighbors,” in IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017.

[5] M. Hussain, S. K. Wajid, A. Elzaart and M. Berbar, “A Comparison of SVM Kernel Functions for Breast Cancer Detection,” in Eighth International Conference Computer Graphics, Imaging and Visualization, 2011.

[6] B. Boser, I. Guyon and V. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in 5th Annual ACM Workshop on COLT (ed. Haussler, D.), Pittsburgh, 1992.