Pulsar Detection
Support Vector Machines

Devcharan K
Indian Institute of Technology, Madras

***Abstract*—Support Vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. It uses a technique called kernel trick to transform the data and then based on these transformations, it finds an optimal boundary between possible outputs. In this paper, data about pulsar candidates is given and Support Vector Machine algorithms are used to classify potential pulsars.**

I. Introduction

A pulsar is a highly magnetized rotating compact star (usually neutron stars but also white dwarfs) that emits beams of electromagnetic radiation out of its magnetic poles. Since the discovery of pulsars by Jocelyn Bell and Antony Hewish at Cambridge in 1967 using a pen chart recorder, pulsar searching has come a long way. Modern pulsar surveys use high performance computing facilities to perform an extensive range of signal processing and search algorithms. These methods are designed to maximize sensitivity to weak, rapid, and dispersed pulsar signals often buried in large amounts of terrestrial radio frequency interference. The aim of this paper is to be able to classify the given samples as pulsars or not.

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier. A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outlier detection. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier

In this paper, continuous data about pulsar candidates is considered for evaluation. The aim is to classify the samples based on the features into two classes of pulsar and not pulsar.

In Section II, Support Vector Machines and techniques used are explained. In Section III, the problem and the method to solve it is shown. In Section IV, the concepts used to prepare the model is evaluated. Finally the inferences of the analysis are presented in Section V.

II. Support Vector Machines

The original SVM algorithm was developed by Vladimir N Vapnik and Alexey Ya. Chervonenkis in 1963. At that time, the algorithm was in early stages. The only possibility is to draw hyperplanes for linear classifier. In 1992, Bernhard E. Boser, Isabelle M Guyon and Vladimir N Vapnik suggested a way to create non-linear classifiers by applying the kernel trick to maximum-margin hyperplanes. The current standard was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995.

SVMs can be used for linear classification purposes. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enable us to implicitly map the inputs into high dimensional feature spaces.

A. Intuition

1) Hyperplane: A hyperplane is a decision boundary which separates between given set of data points having different class labels. The SVM classifier separates data points using a hyperplane with the maximum amount of margin. This hyperplane is known as the *maximum margin hyperplane* and the linear classifier it defines is known as the *maximum margin classifier*.

2) Support Vectors: Support vectors are the sample data points, which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.

3) Margin: A margin is a separation gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the line to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get maximum margin. Fig.1 illustrates these concepts visually.

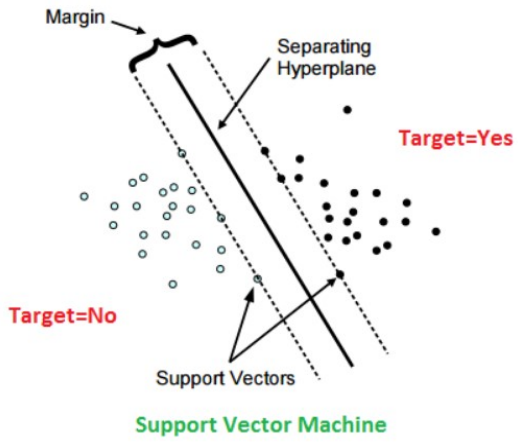

Fig. 1. Margin in SVM

In SVMs, the main objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum margin hyperplane in the following 2 step process –

1) Generate hyperplanes which segregates the classes in the best possible way. There are many hyperplanes that might classify the data. We should look for the best hyperplane that represents the largest separation, or margin, between the two classes.
2) So, we choose the hyperplane so that distance from it to the support vectors on each side is maximized. If such a hyperplane exists, it is known as the **maximum margin hyperplane** and the linear classifier it defines is known as a **maximum margin classifier**.

The following diagram illustrates the concept of maximum margin and maximum margin hyperplane.

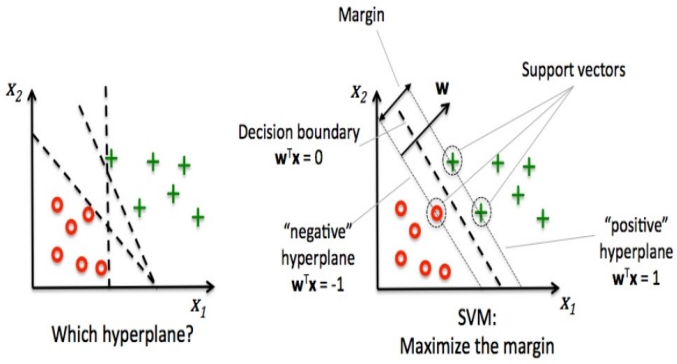

Fig. 2. SVM Intuition

B. Kernel Trick

In practice, SVM algorithm is implemented using a kernel. It uses a technique called the kernel trick. In simple words, a kernel is just a function that maps the data to a higher dimension where data is separable. A kernel transforms a low-dimensional input data space into a higher dimensional space. So, it converts non-linear separable problems to linear separable problems by adding more dimensions to it. Thus, the kernel trick helps us to build a more accurate classifier. Hence, it is useful in non-linear separation problems.

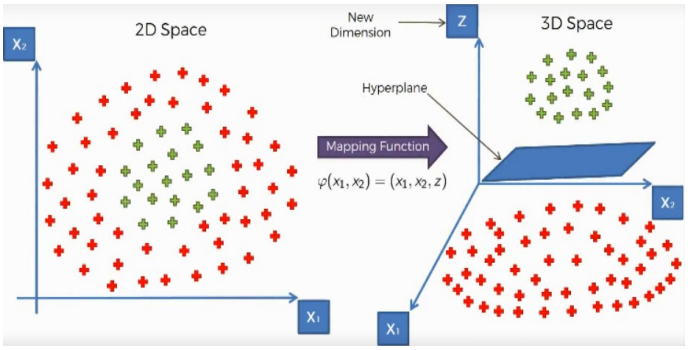

Fig. 3. Kernel Intuition

We can define a kernel function as

$$K(\bar{x}) = \begin{cases} 1, & \text{if } ||\bar{x}|| \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

In the context of SVMs, there are 4 popular kernels – Linear kernel,Polynomial kernel,Radial Basis Function (RBF) kernel (also called Gaussian kernel) and Sigmoid kernel.

III. The Problem

The data given has 8 continuous variables namely *Mean of the integrated profile, Standard deviation of the integrated profile, Excess kurtosis of the integrated profile, Skewness of the integrated profile, Mean of the DM-SNR curve, Standard deviation of the DM-SNR curve, Excess kurtosis of the DM-SNR curve, Skewness of the DM-SNR curve* and one target variable *Class* which takes wither 0 or 1 as values.

A. Data Cleaning and Exploration

The data had some missing values which were replaced with the mean and median of the columns data after considering the distributions of the data. The missing values in “IP Kurtosis”, “DM-SNR” and “DM-SNR Skewness” were replaced with median after considering the skewness of their distributions as shown in Fig. 4.

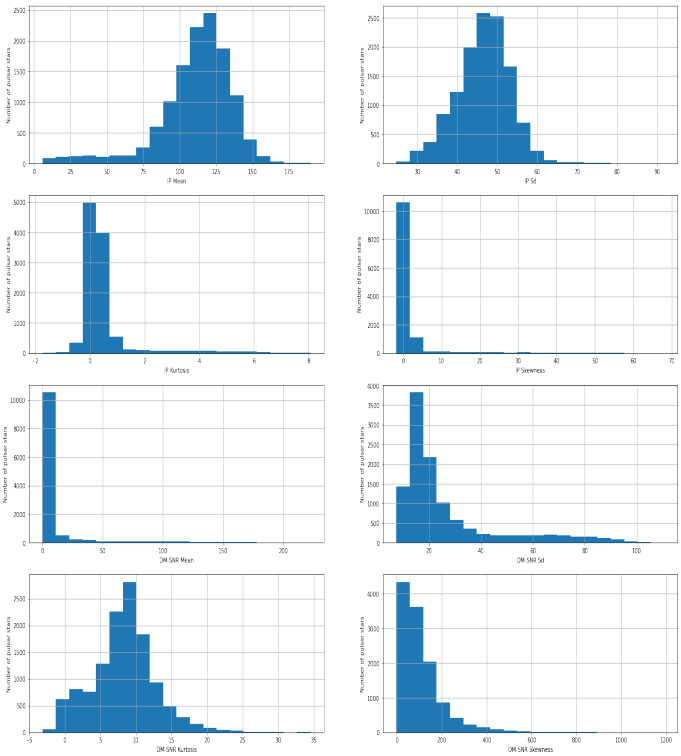

Fig. 4. Distributions of all continuous variables

IV. Model Evaluation

The model to be created is a Support Vector Machine. The above mentioned techniques are used to create the model.

A. Feature Engineering

Feature Engineering is the process of transforming raw data into useful features that help us to understand our model better and increase its predictive power. Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data pre-processing step.

B. SVM with default parameters

First the model is built with default parameters of C=1.0, kernel=rbf and gamma=auto among other parameters. The model accuracy score obtained with this setting is 0.9796 (or 97.96%). The Confusion Matrix is $\begin{bmatrix} 2276 & 9 \\ 42 & 179 \end{bmatrix}$

C. SVM with increased C parameter

Since there are outliers in the dataset, the C value is increased to remove the outliers. The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. The model is rebuilt with C = 1000.0. The model accuracy score obtained with this setting is 0.9808 (or 98.08%). There is a slight improvement in model accuracy. The confusion matrix is $\begin{bmatrix} 2273 & 12 \\ 36 & 185 \end{bmatrix}$

D. SVM with Linear Kernel

The model accuracy of SVM with linear kernel with C = 1000.0 comes out to be 0.9781 (or 97.81%). Confusion matrix is $\begin{bmatrix} 2277 & 8 \\ 47 & 174 \end{bmatrix}$

E. SVM with Polynomial Kernel

The model accuracy with polynomial kernel with C = 100.00 is 0.9792 (or 97.92%) The confusion matrix is $\begin{bmatrix} 2275 & 10 \\ 42 & 179 \end{bmatrix}$

F. SVM with Sigmoid Kernel

The model accuracy with Sigmoid kernel with C = 100.00 is 0.8767 (or 87.67%) The confusion matrix is $\begin{bmatrix} 2118 & 168 \\ 142 & 79 \end{bmatrix}$

G. Test Data

Applying the different models to the test data returned a varied number of Pulsar predictions.

Model applied	Number of Pulsars predicted	Model Accuracy
Default SVM	381	98.08%
Linear Kernel SVM	400	97.81%
Polynomial Kernel SVM	403	97.92%
Sigmoid Kernel SVM	475	87.67%

TABLE I
PREDICTIONS OF EACH SVM MODEL

V. Inferences

The Support Vector Machine to classify pulsars yields a very good performance as indicated by the model accuracy score of $97 - 98\%$. There is no sign of over-fitting. The confusion matrix yields a very good performance. SVM using Sigmoid kernel gives lower accuracy score than the other three models.

The model accuracy score along with the the number of pulsars predicted with each model shows that the best possible estimate of number of pulsars in the test dataset is around 400.

VI. Conclusion

The Support Vector Machine model is a very powerful classifier as indicated by the accuracy scores above. The final number of pulsars predicted by the models is about 400.

References

[1] Aurelien Geron,"Hands on Machine Learning with Scikit-Learn and Tensorflow", 2017.
[2] "Support Vector Machines" Wikipedia [Accessed on 10 November] Available: https://en.wikipedia.org/wiki/Support-vector_machine
[3] Andreas C. Muller and Sarah Guido, "Introduction to Machine Learning with Python"
[4] Support Vector Machines with Scikit-learn Available: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python
[5] SVM Classifier, Introduction to Support Vector Machine Algorithm Available: https://dataaspirant.com/support-vector-machine-algorithm/