March-April 2020
ISSN: 0193-4120 Page No. 2088 - 2094

Pulsar Search using Supervised Machine Learning: An Application of Astronomy

Pushkaraj Sadegaonkar
E&TC Department, Sinhgad College of Engineering, Pune,India
pushkaraj.sadegaonkar@gmail.com

R. B. Dhumale
E&TC Department, Sinhgad College of Engineering, Pune,India
rbd.scoe@gmail.com

Article Info
Volume 83
Page Number: 2088 - 2094
Publication Issue:
March - April 2020

Article History
ArticleReceived*: 24July 2019*
Revised*: 12September 2019*
Accepted*: 15 February 2020*
Publication*: 18March 2020*

Abstract

Identification of pulsars in radio astronomy is a tough task because the radio telescopes detecting most of the radiations are noise. Selecting proper radiations emitted by pulsars is a cognitively demanding process. In this paper, the Support Vector Machine-based classifier is implemented to identify pulsar stars from noise by classifying pulsar candidates from non-pulsar candidates. The support vector machine concept is explained with the classification of data. The algorithm is implemented using four statistics values of the two input features. The statistics are mean, standard deviation, excess kurtosis, and skewness. The input features are integrated profile and DM-SNR curve. The two class Support Vector Machine algorithm is trained using 17,897 observations. Average predicting accuracy obtained is 97.54%.

Keywords – *pulsars, radio astronomy, support vector machine, candidates.*

I. INTRODUCTION

Pulsars are some kind of neutron stars that are formed after the disintegration of massive stars. These pulsars emit radiations that can be detected on earth. Their study is very imperative in astronomy as it provides means for understanding the different characteristics of gravitation, cosmological evolution and the composition of interplanetary medium etc. Hence their identification is very important and mostly difficult because there are a large number of pulsars like signals detected by radio telescopes. Compared to these false signals, the genuine signals that are actually emitted by pulsars are very few. Conventional approach for their identification was manual by a human observer which is time consuming and cognitive demanding [1]-[4].

Pulsar candidate identification with ***Machine Learning*** [ML] algorithms is proposed in [5]. ***ML*** is a discipline of ***Artificial Intelligence (AI)*** that specializes in algorithms that learn through data which is fed to them. Unlike conventional methods, ML algorithms use computational methods to learn information from data. The more the data, more trained the algorithm is and better is the performance. In ML, there are two main types namely supervised learning and unsupervised learning. Supervised learning deals mainly with two types of problems which are Classification and Regression. Classification in particular contains algorithms such as ***Support Vector Machine (SVM)***, Discriminant Analysis, Naïve Bayes and K-Nearest Neighbor. The classification algorithm classifies data points into either two classes or more than two classes. SVM is used efficiently when there is binary classification [6].

SVM is mainly an extension of the ***Support Vector Classifier (SVC)***, which is itself an extension of the ***Maximal Margin Classifier (MMC).*** The drawback of the MVC for which data points need to be split separately was crossed by the SVC. And further, the SVC constraint that cannot construct a nonlinear boundary for linear segmented data was eliminated by the SVM [7].

In the SVM, the ***hyperplane*** is a flat decision boundary that can or may not pass through the origin and is ***the dimension of (P-1) for P dimensional space***. For example, a hyperplane is a line in two dimensional spaces. And suppose if a point is not on the hyperplane, it will be on the positive or negative side of the hyperplane. So, for a dataset with N observations and P features, our goal is to train the classifier using training data to properly classify test data based on the hyperplane using its feature measurements [8].

Although the hyperplane accurately categorizes the data points into two parts, there exist numerous numbers of hyperplanes. All possible hyperplanes can be achieved either by ***moving at a small distance or by rotating at a small angle without touching any data points***. So in all these possibilities a hyperplane must be chosen. The selected hyperplane is known as the ***maximal margin*** hyperplane with the shortest distance from all the data points. At least that means that the data point with the least vertical distance from the hyperplane is also known as the margin, and furthermore, you mean the largest margin or the maximum margin. Then the test data is classified according to whose maximum margin is based on which side of the hyperplane. Therefore, ***classification is known as maximum margin classifier (MMC)***. The data points that lie on the margin are known as ***Support Vectors***. They are called support vectors in the sense that slight change in their position causes the hyperplane to change and also the classifier. Change in position of data points other than support vectors do not affect the hyperplane and a very small number of data points decide the position of the hyperplane [9].

When the data points are not linearly separable, the hyperplane does not perfectly separate the data points into two halves. In such cases, addition of a single data point leads to large change in the hyperplane and the classifier would become overfit to the data. Such classifier might not be desirable for classification. The solution is obtained by using SVC. In SVC, some specific data points are allowed to violate the margin and sometimes the hyperplane too. The margin is known as ***soft margin*** and is based on the fact that it is worthwhile to misclassify a few training data points in order to correctly classify remaining other data points. The numbers of data points that can violate the margin are bounded by a tuning factor C and it is often chosen by cross validation. For SVC, support vectors are the data points that either lie on the margin or the points that violate the margin. The fact that a very few data points affect the classification makes SVC a very robust classifier. For classification ofdata points with ***nonlinear decision boundaries***, the feature space is enlarged using functions of the input variables such as quadratic and cubic terms. That is, the SVC can be fitted using 2P features. The solution to SVC is obtained by taking inner products of data points instead of taking the data points themselves. The inner products are further taken over by ***generalization of the inner products using function known as Kernel function***. The kernel function is used to quantify the similarity between the two data points. A polynomial kernel with degree d is used for linear boundary. When the nonlinear boundary is needed, d should be greater than 1. The SVM is obtained when a nonlinear kernel is used with SVC [10].

In this paper a classifier based on SVM algorithm is used which identifies the given star as a pulsar. The algorithm is implemented using mean values of two features of candidates. The two features being integrated profile and DM-SNR curve.

II. Support Vector Machine for Pulsar Identification

The algorithm is implemented using mean values of two features of candidates. The two features being integrated profile and DM-SNR curve. The equation of the hyperplane is given by Eq. 1.

$$\beta 0 + \beta 1X1 + \beta 2X2 + \ldots. + \beta nXn = 0 \qquad (1)$$

Where, β0 is known as bias,

β1…. βn are the weights,

X1….Xn are the coordinates of the point on the hyperplane. Eq. 1 can also be written as,

$$D(x) = \beta 0 + \beta iXi = 0 \qquad (2)$$

Where, i = 1….n.

This is known as the ***Linear Discriminate Function D(x)***.

If the point X does not lie on the hyperplane, then it lies either on the positive or on the negative side of the hyperplane. Then Eq. 2 becomes

$$D(x) = \beta 0 + \beta iXi > 0 \text{ If X lies on positive side} \qquad (3)$$

$$D(x) = \beta 0 + \beta iXi < 0 \text{ If X lies on negative side} \qquad (4)$$

Then the classification can be done by the following rules,

$$D(x) = \beta 0 + \beta iXi > 0, \quad Xi \in \text{class C1 i.e. a pulsar candidate} \qquad (5)$$

$$D(x) = \beta 0 + \beta iXi < 0, \quad Xi \in \text{class C2 i.e. not a pulsar} \qquad (6)$$

Now, to classify the given data point as a pulsar or not a pulsar, the values of β0 and βi must be found. But, before that it should be ensured that for point Xi taken from class C1, the D(x) should be positive or greater than zero. If it is not, then the values of β0 and βi are modified such that it comes to the positive side of the hyperplane. I.e. the hyperplane is adjusted accordingly. Similarly, for point Xitaken from class C2, the D(x) should be negative or less than zero. And for this also the values of β0 and βi are modified such that it comes to the negative side of the hyperplane.

If the hyperplane lies closer to one data point of class C1 and farther from a data point of class C2, then the hyperplane is putting large bias against class C2 and a penalty towards class C1. In that case, a small noise can misclassify the given data point. Hence, the hyperplane should be equally and maximum distanced from the classes.

Now, for every input data point Xi with ‘p’ features, the output Yi is a class of value ±1. It is given by,

$$Xi \in \text{class C1} \qquad Yi = +1 \qquad (7)$$

$X_i \in \text{class } C2 \qquad Y_i = -1$ *(8)*

The generalized equation is written as,

$$Y_i(\beta_0 + \beta_i X_i) \geqslant 0 \qquad (9)$$

To keep the margin maximum, Eq. 9 becomes

$$Y_i(\beta_0 + \beta_i X_i) \geqslant M \qquad (10)$$

Where, M is the margin.

The distance of a data point X from the hyperplane is calculated by,

$$\frac{\beta 0 + \beta i X i}{||\beta i||} \geqslant M \qquad (11)$$

The parameter 'M' is set to unity by proper scaling. Then Eq. 10 becomes,

$Y_i\ (\beta 0 + \beta i X i) = 1$ *If Xi is support vector* *(12)*

$Y_i\ (\beta 0 + \beta i X i) > 1$ *If Xi is not a support vector* *(13)*

The SVM is a linear machine and the classification of data points depends only on the *Support Vectors*. The data points other than *Support Vectors* do not affect the classifier. Thus only a few points decide the classification process.

For Eq. 11, to maximize the margin 'M', the weights 'βi' should be minimized and the bias 'β0' should be maximized. To minimize the weights, the function is given by,

$$G(x) = \beta t\, \beta = \ \beta.\beta \qquad (14)$$

Here, $\frac{1}{2}$ β.β is to be minimized. This is constraint optimization problem and is converted to unconstraint optimization by using *Lagrangian Multiplier*. It is given as,

$$L(\beta,\ \beta_0) = \tfrac{1}{2}\ \beta.\beta - \textstyle\sum \alpha_i\ [Y_i\ (\beta_0 + \beta_i X_i) - 1] \qquad (15)$$

Where, $\alpha_i$is *Lagrangian Multiplier*. Optimization of Eq. 15 can be obtained by taking derivatives with respect to β_0 and β and equating it to zero.

$$\frac{\partial L}{\partial \beta 0} = \frac{\partial L}{\partial \beta 0}[\tfrac{1}{2}\beta.\beta \ - \textstyle\sum \alpha_i Y_i(\beta_i.X_i) - \sum \alpha_i Y_i(\beta_0) - \sum \alpha_i] \qquad (16)$$

$$\frac{\partial L}{\partial \beta 0} = \frac{\partial L}{\partial \beta 0}[- \textstyle\sum \alpha_i Y_i(\beta_0)\ ] \qquad (17)$$

Where, remaining terms are constant. Equating Eq. 17 to zero can be written as Eq. 18.

$$\textstyle\sum_{i=0}^{p} \quad \alpha_i Y_i = 0 \qquad (18)$$

Where, p is the number of features. Eq. 18 gives one of the constraints to design the SVM. Now, taking derivative of Eq. 15 w.r.t. β can be written as Eq. 19. ,

$$\frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial \beta}[\tfrac{1}{2}\beta.\beta \ - \textstyle\sum \alpha_i Y_i(\beta_i.X_i) - \sum \alpha_i Y_i(\beta_0) - \sum \alpha_i\ ] \qquad (19)$$

$$\frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial \beta}[\tfrac{1}{2}\beta.\beta \ - \textstyle\sum \alpha_i Y_i(\beta_i.X_i)\ ] \qquad (20)$$

Where, remaining terms are constant. Equating Eq. 20 to zero can be written as Eq. 21.

$$\beta = \textstyle\sum \alpha_i Y_i X_i \qquad (21)$$

Putting Eq. 18 and Eq. 21 in Eq. 15 can be written as Eq. 22.

$$L(\beta,\ \beta_0) = \textstyle\sum \alpha_i - \tfrac{1}{2}\sum\ \alpha_i\ .\alpha_j\ .Y_i\ .Y_j.(X_i\ .X_j) \qquad (22)$$

Where, $\beta.\beta$ is dot product written as $\alpha_i\ .\alpha_j$. Here, the target is to find the *Lagrangian*

Multiplier 'α_i' that will maximize Eq. 22. If the *Lagrangian Multiplier* is zero, then corresponding X_i is not support vector. If it is very high, then corresponding X_i has high influence on hyperplane. If it is extra ordinary high, then corresponding X_i is an outlier. For unknown data point 'i', the classification decision will be given by Eq. 23 and Eq. 24,

$$D_i = \beta_i . X_i + \beta_0 \qquad (23)$$

$$D_i = Sign \sum \alpha_i Y_i X_i X_i + \beta_0 \qquad (24)$$

In Eq. 24, the sign decides the class of the data point. If the sign is positive, then it belongs to class C_1. If the sign is negative, then it belongs to class C_2. The value of β_0 is calculated by Eq. 25.

$$B_0 = \tfrac{1}{2}[min\ (\sum \alpha_i Y_i\ (X_j\ X_i)\ max\ (\sum \alpha_i Y_i\ (X_j\ X_i))] \qquad (25)$$

Hence, in this way finding the values of βi and β0, SVM is designed.

III. Algorithm

The algorithm is shown below and flowchart is given in Fig. 1.

1. Start.
2. Load the data.
3. Store attributes values of class 1 as 'data1'.
4. Store Attributes values of class 2 as 'data2'.
5. Plot these values on scatter plot.
6. Store values of both classes in 'data3'.
7. Label the 'data1' values as -1 and 'data2' values as +1.
8. Train the SVM classifier using 'data3'.
9. Predict scores over the grid.
10. If score -1, classify into group 1 else group 2.
11. Plot decision boundary around the data classified.
12. End

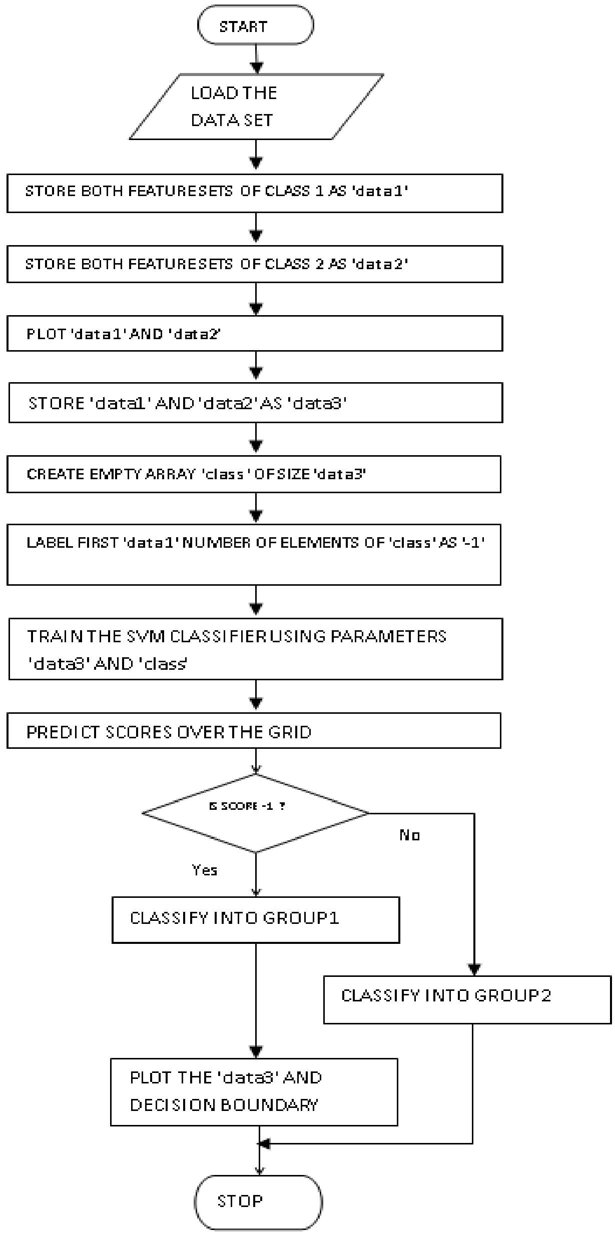
START
LOAD THE DATA SET
STORE BOTH FEATURESETS OF CLASS 1 AS 'data1'
STORE BOTH FEATURESETS OF CLASS 2 AS 'data2'
PLOT 'data1' AND 'data2'
STORE 'data1' AND 'data2' AS 'data3'
CREATE EMPTY ARRAY 'class' OF SIZE 'data3'
LABEL FIRST 'data1' NUMBER OF ELEMENTS OF 'class' AS '-1'
TRAIN THE SVM CLASSIFIER USING PARAMETERS 'data3' AND 'class'
PREDICT SCORES OVER THE GRID
IS SCORE -1 ?
No
Yes
CLASSIFY INTO GROUP1
CLASSIFY INTO GROUP2
PLOT THE 'data3' AND DECISION BOUNDARY
STOP

Fig.1. Flowchart of proposed method

IV. Result And Discussions

The dataset used in this experiment has mainly two attributes. First one is Integrated Profile and the other one is DM-SNR curve. The values of four statistics of the two features are considered. The statistics considered are Mean, Standard Deviation, Excess Kurtosis and Skewness. Hence there are total eight input values and one output value for each sample. The output 0 stands for class 1 which consists of non-pulsar candidates and output 1 stands for class 2 which consists of pulsar candidates. Total 17,897 samples are considered for training the data. Table below shows obtained results of the experiment.

The predicting accuracy of the dataset for Linear SVM for different number of samples is calculated by cross validation with five folds. Training Time with different samples is also noted down.

The Predicting Accuracy for the original dataset is 97.9% and the Training Time is 9.993 seconds. When the number of samples taken is 66 % of the original, Predicting Accuracy is 97.5% and Training Time is 6.41 seconds. Further when the number of samples taken is 50% of the original, the Predicting Accuracy is 97.3% and Training Time is 1.07 seconds. Hence as the number of samples are reduced both Predicting Accuracy and Training Time decrease.

The scatter plot of the original data and predicted data after trained by SVM algorithm is shown in Fig. 2. Blue dots indicate non-pulsar candidate and red dots indicate pulsar candidates. Support vectors are shown by dots and decision boundary by black line.

TABLE I. Predicting accuracy and Training Time

Samples	**Predicting Accuracy**	**Training Time (sec)**
17897	97.9%	9.9930
14433	97.6%	8.3629
12123	97.5%	6.4153
06350	97.4%	2.3307
03579	97.3%	1.0723

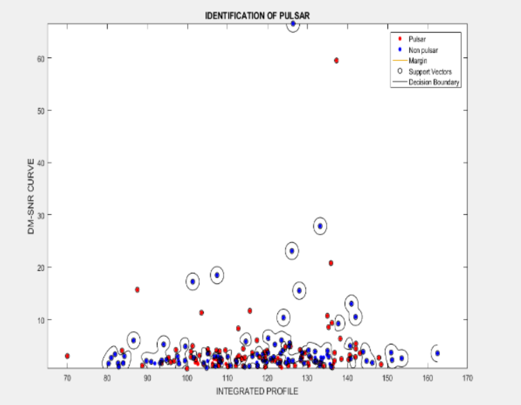

Fig. 2. *Scatterplot of original data set*

V. Conclusions

ML algorithms have been founded as an effective approach in the field of radio astronomy for identification of celestial objects such as pulsars. In particular supervised learning algorithms used for classification are used. The SVM algorithm is studied by experimenting it with a dataset which consists of observations of pulsar candidates. The dataset was collected during the High Time Resolution Universe Survey. The algorithm identifies a given candidate as a pulsar or no pulsar by performing binary classification on data points. The classification is done by demonstrating decision boundary in the feature space. The decision boundary or the hyperplane is explained with equations. The mathematical construction of the algorithm is given. The predicting accuracy of the algorithm is estimated by 5-fold cross validation after performing linear SVM. The average accuracy of the experiment was found to be 97.54%. It is observed that as the number of observations increase the corresponding accuracy also increases. The training time is also directly proportional to the number of observations. The original dataset and the dataset with predicted values are plotted.

References

[1] D. R. Lorimer and M. Kramer, 'Handbook of Pulsar Astronomy', Cambridge University Press, 2005.

[2] R. J. Lyon, 'PulsarFeatureLab', 2015, .

[3] R. J. Lyon et al., 'Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach', Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656.

[4] M. J. Keith et al., 'The High Time Resolution Universe Pulsar Survey - I. System Configuration and Initial Discoveries',2010, Monthly Notices of the Royal Astronomical Society, vol. 409, pp. 619-627. DOI: 10.1111/j.1365-2966.2010.17325.x.

[5] Ping Guo,Pulsar Candidate Identification with Artificial Intelligence Techniques., MNRAS,Vol. 1, Iss. 23, 2017.

[6] R. B. Dhumale, N. D. Thombare, P. M. Bangare, “Machine Learning: A Way of dealing with Artificial Intelligence”, In Proc. IEEE Int. Conf. on Innovation in Information and Communication Technology-2019 (ICIICT 2019), St. Peter College of Engineering and Technology, Chennai.

[7] R. B. Dhumale, N. D. Thombare, “Leaf Disease Classification using Machine Learning”, Proc. In International Conference of Ideas, Innovation and Impactsin Science and Technology (ICIIST-2019), Smt. KashibaiNavale College of Engineering, Pune, pp. 121, 19 March,2019.

[8] R. B. Dhumale, “An Overview of Artificial Neural Networks: Part 3 Activation Functions”, CiiT International Journal of Artificial Intelligent Systems and Machine Learning, Vol. 10. Iss. 3. April 2018, pp 66-71.

[9] An Introduction to Statistical Learning with applications in R by Trevor Hastie, Robert Tibshirani, Daniela Witten and Gareth James.

[10] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society 459 (1), 1104-1123, DOI: 10.1093/mnras/stw656.