Improving the prediction of players in IPL analytical system using Support Vector Machines (SVM) and Kernel functions

Aakash Tiwari, Ashwini Pandit, Pratyush Mohapatra, Merly Thomas

***Abstract*— As we tread into analysis of more complex datasets; the problem of nonlinear separability arises. As there are many attributes to be considered in player prediction at the Indian Premier League, the dataset becomes complex and nonlinearly separable. This leads to classification inaccuracies and inefficient prediction. Initially, this paper investigates the root cause of these inaccuracies followed by a comprehensible comparison of sophisticated supervised algorithms. Concludingly, it presents the use of SVM as a possible solution to the above mentioned shortcomings. It demonstrates the use of Kernel methods to increase the classification and prediction efficiency.**

***Index Terms*— SVM, WEKA, RBFKernel, PolyKernel**

I. INTRODUCTION

Cricket, the second most popular sport in the world after soccer, is a bat-and-ball game played between two teams of 11players. The latest form of cricket is the Twenty20 or T20 cricket. It involves two teams with each team batting maximum of 120 balls i.e. twenty overs and is completed in an about two and half hours, a much shorter duration. The England and Wales Cricket Board (ECB) in England originally introduced Twenty20 for professional inter-county competition in 2003 but it is now famous all around the world. This form of cricket has attracted large crowds to the stadiums and viewers on televisions because of its fast-paced nature and intense competition. Unexpected results of the games make it even more attractive.

Indian Premier League (IPL) is one of the famous Twenty20 cricketing events and takes place every year in India. Team selection is a highly critical process in every sport as players are selected based on their past performance. Forecasting future from the past is highly subjective and thus requires extraordinarily expert decision making. It becomes more prominent when a huge amount of money is involved.IPL is the fastest growing cricket league in the world. An important module of the Indian Premier League is player analytics. Data mining concepts can be applied to the dataset of players to get insightful patterns. The models that are developed can help in effective decision making. One of the most crucial aspect of any analytical system that takes into account supervised learning is the selection of classification algorithm.
This in turn is based on many factors. One such factor involved is separability of data. For better classification accuracy the class boundaries should be well defined. The classification models built classify data with higher efficiency and correctness when the predictive modelling classes are properly separated. We start off with the problem of nonlinear separability along-with the discussion the dataset used in the system. We then evaluate the effect of this problem on the supervised algorithms. Finally, we discuss the solution to this problem using SVM [1] and mention the future research scope.

Manuscript received March 02, 2015.
Aakash Tiwari, Student, FRCRCE, Mumbai University
Ashwini Pandit, Student, FRCRCE, Mumbai University
Pratyush Mohapatra, Student, FRCRCE, Mumbai University
Merly Thomas, Associate Professor, FRCRCE, Mumbai University

II. IPL DATASET VARIABLES

The system uses a dataset consisting of 500 entries of players categorized into batsmen and bowlers. The file is stored in an attribute relation file format (.arff). This make the file suitable for analysis in Netbeans IDE using WEKA [2] JAVA API. The different attributes are:

A. BATSMEN

Player_id - Indicates unique identifier for each player which is thus primary key for database.
Player_name - Indicates name of player.
Strike_rate - It is a measure of how frequently a batsman achieves the primary goal of batting, namely scoring runs. In other words strike rate is an ability of batsman to score runs without getting out.
Batting_Average - Gives batting average which means total number of runs divided by total number innings.
100s - Total number of centuries scored.
50s - Total number of half-centuries scored.
Age - Gives age of player.
Captain_% - Percentage of total number of times player played as a captain.
Nationality-Indicates whether player is an Indian player or foreigner.
Base price - Indicates base price assigned to each player.
Sold_player_price - It indicates price at which player was sold.
Decision – It indicates whether the player is playing in the match or not

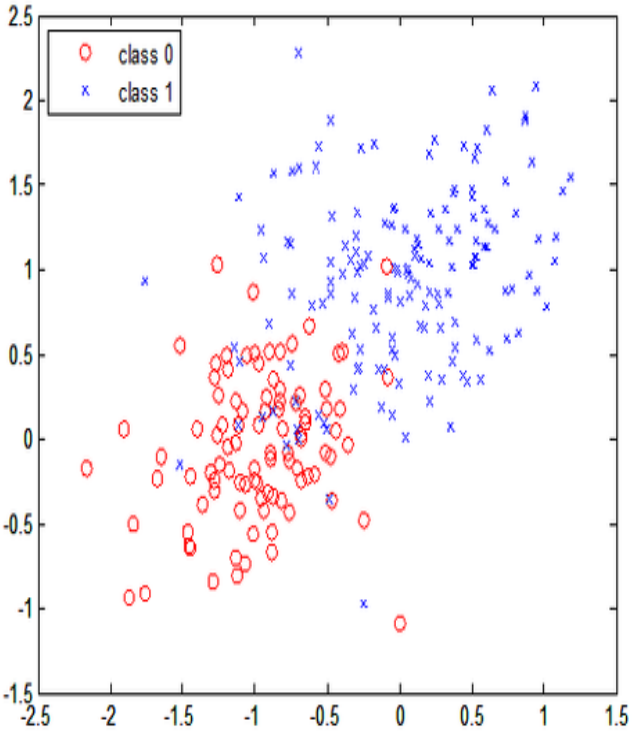
B. BOWLER

Player_Id - Indicates unique identifier for each player which is thus primary key for database.
Player_name - Indicates name of player.
Economy_rate - It is the average number of runs conceded per over.
Strike_rate - Indicates average number of balls bowled per wicket taken.
Catches taken - Total number of catches taken by player.

Age - Gives age of player.
Captain_% - Percentage of total number of times player played as a captain.
Nationality - Indicates whether player is an Indian player or foreigner.
Base price - Indicates base price assigned to each player.
Sold_player_price - It indicates price at which player was sold.
Decision – It indicates whether the player is playing in the match or not

III. NON-LINEARLY SEPERABILITY

Let's assume that you are having data points which can belong to either class 'A' or class 'B'. Let these data points be scattered in any n-dimension space. If any hyperplane in dimension (>= n) can separate these data points into two classes such that data points of class 'A' lie on one side of hyperplane and data points of class 'B' lie on other side, then you can say that your data set is linearly separable. For example, Assume data points are scattered in 2-dimensional space. If you can draw a line or hyperplane that can separate those points into two classes, then the data is separable. If not, then it may be separated by a hyperplane in higher dimensions. Still if any of the hyperplanes could not separate them, then the data is termed as non-linearly separable data.

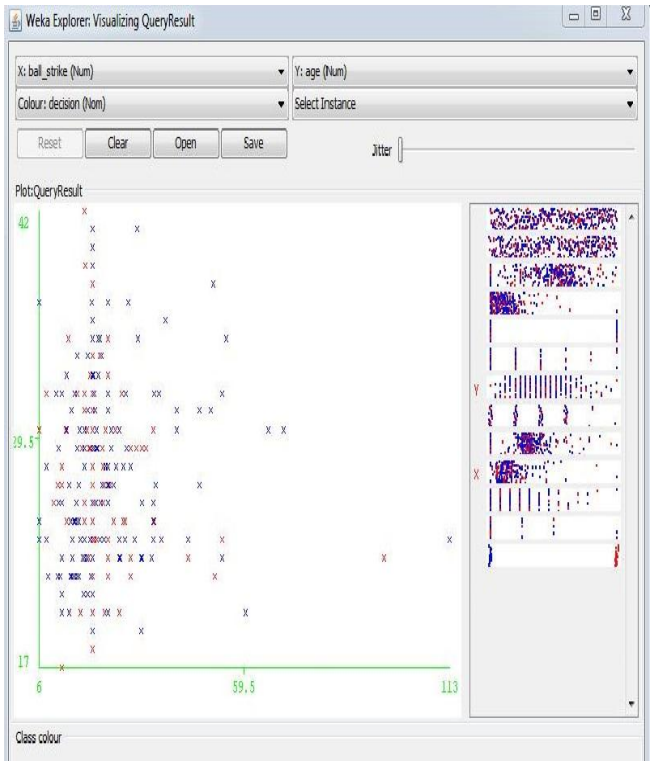


The above figure displays nonlinear separability of class 0 and class 1.

IV. DATASET VISUALIZATION

The Visualize panel in WEKA shows a scatter plot matrix, where individual scatter plots can be selected and enlarged, and analyzed further using various selection operators. Using the WEKA visualization [3] we see the nonlinear separability of IPL dataset used in the analytical system. In the figure we can see that our dataset cannot be separated linearly.

Here the Red Cross indicates class 0 i.e. playing in the match while the Blue Cross indicates class 1 i.e. not playing in the match. This leads undefined boundary between the decision variables.



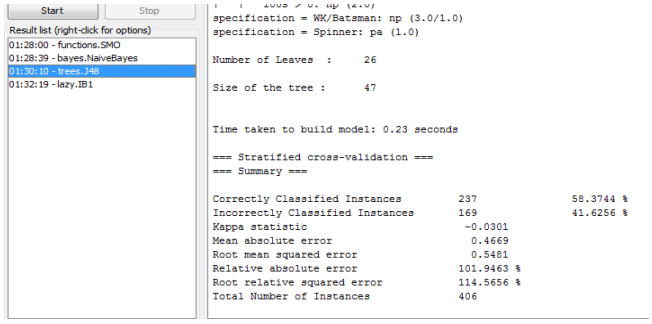
V. EFFECT ON LINEAR CLASSIFIERS

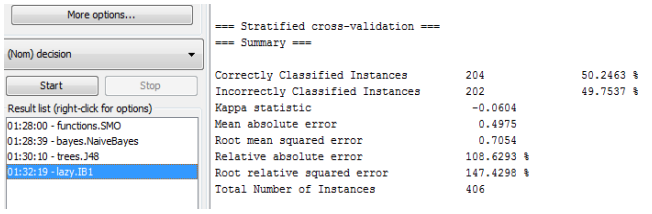
Using the WEKA JAVA API we can implement predictive modelling using various classification algorithms. The relative efficiency of the algorithms can be measured using a variety of parameters among which 'correctly classified instance' and 'kappa statistic' are the most reliable and unique.

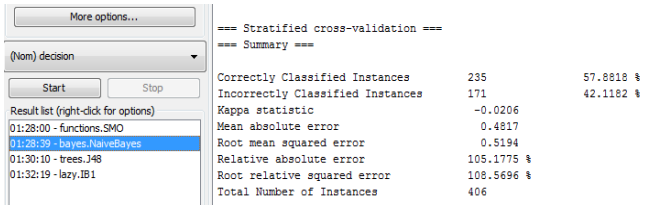
Correctly classified instance – The labels on the test set are supposed to be the actual correct classification. Performance is computed by asking the classifier to give its best guess about the classification for each instance in the test set. Then the predicted classifications are compared to the actual classifications to determine accuracy. If the prediction was correct we say that it is a correctly classified instance.

Kappa statistic [4] - Kappa is a chance-corrected measure of agreement between the classifications and the true classes. It's calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. **A value greater than or equal to 0 means that your classifier is doing better than chance**.

Following are the results obtained on running various algorithms on our dataset.







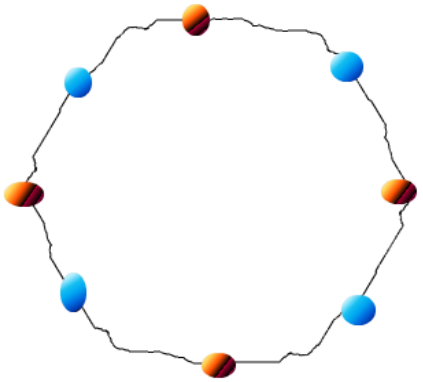
CLASSIFIER	CORRECTLY CLASSIFIED INSTANCE	KAPPA MEASURE
ITERATIVE DICHOMOTISER 3(J48)	58.3744%	-0.0301
NAÏVE BAYES CLASSIFIER	57.8818%	-0.0206
K NEAREST CLASSIFIER(IB1)	50.2463%	-0.604

We can infer that the correctly classified instance is less as well as the kappa measure is negative. This is due to the nonlinear separability of our dataset. To solve this problem we require a nonlinearly separable classifier.

VI. SVM

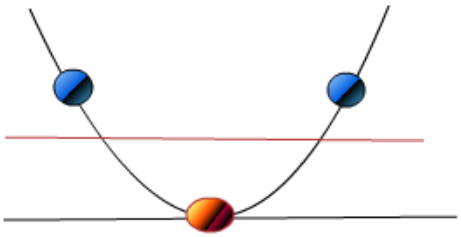
In typical setting of Classification problem, we'd be given some red dots and some blue dots in some space and we'd be required to find out a curve (called separating boundary) that can separate all blue dots from all red dots.

As it turns out, it is much easier and efficient to find out boundaries which are in the form of a straight line (or an analogous construct in higher dimensions called hyper plane) compared to curvy boundaries. Hyper-plane [5] is just a generalization of a line in 2D and plane in 3D. SVMs help us to find a hyper plane that can separate red and blue dots. SVMs can probably help to find out a separating hyper plane if it exists. What if there is no hyper plane which can separate red and blue dots? For example - imagine a circle in 2D with dots all over it such that adjacent dots are of alternating colours. There is no straight line (hyper plane in 2 dimensions) which can separate red and blue dots.



SVMs won't find a solution here simply because there is no solution! We're pretty much stuck - and this is where KERNEL comes into the picture. Consider a simple 1-Dimensional example. Assume that given points are as follows

A 1-dimensional hyper plane would be a vertical line. Clearly no vertical line can separate the given data set. However, if we project all points up to a two dimensional space using the mapping, we would get following data set in 2 dimensions:

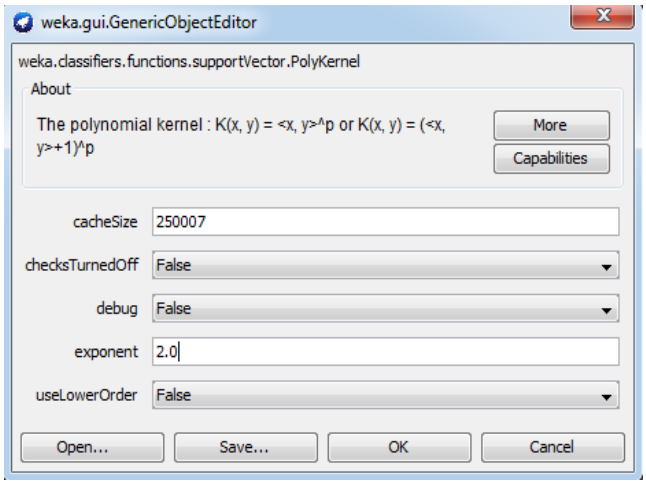


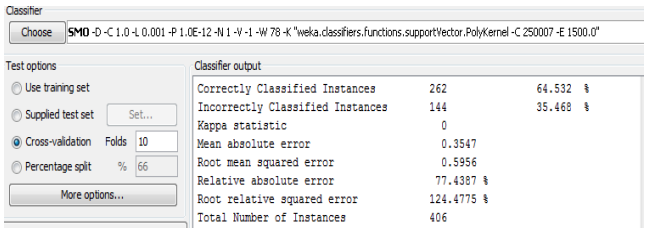
Now we can indeed find a hyper plane (an arbitrary line in 2 dimensions) that separates red and blue points - and hence our data can now be separated using an SVM. One possible separating line is shown in red ink. So the central idea is to be able to project points up in a higher dimensional space hoping that separability of data would improve. This mapping is called the KERNEL [6] function which in this case was Polynomial function with exponent 2 i.e. Quadratic Kernel.

VII. IMPLEMENTING SVM

Using the WEKA JAVA API we build the classification model using the IPL dataset. We can choose from a variety of

Kernels such as Polynomial, RBF [7] amongst others to convert to data to a dataset in another dimension.





CLASSIFIER	CORRECTLY CLASSIFIED INSTANCE	KAPPA MEASURE
SVM(SMO)	64.532%	0

For our dataset we use SVM (SMO [8]) and a quadratic polynomial kernel. This can be done by adjusting the exponent of the Polykernel to 2.0. This kernel function transforms our nonlinearly separable data to a linearly separable data in another dimension. Thus, the decision boundaries are clearer in this dimension which leads to a higher correctly classified instance and non-negative kappa statistic.

This leads to an efficient supervised predictive analysis.

VIII. CONCLUSION

Thus, we conclude that solving the problem of nonlinear separability using kernel based SVM helps in efficiently classifying the IPL dataset. As the dataset becomes linearly separable into another dimension, it can be separated properly. Such reliable prediction is of utmost importance during player selection in the Indian Premier League. Owing to the above finding, we can conclude that kernel based SVM is the most suited supervised learning algorithm for predictive analysis on the IPL dataset.

IX. FUTURE SCOPE

The SVM (SMO) that is used in the analysis in this paper is only applicable when the number of class variable is binary i.e. we can't have more than 2 classes. To solve this problem scientists have come up with multiclass SVM. Further research in this domain such as the creation of SVM classes like LIBSVM [9] has taken place. This field has further scope of research.

REFERENCES

[1] www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf
[2] Weka: Practical Machine Learning Tools and Techniques with Java Implementation : Ian h witten , Eibe Frank , Len Trigg
[3] weka.wikispaces.com/Visualization
[4] stackoverflow.com/questions/.../how-to-interpret-weka-classification
[5] An Idiot's guide to Support vector machines (SVMs) by R. Berwick, Village Idio
[6] www.support-vector.net/icml-tutorial.pdf
[7] http://www.quora.com/How-does-one-decide-on-which-kernel-to-choose-for-an-SVM-RBF-vs-linear-vs-poly-kernel
[8] Sequential Minimal Optimisation(SMO) : John C Platt, MSRN
[9] LIBSVM : A library for SVM , Chih Chung chang , National Taiwan University.