Bare Demo of IEEEtran.cls for Conferences

Anupam Panwar
Arizona State University
Tempe, Arizona
Email:

Aloma Lopes
Arizona State University
Tempe, Arizona
Email: aloma.lopes@asu.edu

Soumya Varanasi
Arizona State University
Tempe, Arizona
Email: varanasi.soumya@gmail.com

***Abstract*—Cricket is a popular sport played by about 120 million players and is the world's second most popular sport after association football. Moreover cricket betting is a multi-billion dollar market. Hence, there is a lot of demand in the prediction of the outcome of cricket matches particularly the One-Day International matches. There are various natural parameters as well as the ones that govern the rules of the match that makes it challenging to make an accurate prediction. Our project presents three algorithms for forecasting the instantaneous state of a match by taking into consideration the historical data. These three models are trained using the during-play parameters and tested using two matches as examples. The results of all the three models are compared with each other to determine the model with the best accuracy.**

***Keywords*—Dynamic, Regression, SVM, Random Forest, analytics**

I. Introduction

Cricket has the second largest viewership by population for any sport, which is responsible for the enormous commercial interest in game outcome prediction. Also, most of the work that has been done on cricket forecasting is mostly related to pre-match forecasting. Thus,demand for during-play forecasting is in high demand. There is a large amount of profit made in cricket betting even if you're not an expert in the game as cricket is a game dominated by statistics. Studying historical results of the stadium and researching variables that have affected variables in the past will help in the prediction.

The models that we present in our project, can serve various other purposes like analysis of team and player performance, identifying the key moments in a match, in addition to betting. We use the Logistic Regression Model, Support Vector Machine and Random Forest methods to build a forecasting tool that will predict if a team is going to win or lose by testing the current pre-match covariates and during-match statistics on the trained model. These forecasts of probabilities of win or lose can be done at any point as the match progresses.

II. Data and Features

The data files to form our data-set were taken from the website cricksheet.org which were in the YAML format. These files included data of 1165 One-Day International Matches played between May 2006 and February 2016. Each of these 1165 files provided ball-by-ball data for the corresponding match for the 1st and 2nd innings. The files had a lot of attributes that were irrelevant for our models. The YAML files were thus converted to CSV files with the required attributes as it allows data to be stored in a table structured format that can be queried or retrieved easily. This gives us a total of 1165×2 tables.

Our data-set thus formed had several attributes that could be potential covariates for our models. The attributes that could be used to form covariates are as follows: Date (primary key for the table), Venue, Balls, Runs, Wickets, Toss, Team-Playing, RPO (Runs per over), Team1, Team2, Team-Won.We use these attributes to calculate the pre-match and during-match covariates for our match.

A. Pre-match covariates

These are the features that are known before a match actually starts. Thus, attributes like the venue and toss are used to calculate the pre-match covariates. It is observed that the team that plays at home is always at an advantage due to the presence of home crowd for motivation. The venue is used to calculate the binary covariate *home*. Also we experiment on the binary covariate *toss* as a team that wins the toss may also have an added advantage of winning. We also calculate a team's form as given by Muhammad Asif in [1]. Let $y^t = 1$ if a team won the match played t matches ago.

$$form = \frac{\sum_{t=1}^{5} w(t,\theta) y^t}{\sum_{t=1}^{5} w(t,\theta)} \tag{1}$$

where $w(t, \theta) = (1-\theta)^{(t-1)}$ and $0 < \theta < 1$ The function $w(t, \theta)$ gives the highest weight to the most recent match. The difference in form between the two teams playing the particular match is calculated and added as a new attribute *fd* in addition to the above attributes of the table. This form difference *fd* is one more pre-match covariate.

B. In-play covariates

These are the covariates that change as the game is progressing. The runs scored, number of wickets, runs per over and the number of balls. These attributes are available for both the innings of the match and give the current state of the match. Our data-set gives the runs scored, the number of wickets lost, the run rate (runs per over) at every ball of every inning.

III. Model preparation

To prepare our model for training, we need to perform feature scaling or normalisation as the range if raw data values varies widely. To extract the data over which the model has to be trained, the complete ball-by-ball data is better organized one for each ball of each innings (first or second). Letting K be the number of balls bowled and I be the innings number, tuples corresponding to the I^{th} inning and K^{th} ball are extracted from our dataset and written to a new table. This model is dynamic as the parameters vary with the progression of the match. We use separate models for each innings because, the batting teams in each innings play with different strategies. The first team plays with the aim of scoring as many runs as possible while the second one plays with respect to the runs scored by the first one and has the aim of achieving its target. It may do so before the pre-allocated number of overs are exhausted.

A. Training

We use Support Vector Machines, Logistic Regression and Random Forest methods to train the model corresponding to the ball number where we want to predict whether the team will lose or win. Hence for training, we take the number of balls (K) and innings number (I) as input to train our model.

B. Validation

We use ten-fold cross validation to see how accurate our model is and know how effective the features and covariates are to predict the outcome of the game. We get validation accuracy of about 80

C. Testing

We take input the two teams playing, venue, current ball, runs, runs per over, inning, team-playing, toss, wickets and test it against our trained and validated model. Our model will output the pediction of whether the playing team will either win or lose.

IV. Conclusion

Appendix A
Proof of the First Zonklar Equation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.