Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach

by

Madan Gopal Jhawar, Vikram Pudi

in

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
(*ECML-PKDD 2016*)

Conference Center, Riva del Garda.

Report No: IIIT/TR/2016/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2016

Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach

Madan Gopal Jhanwar and Vikram Pudi

International Institution of Information Technology - Hyderabad,
Gachibowli, Hyderabad, India
{madangopal.jhanwar@research.iiit.ac.in,vikram@iiit.ac.in}
http://www.iiit.ac.in

Abstract. With the advent of statistical modeling in sports, predicting the outcome of a game has been established as a fundamental problem. Cricket is one of the most popular team games in the world. With this article, we embark on predicting the outcome of a One Day International (ODI) cricket match using a supervised learning approach from a team composition perspective. Our work suggests that the relative team strength between the competing teams forms a distinctive feature for predicting the winner. Modeling the team strength boils down to modeling individual player's batting and bowling performances, forming the basis of our approach. We use career statistics as well as the recent performances of a player to model him. Player independent factors have also been considered in order to predict the outcome of a match. We show that the k-Nearest Neighbor (kNN) algorithm yields better results as compared to other classifiers.

Keywords: Modeling Players, Modeling Teams, Winner Prediction, Supervised Learning

1 Introduction

Statistical modeling has been used in sports since decades and has contributed significantly to the success on field. Cricket is one of the most popular sports in the world, second only to soccer. Various natural factors affecting the game, enormous media coverage, and a huge betting market have given strong incentives to model the game from various perspectives. However, the complex rules governing the game, the ability of players and their performances on a given day, and various other natural parameters play an integral role in affecting the final outcome of a cricket match. This presents significant challenges in predicting the accurate results of a game.

The game of cricket is played in three formats - Test Matches, ODIs and T20s. We focus our research on ODIs, the most popular format of the game. To predict the outcome of ODI cricket matches, we propose an approach where we first estimate the batting and bowling potentials of the 22 players playing the match using their career statistics and active participation in recent games. We

use these player potentials to render the relative dominance one team has over the other. Taking two other base features into account, namely, toss decision and the venue of the match, along with the relative team strength, we adopt supervised learning algorithms to predict the winner of the match.

The major contributions of our paper are as follows:

- We propose novel methods to model batsmen, bowlers and teams, using various career statistics and recent performances of the players.
- To predict the winner of ODI cricket matches, we propose a novel dynamic approach to reflect the changes in player combinations.

2 Related Work

In literature, Duckworth and Lewis proposed a solution, called the D/L method [1], to reset targets in rain interrupted matches which was adopted by the International Cricket Council (ICC) in 1998. Further, the use of Duckworth-Lewis resources to assess players performances has been studied in [1], [2] and [3]. Optimal batting orders are discussed in [4] and [5]. The methods of graphical representation to compare players are presented in [6], [7], and [8]. [9] considers the strength of opponent team, along with other factors, in modeling the performance of batsmen and bowlers. However, like in any sport, winning is the ultimate goal in cricket. [10] takes into account various factors affecting the game including home team advantage, day/night effect and toss, etc., and uses the Bayesian classifier to predict the outcome of the match. [11] uses a combination of linear regression and nearest-neighbor clustering algorithms to predict the outcome of a match. They take into account both historical data as well as instantaneous state of a match while the game is still in progress. [12] studied the role of multiple factors including home field advantage, toss, match type (day or day and night), competing teams, venue familiarity, and season, etc., and applied Support Vector Machines(SVM) and Naive Bayes Classifiers for predicting the winner of a match.

In this paper, we embark upon a very critical aspect that the **team composition** changes over time, which has not been studied yet. A team is comprised of 11 players, and these 11 players are replaced over time. A team changes its composition depending upon the match conditions, venue, opponent team, etc. There could be various other reasons for the same, such as a player getting injured, or getting dropped from the team for his poor performance, or taking retirement from the sport itself. Figure 1 shows that on average at least 2 players change per match for each team. Therefore, relying completely on the historical data is not only insufficient, but also fallacious since it does not portray the current competence of a team. Taking such obsolete factors into account might lead us to incorrect conclusions.

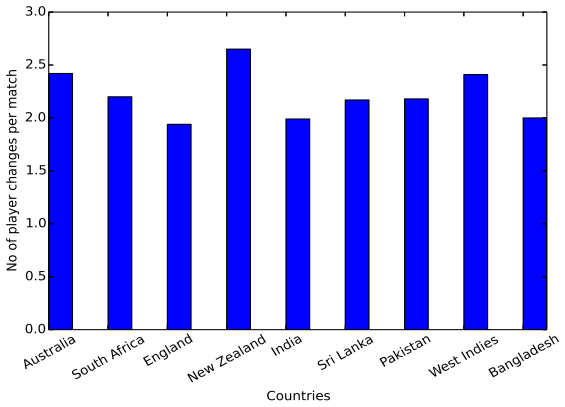

Fig. 1. The average number of player changes per match for all the teams in the years 2010-2014.

3 Methodology

In this section, we explain our approach to the problem in detail, including the definitions and the mechanics of various algorithms used to model the batsmen, bowlers and the teams.

Notations: We use A and B as the two teams competing in a match m, $P(T,m)$ as the set of all the players in team T playing in match m, and $\phi(p)$ as the set of the career statistics of the player p throughout the paper. The major career statistics of a player p (as used by [14], [15], [16] and [17]) are explained in Table 1.

Table 1. Player Features and their Notations

Notation	Description
$\phi_{Matches\_Played}$	*#Matches played by the player*
$\phi_{Batting\_Innings}$	*#Matches in which the player batted*
$\phi_{Batting\_Average}$	*#Runs scored divided by the #times the player got out*
$\phi_{Num\_Centuries}$	*#Times the player scored $\geq$ 100 runs in a match*
$\phi_{Num\_Fifties}$	*#Times the player scored $\geq$ 50 but less than 100 runs in a match*
$\phi_{Bowling\_Innings}$	*#Matches in which the player bowled*
ϕ_{Wkts_Taken}	*#Wickets taken by the player*
ϕ_{FWkts_Hauls}	*#Times the player has taken $\geq$ 5 wickets in a match*
$\phi_{Bowling\_Average}$	*#Runs conceded by the player per wicket taken*
$\phi_{Bowling\_Economy}$	*Average #runs conceded by the player per over bowled*

Modeling Batsmen: The batting ability of a player has a significant contribution in shaping the outcome of a match. A team usually comprises of a set of 6-7 specialist batsmen out of the 11 players. To model a batsman's aptitude, we have two types of statistics to get necessary insights into a player's characteristics. First, we examine his career performances to determine his potency as a contender. Second, we consider his recent match scores to analyze his prevailing *form*: where *form* of a batsman determines his contribution to the team in recent matches, which in turn reflects his confidence levels.

Algorithm 1 Modeling Batsmen

Input: Players $p \in \{P(A,m) \cup P(B,m)\}$, Career Statistics of player p: $\phi(p)$
Output: $Batsmen\_Score$ of all the players: $\phi_{Batsman\_Score}$

1: **for all** players $p \in \{P(A,m) \cup P(B,m)\}$ **do**
2: $\quad \phi \leftarrow \phi(p)$
3: $\quad u \leftarrow \sqrt{\frac{\phi_{Bat\_Innngs}}{\phi_{Matches\_Played}}}$
4: $\quad v \leftarrow 20 * \phi_{Num\_Centuries} + 5 * \phi_{Num\_Fifties}$
5: $\quad w \leftarrow 0.3 * v + 0.7 * \phi_{Bat\_Avg}$
6: $\quad \phi_{Career\_Score} \leftarrow u * w$
7: $\quad M \leftarrow$ *Last* 4 *matches played by* p
8: $\quad \phi_{Recent\_Score} \leftarrow mean(M^{p}_{Runs})$
9: **end for**
10: **for all** players $p \in \{P(A,m) \cup P(B,m)\}$ **do**
11: $\quad \phi_{Career\_Score} \leftarrow \frac{\phi_{Career\_Score}}{max(\phi_{Career\_Score})}$
12: $\quad \phi_{Recent\_Score} \leftarrow \frac{\phi_{Recent\_Score}}{max(\phi_{Recent\_Score})}$
13: $\quad \phi_{Batsmen\_Score} = 0.35 * \phi_{Career\_Score} + 0.65 * \phi_{Recent\_Score}$
14: **end for**

The pseudo code of the algorithm to model the batsmen for a given match is given in Algorithm 1. Lines 2-6 calculate a player's *Career_Score* using his overall career statistics. Variable u (line 3) is the ratio of the number of matches in which the batsman batted to the total number of matches he played. It captures whether the player is a full-time specialist batsman or not. Higher values of u indicate that the player often bats at the top of the batting order and hence he gets to bat in almost every match. On the other hand, lower values of u tell us that the player bats lower down the batting order and his chances of batting in the next match is also comparatively low. Variable ϕ_{Career_Score} (line 6) takes all the career statistics into account, and therefore signifies the *Career_Score* of the batsman. Similarly, lines 7-8 calculate the *Recent_Score* of a batsman. Variable M (line 7) holds the recent matches played by the player. Variable ϕ_{Recent_Score} (line 8) captures the *Recent_Score* of a batsman, which is the average number of runs scored by the player in his recent games. Since the $Career\_Score$ and the *Recent_Score* of players have different ranges, we have normalized them (lines 11-12) to lie in a common range of [0,1]. Finally, variable $\phi_{Batsman\_Score}$

(line 13) stores the *Batsman_Score* of a player which is a combination of his *Career_Score* and *Recent_Score*.

Modeling Bowlers: Even though, cricket is called a batsman's game, one cannot undermine the importance of specialist bowlers in a team. A team usually comprises of a set of 4-5 specialist bowlers out of the 11 players. To model a bowler, we are examining his career performances to estimate his potential for the next match.

Algorithm 2 Modeling Bowlers

Input: Players $p \in \{P(A,m) \cup P(B,m)\}$, Career Statistics of player p: $\phi(p)$
Output: *Bowler_Score* of all the players: ϕ_{Bowler_Score}

1: **for all** players $p \in \{P(A,m) \cup P(B,m)\}$ **do**
2: $\phi \leftarrow \phi(p)$
3: $u \leftarrow \sqrt{\frac{\phi_{Bowl\_Inngs}}{\phi_{Matches\_Played}}}$
4: $v \leftarrow 10 * \phi_{FWkts\_Hauls} + \phi_{Wkts\_Taken}$
5: $w \leftarrow \phi_{Bowl\_Avg} * \phi_{Bowl\_Eco}$
6: $\phi_{Bowler\_Score} \leftarrow \frac{u*v}{w}$
7: **end for**

The pseudo code of the algorithm to model bowlers for a given match is given in Algorithm 2. Variable u (line 3) is the ratio of the number of matches in which the bowler bowled to the total number of matches he played. It captures whether the player is a full-time specialist bowler or not. Higher values of u indicate that the player often bowls at the top of the bowling order and hence, he gets to bowl in almost every match. On the other hand, lower values of u tell us that the player is a part-time bowler who doesn't bowl in every match he plays and his chances of bowling in the next match are also comparatively low. Variables v and w (lines 4-5) consider other statistically significant features of a bowler. Finally, variable ϕ_{Bowler_Score} (line 6) takes everything into account, and therefore signifies the *Bowler_Score* of the player.

Notice that unlike batsmen, we haven't considered the recent performances of a bowler in calculating his *Bowler_Score*. This is due to the lack of data, as we do not have match-wise individual performances of every bowler.

Modeling Teams: The batsmen and the bowlers are the fundamental units of a team. Therefore, using the modeled batsmen and bowlers, we intend to define an overall score of a team with respect to the other. We define the batting score of a team as the summation of the batting scores of all its players. Similarly, the bowling score of a team is defined as the summation of the bowling scores of all its players. We have directly used the scores of all the players in the team score, as the variable u in the Algorithms 1 and 2 already takes care of the weighted contribution of individual players to the team score.

Our algorithm to find the relative strength between two teams, A and B, competing against one another in a match m is shown in Algorithm 3. Since the *Batsman Scores* and the *Bowler Scores* have different ranges, we first normalize them to lie in the same range of [0,1] (lines 1-4). Lines 5-8 of the Algorithm calculate the batting and bowling scores of both the teams. Variable $S(A/B)$ (line 9) captures the relative strength of team A against team B. The algorithm follows the fundamental aspect of the game strategy where the batsmen of one team work against the bowlers of the other team and vice-versa.

Algorithm 3 Relative Strength between Two Teams

Input: Players $p \in \{P(A,m) \cup P(B,m)\}$,
Batsman_Score: $\phi^p_{Batsman\_Score}$, *Bowler_Score*: $\phi^p_{Bowler\_Score}$

Output: Strength of Team A against Team B: $S_{A/B}$

1: **for all** players $p \in \{P(A,m) \cup P(B,m)\}$ **do**

2: $\quad \phi_{Batsman\_Score} \leftarrow \frac{\phi_{Batsman\_Score}}{max(\phi_{Batsman\_Score})}$

3: $\quad \phi_{Bowler\_Score} \leftarrow \frac{\phi_{Bowler\_Score}}{max(\phi_{Bowler\_Score})}$

4: **end for**

5: $Bat\_Strength_A \leftarrow \left(\sum_{p \in P(A,m)} \phi^p_{Batsman\_Score} \right)$

6: $Bowl\_Strength_A \leftarrow \left(\sum_{p \in P(A,m)} \phi^p_{Bowler\_Score} \right)$

7: $Bat\_Strength_B \leftarrow \left(\sum_{p \in P(B,m)} \phi^p_{Batsman\_Score} \right)$

8: $Bowl\_Strength_B \leftarrow \left(\sum_{p \in P(B,m)} \phi^p_{Bowler\_Score} \right)$

9: $S_{A/B} = \frac{Bat\_Strength_A}{Bowl\_Strength_B} - \frac{Bat\_Strength_B}{Bowl\_Strength_A}$

Feature Construction: To predict the winner of an ODI cricket match, we choose the venue of the match and the outcome of the toss as two other important features, along with the relative strength of one team against the other. Therefore, every match played between team A and team B in our dataset has three features: *Toss*, *Venue*, and $Stength_{A/B}$. The value of *Toss* is 1 if team A is batting first, or 0 otherwise. The value of *Venue* is 1 if the match is being played at a home ground of team A, 0 if it is played at a home ground of Team B, and 2 otherwise. The value of $Stength_{A/B}$ is the relative strength of team A against team B, as calculated in the Algorithm 3. The target variable *Winner* defines the winner of a match. It is a binary variable. The value of *Winner* is 1 if the winner of the match is team A, and 0 if the winner is team B. Using these three features, we apply machine learning algorithms to predict the winner of a match.

Note that out of the two competing teams, any one of them could be considered as team A and all the feature values and the target value would update accordingly.

4 Experiments and Results

Dataset: To retrieve all the required statistics, the entire dataset has been scraped from the *cricinfo* website [13]. The dataset includes all the matches played between 2010 and 2014. The dataset contains the basic match details including the two competing teams, the outcome of the toss, the date when it was held, the venue and the winner of the match for all the matches. Along with these, the career statistics of the participating players and their performances in every match is also included.

We have restricted our study to only top 9 ODI-playing teams, namely, Australia, South Africa, India, England, Sri Lanka, Pakistan, New Zealand, Bangladesh and West Indies. Since the impact of the nature on the game cannot be foreseen, a total of 109 matches which were either interrupted by rain or ended up in a draw/tie, have been removed from the dataset. Finally, we divided the dataset into two parts, namely, the test data and the training data. The training dataset contains all the matches played during the years 2010 to 2013, and the test dataset contains all the matches played in the year 2014. There are a total of 299 matches in training dataset and 67 matches in test dataset.

Learning Weights: To assign the weights to various features in the Algorithms 1 and 2, we have used the 5-match ODI series played between India and Sri Lanka in July, 2012. A series of consecutive matches was deliberately chosen to study the impact of the recent scores of a batsman on his upcoming performances. The estimated scores of the players are compared against their actual performances. After exhaustive experimentation, the final weights are chosen such that the top 6 performing batsmen and bowlers (in terms of runs scored and wickets taken, respectively) from both the teams match with the top 6 batsmen and bowlers estimated by our algorithms.

Binary Classifiers: Using various binary and numeric features and the outcome of the match as the label, we evaluated a large number of binary classifiers using their scikit-learn implementations [18] to generate supervised classification models, including SVM, Random Forests, Logistic Regression, Decision Trees and kNN. We used the *sweep* feature to experiment with all the possible values and combinations of the parameters for all the algorithms. The efficacy of the kNN algorithm, with k=4, was statistically superior to those obtained by the best models of other classifiers, as shown in Figure 2. The idea of using the data of future matches to predict the outcome of past matches is absurd. Consequently, we could not carry out any sort of cross-validation procedure as it would interfere with the chronological order of the data.

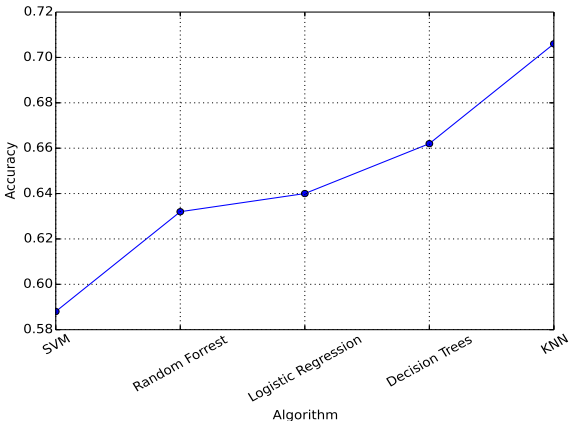

Fig. 2. Accuracy of different classifiers used

The only obstacle we faced while evaluating our approach is the inability to compare against previous models like [10] and [11], due to the different underlying datasets used. Our dataset does not have some of the features used by them. For instance, we do not have the details on the timings of the matches (day/night) as used by [10], and the instantaneous state of the matches at multiple stages as used by [11]. However, we compared our model with two other baseline models – the team winning the toss wins the match (Model_1), and the team with positive relative strength, as calculated in the algorithm 3, wins the match (Model_2). The results are tabulated in Table 2. The superiority of our model against the others proves the significance of the combination of various features used.

Although we cannot directly compare these results with the prior state-of-the-art approaches due to differences in the dataset, it is noteworthy that the best accuracy in predicting the outcome of ODI cricket matches reported so far in the literature is between 0.68 and 0.70 ([11]). Team-wise winning accuracy, as predicted by our model, is shown in Figure 3.

Table 2. Comparing our kNN based model with other baseline models

Model	**Accuracy**
Model_1	0.56
Model_2	0.63
Our Model	0.71

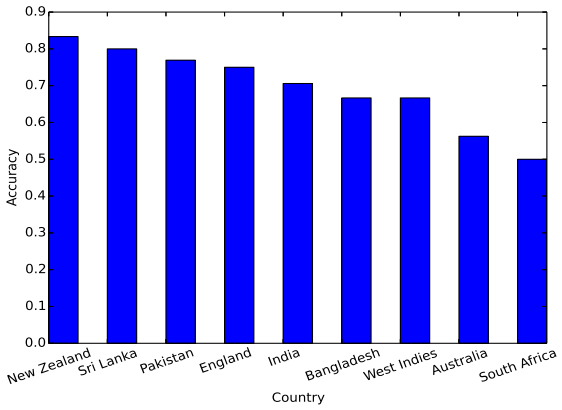

Fig. 3. Accuracy for different countries

5 Conclusion

The paper addresses the problem of predicting the outcome of an ODI cricket match using the statistics of 366 matches. The novelty of our approach lies in addressing the problem as a dynamic one, and using the participating players as the key feature in predicting the winner of the match. We observe that simple features can yield very promising results.

References

1. Duckworth, Frank C., and Anthony J. Lewis. "A fair method for resetting the target in interrupted one-day cricket matches." Journal of the Operational Research Society 49.3 (1998): 220-227.
2. Beaudoin, David, and Tim B. Swartz. "The best batsmen and bowlers in one-day cricket." South African Statistical Journal 37.2 (2003): 203.
3. Lewis, A. J. "Towards fairer measures of player performance in one-day cricket." Journal of the Operational Research Society 56.7 (2005): 804-815.
4. Swartz, Tim B., Paramjit S. Gill, and David Beaudoin. "Optimal batting orders in one-day cricket." Computers and operations research 33.7 (2006): 1939-1950.
5. Norman, John M., and Stephen R. Clarke. "Optimal batting orders in cricket." Journal of the Operational Research Society 61.6 (2010): 980-986.
6. Kimber, Alan. "A graphical display for comparing bowlers in cricket." Teaching Statistics 15.3 (1993): 84-86.
7. Barr, G. D. I., and B. S. Kantor. "A criterion for comparing and selecting batsmen in limited overs cricket." Journal of the Operational Research Society 55.12 (2004): 1266-1274.
8. Van Staden, Paul Jacobus. "Comparison of cricketers bowling and batting performances using graphical displays." (2009).

9. Lemmer, Hermanus H. "THE ALLOCATION OF WEIGHTS IN THE CALCULATION OF BATTING AND BOWLING PERFORMANCE MEASURES." South African Journal for Research in Sport, Physical Education and Recreation (SAJR SPER) 29.2 (2007).
10. Kaluarachchi, Amal, and S. Varde Aparna. "CricAI: A classification based tool to predict the outcome in ODI cricket." 2010 Fifth International Conference on Information and Automation for Sustainability. IEEE, 2010.
11. Sankaranarayanan, Vignesh Veppur, Junaed Sattar, and Laks VS Lakshmanan. "Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction." SDM. 2014.
12. Khan, Mehvish, and Riddhi Shah. "Role of External Factors on Outcome of a One Day International Cricket (ODI) Match and Predictive Analysis."
13. ESPN Cricinfo, `http://www.espncricinfo.com`
14. Barr, G. D. I., and R. van den Honert. "Evaluating batsman's scores in test cricket." South African Statistical Journal 32.2 (1998): 169-183.
15. Croucher, J. S. "Player ratings in one-day cricket." Proceedings of the fifth Australian conference on mathematics and computers in sport. Sydney, NSW: Sydney University of Technology, 2000.
16. Lemmer, Hermanus H. "The combined bowling rate as a measure of bowling performance in cricket." South African Journal for Research in Sport, Physical Education and Recreation 24.2 (2002): 37-44.
17. Barr, G. D. I., C. G. Holdsworth, and B. S. Kantor. "Evaluating performances at the 2007 cricket world cup." South African Statistical Journal 42.2 (2008): 125.
18. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12.Oct (2011): 2825-2830.