Dynamic Winner Prediction in Twenty20 Cricket: Based on Relative Team Strengths

Sasank Viswanadha[1], Kaustubh Sivalenka[1], Madan Gopal Jhawar[2⋆], and Vikram Pudi[3]

[1] Mahindra École Centrale, Hyderabad, India,
sasank14168@mechyd.ac.in, kaustubh14141@mechyd.ac.in
[2] Microsoft, India
majhawar@microsoft.com
[3] DSAC, Kohli Centre on Intelligent Systems, IIIT Hyderabad, India
vikram@iiit.ac.in

Abstract. Predicting the outcome of a match has always been at the center of sports analytics. Indian Premier League (IPL), a professional Twenty20 (T20) cricket league in India, has established itself as one of the biggest tournaments in cricket history. In this paper, we propose a model to predict the winner at the end of each over in the second innings of an IPL cricket match. Our methodology not only incorporates the dynamically updating game context as the game progresses, but also includes the relative strength between the two teams playing the match. Estimating the relative strength between two teams involves modeling the individual participating players' potentials. To model a player, we use his career as well as recent performance statistics. Using the various dynamic features, we evaluate several supervised learning algorithms to predict the winner of the match. Finally, using the *Random Forest Classifier* (RFC), we have achieved an accuracy of 65.79% - 84.15% over the course of second innings, with an overall accuracy of 75.68%.

Keywords: Winner Prediction, Sports Analytics, Supervised Learning, Player Modeling, Cricket

1 Introduction

The use of statistical analysis in sports has been growing rapidly since the past decade. It has not only changed the way game strategies are formed or the players are evaluated, but also has impacted the way sports is viewed by the audience. Cricket is one of the most followed team games in the world with billions of fans all across the globe. The complex rules governing the game along with many other player-dependent and natural parameters provide ample opportunities to model the game from various perspectives.

Cricket has evolved over time. Today, it is played in three major formats – Test Matches, One Day Internationals (ODIs) and the T20 cricket. T20 cricket

⋆ This work was done when the author was a student at IIIT-Hyderabad.

is the latest and the most exciting format of the game. Ever since its inception in 2007, IPL has been a huge success and has generated a billion-dollar industry. It is played during April and May of every year by teams representing Indian cities and has already completed 10 successful seasons. Therefore, in this paper, we focus our study on the IPL cricket matches. We propose a dynamic model to predict the winner of a match at the end of each over in the second innings of the match. Apart from various game dependent features such as the number of balls remaining, the number of runs to be scored remaining, and the number of wickets remaining, we have used the relative team strength between the competing teams as a distinctive feature in predicting the winner of the match. A team is composed of players, hence, estimating the relative team strength between two competing teams requires us to estimate the potential of the players. Therefore, using the recent and career performance statistics of a player we define novel methods to render his batting and bowling capabilities, the two major roles of a player in the game of cricket. Using these features, we have evaluated various supervised learning algorithms to predict the winner of the match at the end of each over as the match progresses.

2 Related Work

Over the last decade, the application of statistical methods in cricket analysis has manifolded, particularly in the context of winner prediction. The application of supervised learning techniques – Support Vector Machines (SVM) and Naive-Bayes Classification towards predictive analysis, considering various factors such as coin toss outcome, competing teams, home venue etc., in ODI matches is presented in Khan, Mehvish, Riddhi Shah [6]. Kaluarachhi, Amal [7] studied the impact of several factors in predicting the outcome of ODI cricket matches using Bayesian classifiers. Madan Gopal Jhawar, Vikram Pudi [8] proposes an approach to predict the winner of ODI cricket matches based on the team composition of the competing teams. Deep C Prakash, C Patwardhan et al. [9] presents an approach of winner predictions for the ninth season of IPL, at the start of the season, by modeling the individual player strengths into cumulative batting and bowling scores.

However, the problem of winner prediction, while the game is in progress, has not been studied in detail. Shankarnarayanan et al. [11] considers both the historical data as well as instantaneous match states for ODI cricket to predict the match winner using nearest-neighbor clustering and linear regression algorithms. Shankarnarayanan et al.[11] introduces the idea of using segments to break down an innings and make predictions for each segment. Michael Bailey, Stephen R. Clarke [12] studied a range of variables that could independently explain statistically significant proportions of variation associated with the predicted run totals and match outcomes were created. Further, they used a linear regression model to predict the winner.

3 Problem Formulation and Notation

3.1 Overview of T20 Cricket : Rules

In the T20 format of cricket, each of the two playing teams bats for a maximum of 120 deliveries and bowls for a maximum of 120 deliveries. The team that scores the maximum amount of runs in the 120 deliveries or before they lose their 10 wickets, wins the match

Over: A sequence of six balls bowled by a bowler from one end of the pitch is called an *over* in cricket terminology.

Innings: An *innings* is one of the divisions of a cricket match during which one team takes its turn to bat. There are two *innings* in a game of cricket. In this paper, we restrict our study to the second *innings* of a match.

State: In our study, we define *state* to represent the different stages in the match at which we make the predictions using our model. We consider 21 *states* for each match; 1 at the beginning of the second *innings* and 20 at the end of each *over* of the second *innings*. It is to be noted here that the number of *states* considered to make predictions can be changed.

3.2 Notation

In this section, we introduce the notation to be used throughout this paper. We use m to represent a match, $innings_1$ and $innings_2$ to denote the first and second innings respectively. We use $Team_A$ to represent the team batting in $innings_1$ and $Team_B$ to represent the team batting in $innings_2$. $Score_A$ denotes the *runs* scored by $Team_A$ in $innings_1$. $Target$ denotes the number of *runs* that $Team_B$ needs to score to win the match, $Target = Score_A + 1$. S_i, $0 \leq i \leq 20$ represents the *states* in a m. S_0 corresponds to the *state* at the end of $innings_1$ and the remaining *states* $1 \leq i \leq 20$ each correspond to the *state* at the end of *over* i in $innings_2$. S_{20} has been considered for training examples so as to make sure that the model learns which team has won the game. S_{20} has been used in the testing set as well and it serves as a confirmation that the model is working as expected. Pl_A^m denotes the set of 11 players in $Team_A$ playing in m and Pl_B^m denotes the set of 11 players in $Team_B$ playing in m.

$C(p)$ denotes the set of career statistics of a player p and $F(p)$ denotes the set of recent statistics (recent 4 games) or form of a player p. The career statistics are shown in Table 1 and recent statistics are similar to career statistics, replacing C with F.

At each state, there are 3 parameters along with the relative team scores that we use in our model to make predictions.

- $R^i_{runs\_remaining}$ denotes the number of runs $Team_B$ needs to get to win the m at *state i*. $R^i_{runs\_remaining} = Target - runs\ scored\ by\ Team_B\ at\ state\ i$
- $R^i_{wickets\_remaining}$ denotes the number of wickets $Team_B$ has in hand at *state i*. $R^i_{wickets\_remaining} = 10 - wickets\ lost\ by\ Team_B\ at\ state\ i$
- $R^i_{ball\_remaining}$ denotes the number of balls $Team_B$ is yet to play at *state i*. $R^i_{balls\_remaining} = 120 - balls\ played\ by\ Team_B\ at\ state\ i$

Table 1. Career Statistics

Notation	Description
MP_C	# Matches Played by the player
BaI_C	# Matches in which the player has batted
RS_C	# Runs Scored by the player
OB_C	# Overs in which the player has batted
NO_C	# The player remained $not - out$
Ba_C	# Average Runs scored by the player before getting out
$BaSR_C$	# Average runs scored by the player per 100 balls
BoI_C	# Matches in which the player has bowled
WT_C	# Wickets taken by the player
RC_C	# Runs conceded by the player
OB_C	# Overs in which the player has bowled
BE_C	# Runs conceded by the player per over
$BoSR_C$	# Balls bowled by the player per $wicket\_taken$

4 Methodology

4.1 Batsman Rating

Calculation of Batting Average: Batting Average is defined as the average number of *runs* scored by the batsman before he gets *out*. Batting average for the career statistics is calculated in the following way

$$Ba_C = \frac{RS_C}{BaI_C - NO_C}, \tag{1}$$

Calculation of Batting Strike Rate: Batting Strike Rate is defined as the average number of *runs* scored by the batsman before per 100 balls faced. Batting strike rate for the career statistics is calculated in the following way

$$BaSR_C = \frac{RS_C}{(OB_C * 6)} * 100 \tag{2}$$

The batting average and strike rate for the recent statistics is calculated similar to equations 1 and 2.

Calculation of Batsman Score The quality of the batsmen a team possesses can greatly affect the outcome of a game. Consistency and fast run-scoring ability are two traits common to all the good batsmen. Batting average and is a measure of the consistency of the batsman and batting strike rate is a measure of his fast run-scoring ability. As illustrated in [10], $batting\_average$, $batting\_strike\_rate$ can be used to effectively estimate the batting scores of participating players.

Career and recent scores of a player are calculated as shown in equations 3 and 4.

$$\phi^p_{career\_batting\_score} = \sqrt{\frac{BaI_C}{MP_C}} * Ba_C * BaSR_C \tag{3}$$

$$\phi^p_{recent\_batting\_score} = \sqrt{\frac{BaI_F}{n}} * Ba_F * BSR_F \tag{4}$$

The final batting score $\phi^p_{final\_batting\_score}$ of a player considering his career and recent statistics is given by the equation 5

$$\phi^p_{final\_batting\_score} = \mu * \phi^p_{career\_batting\_score} + (1 - \mu) * \phi^p_{recent\_batting\_score} \tag{5}$$

where n represents the number of recent matches considered and μ represents the weight assigned to the career score in calculating the final batting score of a player.

4.2 Bowler Rating

Calculation of Bowling Average: Bowling Economy is defined as the average number of *runs* conceded by the bowler per *over* he bowls. Bowler average for the career statistics is calculated in the following way

$$BE_C = \frac{RC_C}{OB_C} \tag{6}$$

Calculation of Bowling Strike Rate: Bowling Strike Rate is defined as the average number of *balls* bowled by the bowler per wicket taken. Bowling strike rate for the career statistics is calculated in the following way

$$BoSR_C = \frac{(OB_C * 6)}{WT_C} \tag{7}$$

The bowling average and strike rate for the recent statistics is calculated similar to equations 6 and 7.

Calculation of Bowler Score The quality of the bowlers a team possesses also has significant impact on the game's outcome. Economical bowling and high wicket-taking ability are two traits common to all the good bowlers. Bowling economy and bowling strike rate is a measure of the economical bowling while bowling strike rate is a measure of the bowler's high wicket-taking ability.As illustrated in [10], *bowling_average* and *bowling_strike_rate* can be used to effectively estimate bowling scores of participating players. Career and recent scores of a player are calculated as shown in equations 8 and 9.

$$\phi^p_{career\_bowling\_score} = \sqrt{\frac{BoI_C}{MP_C}} * (\frac{1}{BE_C * BoSR_C}) \tag{8}$$

$$\phi^p_{recent\_bowling\_score} = \sqrt{\frac{BoI_F}{n}} * (\frac{1}{BE_F * BoSR_F}) \quad (9)$$

The final bowling score $\phi^p_{final\_bowling\_score}$ of a player considering his career and recent statistics is given by the equation 10

$$\phi^p_{final\_bowling\_score} = \mu * \phi^p_{career\_bowling\_score} + (1-\mu) * \phi^p_{recent\_bowling\_score} \quad (10)$$

where n represents the number of recent matches considered and μ represents the weight assigned to the career score in calculating the final batting score of a player, and are same as the ones introduced in Equations 4 and 5, respectively.

4.3 Calculation of Relative Team Strength

A team's batting and bowling strength will be a consolidated measure of the batting and bowling strengths of the 11 players playing in that match. Algorithm 1 illustrates the computation of $Relative\_strength_{Team_B/Team_A}$. Lines 1- 4 normalize the $\phi^p_{batting\_score}$ and $\phi^p_{bowling\_score}$ for all the players. As the match progresses through the $innings_2$, there is every possibility of some *batsmen* getting *out* and some the *bowlers* using up their quota of *deliveries* (24 balls). Thus, we compute the $\phi^{Team}_{batting\_score}$ and $\phi^{Team}_{batting\_score}$ of a team as a weighted sum of players (*batsmen*) who are not yet *out* and the players (*bowlers*) who still retain their quota of deliveries respectively, Lines 5- 8. This introduces dynamism in the team scores by removing players, who cannot contribute to the game any longer (in terms of batting and bowling), from the respective batting and bowling scores of the team. Line 9 computes the $Relative\_strength_{Team_B/Team_A}$. We only calculate $Relative\_strength$ with respect to $Team_B$ because $Team_B$ bats in $innings_2$ according to our notation and we make predictions only for $innings_2$ in our model. $Team_A$ bowling score has a negative impact on $Team_B$ batting score and vice-versa in the formula in line 9.

4.4 Features

To predict the outcome of an ongoing T20 (IPL) match we first split the $innings_2$ into 21 states. S_0 at the end of $innings_1$ and $S_i (1 \leq i \leq 20)$ at the end of i overs. At a state S_i we use the following dynamic features to make the prediction:

- Runs remaining to be scored to win the match $R^i_{runs\_remaining}$
- Wickets that $Team_B$ still has in hand $R^i_{wickets\_remaining}$
- Balls remaining to be played by $Team_B$ in the innings $R^i_{ball\_remaining}$
- $Relative\_strength_{Team_B/Team_A}$; serving as a dynamic metric of team strengths to better forecast predictions.

The aforementioned features capture the state of the match at any given instance while the match is in progress and these features change as the match progresses towards completion. All these features are parsed to a classifier along with the label (1 if $Team_B$ wins, 0 otherwise) to forecast predictions for the match winner. Venue or home advantage is not used as a feature because most of the pitches in IPL are somewhat similar and the crowd support is even. Also, the shorter format of the game makes this feature negligible.

Algorithm 1 Modeling Teams at the beginning of an over

Input: Pl_A^m, Pl_B^m,$\phi^p_{batting\_score}$, $\phi^p_{bowling\_score} \forall p \in (Pl_A^m, Pl_B^m)$
Output: $Relative\_strength_{Team_B/Team_A}$

1: **for** $p \in (Pl_A^m \cup Pl_B^m)$ **do**
2: $\quad \phi^p_{batting\_score} \leftarrow \frac{\phi^p_{batting\_score}}{max(\phi^p_{batting\_score})}$
3: $\quad \phi^p_{bowling\_score} \leftarrow \frac{\phi^p_{bowling\_score}}{max(\phi^p_{bowling\_score})}$
4: **end for**
5: $\phi^{Team_A}_{batting\_score} = \sum_{p \in (Pl_A^m)} \phi^p_{batting\_score}$
6: $\phi^{Team_A}_{bowling\_score} = \sum_{p \in (Pl_A^m)} (\frac{24 - balls\_bowled}{24}) * \phi^p_{bowling\_score}$
7: $\phi^{Team_B}_{batting\_score} = \sum_{p \in (Pl_B^m (not\ yet\ out))} \phi^p_{batting\_score}$
8: $\phi^{Team_B}_{bowling\_score} = \sum_{p \in (Pl_B^m)} \phi^p_{bowling\_score}$
9: $Relative\_strength_{Team_B/Team_A} = \frac{\phi^{Team_B}_{batting\_score}}{\phi^{Team_A}_{bowling\_score}} - \frac{\phi^{Team_A}_{batting\_score}}{\phi^{Team_B}_{bowling\_score}}$

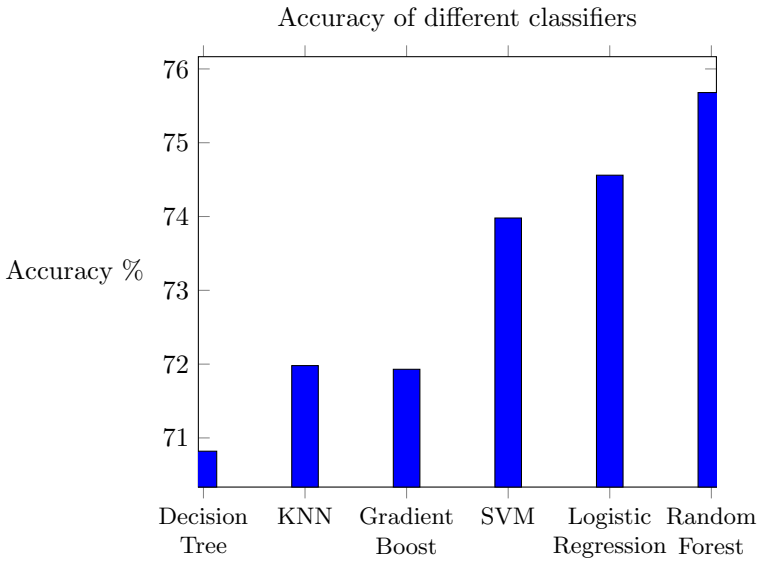
5 Experiments and Results

5.1 Dataset

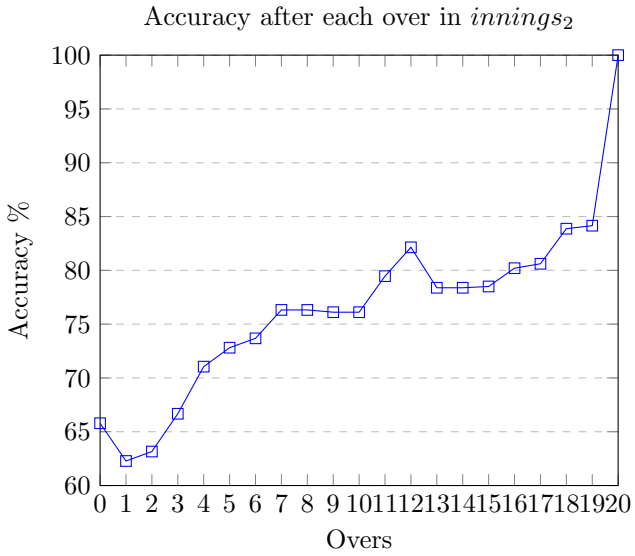
The dataset can be broadly divided into two categories – *historical data*: pertaining to the career statistics of players, and *ball by ball data*: pertaining to various states of a match. The dataset for *career statistics* has been scraped from the *cricinfo* website [13] for all the matches played in the seasons 3-10 of IPL. The *ball by ball* data for each match in seasons 3-10 of IPL has been provided by the *cricsheet* website [14]. The dataset constitutes the match statistics recorded after each *ball*, including *runs scored*, *wickets lost*, *current batsmen*, *current bowler*, *winner of the match*, *date of fixture*, etc. We combined data from both these sources to build our prediction model. IPL seasons 3-7 have been used for training our model, season 8 has been used for validating the parameters, and seasons 9 and 10 have been used as the test data, where each season consists of 59 matches.

5.2 Learning Parameters

To learn the values of the parameters n and μ, used in Equations 4, 5, 9 and 10, we used all the matches played in seasons 3-8. We did a grid search over the values of n and μ. For each combination of n and μ, we ranked the players in the order of their estimated batting scores and estimated bowling scores, which are compared against the actual ranked batting and bowling score lists in the last n matches. Finally, $n = 4$ and $\mu = 0.8$ yielded the *least squared error* in terms of the rank difference between estimated and the actual lists.



5.3 Results



Using the features described and match outcome as the label, we evaluated various binary classifiers such as SVM, Random Forests, k Nearest Neighbors (kNN), Logistic Regression and Decision Trees using their scikit-learn [15] implementations. The *ParameterGrid* mechanism has been used to evaluate all possible combinations of parameters for all the above listed algorithms. Figure 5.2 shows the accuracies of the different classifiers. The small differences in

the accuracy of the different classifiers suggests that the predictive power lies in the features and not the classifer used. The Random Forest algorithm with parameters: $n\_estimators = 28$, has yielded the highest accuracy, for the validation set, among the best models for all other classifiers. The results for this are shown in Figure 5.3.

From the plot in Figure 5.3, we observe an increasing trend in prediction accuracies after each over as the match progresses until completion. This proves the ability of our classifier to predict the winner with increasing confidence after each over. This also agrees with common intuition, that as the game nears its end, it is easier to predict the winner based on a given match state. While we examine the increasing trend of prediction accuracies in Figure 5.3, some fluctuations are observed around the middle overs. This is because a game need not necessarily progress with increasing chances of one team's victory. Most of the times, the game fluctuates between both the team based on their very recent (last few overs) performance in the match. However, when generalized over a set of matches, the probability of accurately predicting the winner increases as the game progresses towards its end.

The overall prediction accuracy obtained regardless of the match state is 75.68%, with an accuracy of 65.79% at the beginning of the second innings which increases to 84.15% at the end of the 19th over.

There have been several works such as [9], [11], etc., specifically addressing the problem of winner prediction in ODI and Twenty20 cricket. However, our study cannot be directly compared to them as we consider our analysis only from the beginning of the $innings_2$ and our model cannot be translated into their works for comparison. Nevertheless, table 2 briefs about some of the previous works and their stated accuracies.

Table 2. Various Winner Prediction Models in Cricket

Author	**Description**	**Accuracy**
[9]	Winner prediction for IPL Season 9 (2016), at the start of the season	69.64%
[11]	A dynamic winner prediction model for ODIs, January 2011 to July 2012	68%-70%
Baseline model	Only $\#runs\_remaining$, $\#wickets\_remaining$, and $\#balls\_remaining$ used as features	69.37%
Our model	Dynamic winner prediction in IPL matches, For seasons 3-10 (2010-2017)	**75.68%**

The accuracy of Our model is greater than the accuracy of Baseline model in Table 2. This shows the significance of $Relative\_strength_{Team_B/Team_A}$ as a feature for making robust predictions.

6 Conclusion and Future Work

The problem of dynamic winner prediction in a Twenty20 cricket match has been successfully addressed in this paper. A combination of features which capture the state of the match have furnished promising results. $Relative\_strength_{Team_B/Team_A}$

has been shown as an important feature that is successful in quantifying and comparing the strengths of the playing teams. In order to further make the prediction model adept at addressing the entire match scenario, we intend to extend our approach in order to account for the $innings_1$ dynamics as well. The primary challange that stands in the way of this is to estimate the score that the team batting first is expected to score.

References

1. Duckworth, Frank C., 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, Tim B. Swartz.: The best batsmen and bowlers in one-day cricket. South African Statistical Journal 37.2 (2003): 203.
3. Kimber, Alan.: A graphical display for comparing bowlers in cricket. Teaching Statistics 15.3 (1993): 84-86.
4. Van Staden, Paul Jacobus.: Comparison of cricketers bowling and batting performances using graphical displays. (2009).
5. 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).
6. Khan, Mehvish, Riddhi Shah.: Role of External Factors on Outcome of a One Day International Cricket (ODI) Match and Predictive Analysis."
7. Kaluarachchi, Amal, Varde Aparna S.: CricAI: A classification based tool to predict the outcome in ODI cricket. 2010 Fifth International Conference on Information and Automation for Sustainability. IEEE, 2010.
8. Madan Gopal Jhawar, Vikram Pudi.: "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach." European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2016 2016), September 2016 , Conference Center, Riva del Garda. Report no: IIIT/TR/2016/32
9. Deep C Prakash, C Patvardhan, Vasantha C Lakshmi.: "Data Analytics based Deep Mayo Predictor for IPL-9". International Journal of Computer Applications 152(6):6-11, October 2016.
10. Barr, G. D. I., 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
11. Sankaranarayanan, Vignesh Veppur, Junaed Sattar, Laks VS Lakshmanan.: Autoplay: A Data Mining Approach to ODI Cricket Simulation and Prediction. SDM. 2014.
12. Michael Bailey, Stephen R. Clarke.: "Predicting the match outcome in One Day International cricket matches, while the game is in progress." The 8th Australasian Conference on Mathematic s and Computers in Sport, 3-5 July 2006, Queensland, Australia, 5 December 2006.
13. ESPN Cricinfo: http://www.espncricinfo.com
14. IPL data: http://cricsheet.org
15. Pedregosa, Fabian, et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12.Oct (2011): 2825-2830.