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

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

[1] Mahindra École Centrale, Hyderabad, India,
sasank14168@mechyd.ac.in, kaustubh14141@mechyd.ac.in
[2] DSAC, Kohli Centre on Intelligent Systems, IIIT Hyderabad, India
madangopal.jhanwar@research.iiit.ac.in, vikram@iiit.ac.in

Abstract. Cricket as a sport has evolved over years into various formats the latest addition being Twenty20 (T20) cricket, in 2007, which is the shortest and the most dynamic of all forms of the sport. Due to the short nature of a T20 game, the match dynamics can change unexpectedly, making the problem of predicting the winner quite challenging and interesting. In this article, we address the problem of predicting the outcome of IPL matches, as the game progresses dynamically using supervised learning approach based on the relative team strengths. The problem of computing relative team strength reduces to modeling and computing the strengths of individual players in a team. We use the career statistics as well as the form (performances in recent games) to compute the batting and bowling strengths of a player, at various stages of the game. We use these statistics along with the progress of the game to construct dynamic features to predict the game outcome, during the second innings. A Random Forest classifier using our constructed features for winner prediction yields accuracies of upto 76% regardless of the states considered. This is a significant improvement over state-of-the-art approaches that achieve upto 70% using only static or a mix of static and dynamic features.

Keywords: Winner Prediction, Sports Analytics, Supervised Learning, Player Modeling, Random Forest, Ensemble Classification, Cricket

1 Introduction

The popularity of data driven decision-making in sports has been on the rise since the advent of statistical modeling in sports analytics. Basketball is one of the best examples of how analytics have changed the way sports are played and player performance is measured. Although cricket is one of the most popular sports on the planet, it falls behind other major sports like basketball and baseball when it comes to using of sports analytics for on-field decision making and prediction. Data-driven decision making and analysis of players in cricket can help teams make better choices regarding player selections and also make the sport more exciting for its viewers.

Today, cricket is played in three formats namely; Test cricket, the longest form of the game, One Day International and T20 format (20 overs per team), which is the shortest. T20 cricket is the most exciting format of cricket. Its popularity and viewer-ship have increased widely since its inception. The Indian Premier League (IPL), founded by Board of Control for Cricket in India (BCCI) is a professional Twenty20 cricket league in India contested during April and May of every year by teams representing Indian cities. The IPL is the most-attended cricket league in the world and ranks sixth among all sports leagues. Our research in this article is focused on Twenty20 cricket, specifically the Indian Premier League but the methodology could be easily extended to all forms of cricket.

To predict the outcome of the Twenty20 cricket matches, we consider our analysis from the beginning of the second innings. The second innings is divided into segments of equal length (measured by number of balls bowled). We propose a method where we estimate the batting and bowling strengths of each of the 22 players. These player strengths are then used to calculate the relative team strength of the team batting second. This relative strength along with features such as runs remaining, balls remaining and wickets remaining are used as features in supervised learning to predict the winner of the game, after each segment.

The key contributions of this paper are :

- We propose a dynamic (first of its kind) approach to predict the winner of Twenty20 cricket matches in the second innings of the match, as the game progresses.
- We use a team composition based approach which largely depends on the strengths of individual players.
- We also discuss a set of dynamic features which prove to be important while trying to analyse a match which is in progress.

2 Literature Survey

In recent times, the application of statistical analysis techniques has been quite extensive for the sport of cricket particularly in the context of player rating and squad optimization. [1] defines and discusses the manner in which, targets can be modified in rain interrupted games. Taking into account the D-L *resources* defined in [1], [2] discusses best bowlers and batsmen in ODI cricket. A more visual comparison of player strengths are put forth in [3] and [4]. The approach of player modeling considering the strength of opposition, venue of the match along with other factors was in discussed in [5]. The application of supervised learning techniques; Support Vector Machines (SVM) and Naive-Bayes Classification towards predictive analysis considering various factors such as toss, competing teams, home venue etc., for the winner prediction in an ODI match was presented in [6].

However, there have been very few efforts in addressing the problem of winner prediction while a cricket match is in progress; one of them; [7] considers both

the historical data along with instantaneous match state for One Day Internationals (ODI),so as to predict the match winner making use of nearest-neighbor clustering and linear regression algorithms. [7] also introduces the idea of using segments to break down an innings and makes predictions for each segment. We have incorporated this idea into our study. [8] puts forth a novel approach based on team composition, computing relative team strengths based on the team's cumulative batting and bowling strengths in order to predict the winner of an ODI match. In addition, [8] also showed that it is likely the team composition keeps changing with every match and that it is important to consider the players playing in every game instead of taking only the statistics of the team as a whole like [6] and [9] did. This idea has been incorporated in our study and forms an important part of our approach. Taking cues from the concepts presented in [10], we have developed the formulae for calculating the *batsman_score* and the *bowler_score*. [11] presents an approach of winner prediction for the ninth season of IPL, by modeling the individual player strengths into cumulative batting and bowling scores. In this paper, we aim to address the problem of dynamic winner prediction in a Twenty20 cricket match based on *team_strengths*, a study first-of-its-kind in Twenty20 cricket.

3 Problem Formulation and Notation

3.1 Overview of Twenty20 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 *match* 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 *match*. 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 $innings_2$. $Players(match, Team_A)$ denotes the set of 11 players in $Team_A$ playing in *match* and $Players(match, Team_B)$ denotes the set of 11 players in $Team_B$ playing in *match*.

Table 1. Career Statistics

Notation	Description
$C_{matches\_played}$	# Matches Played by the player
$C_{batting\_innings}$	# Matches in which the player has batted
C_{runs_scored}	# Runs Scored by the player
C_{overs_batted}	# Overs in which the player has batted
C_{not_outs}	# The player remained $not - out$
$C_{batting\_average}$	# Average Runs scored by the player before getting *out*
$C_{batting\_strike\_rate}$	# Average runs scored by the player per 100 balls
$C_{bowling\_innings}$	# Matches in which the player has bowled
$C_{wickets\_taken}$	# Wickets taken by the player
$C_{runs\_conceded}$	# Runs conceded by the player
C_{overs_bowled}	# Overs in which the player has bowled
$C_{bowling\_economy}$	# Runs conceded by the player per over
$C_{bowling\_strike\_rate}$	# Balls bowled by the player per *wicket_taken*

$C(p)$ 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 recent 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 *match* 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 $Team_B$ yet to play at *state* i. $R^i_{balls\_remaining} = 120 - balls\ played\ by\ Team_B\ at\ state\ i$

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

$$C_{batting\_average} = \frac{C_{runs\_scored}}{C_{batting\_innings} - C_{not\_outs}}, \tag{1}$$

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

$$C_{batting\_strike\_rate} = \frac{C_{runs\_scored}}{(C_{overs\_batted} * 6)} * 100 \tag{2}$$

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.

$$\phi^p_{career\_batting\_score} = \sqrt{\frac{C_{batting\_innings}}{C_{matches\_played}}} * C_{batting\_average} * C_{batting\_strike\_rate} \tag{3}$$

$$\phi^p_{recent\_batting\_score} = \sqrt{\frac{F_{batting\_innings}}{4}} * F_{batting\_average} * F_{batting\_strike\_rate} \tag{4}$$

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

$$\phi^p_{final\_batting\_score} = 0.8 * \phi^p_{career\_batting\_score} + 0.2 * \phi^p_{recent\_batting\_score} \tag{5}$$

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

$$C_{bowling\_economy} = \frac{C_{runs\_conceded}}{C_{overs\_bowled}} \tag{6}$$

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

$$C_{bowling\_strike\_rate} = \frac{(C_{overs\_bowled} * 6)}{C_{wickets\_taken}} \tag{7}$$

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.

$$\phi^p_{career\_bowling\_score} = \sqrt{\frac{C_{bowling\_innings}}{C_{matches\_played}}} * (\frac{1}{C_{bowling\_economy} * C_{bowling\_strike\_rate}}) \tag{8}$$

$$\phi^p_{recent\_bowling\_score} = \sqrt{\frac{F_{bowling\_innings}}{4}} * (\frac{1}{F_{bowling\_economy} * F_{bowling\_strike\_rate}} \tag{9}$$

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

$$\phi^p_{final\_bowling\_score} = 0.8 * \phi^p_{career\_bowling\_score} + 0.2 * \phi^p_{recent\_bowling\_score} \tag{10}$$

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 $Relative\_strength_{Team_B/Team_A}$. 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.

Algorithm 1 Modeling Team

Input: $Players(match, Team_A)$, $Players(match, Team_B)$,$\phi^p_{batting\_score}$, $\phi^p_{bowling\_score}$ $\forall p \in (Players(match, Team_A),\ Players(match, Team_B))$

Output: $Relative\_strength_{Team_B/Team_A}$

1: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **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: $\phi^{Team_A}_{batting\_score} = \sum_{p \in (Players(match, Team_A)} \phi^p_{batting\_score}$

6: $\phi^{Team_A}_{bowling\_score} = \sum_{p \in (Players(match, Team_A)} (\frac{24 - balls\_bowled}{24}) * \phi^p_{bowling\_score}$

7: $\phi^{Team_B}_{batting\_score} = \sum_{p \in (Players(match, Team_B(yettobat))} \phi^p_{batting\_score}$

8: $\phi^{Team_B}_{bowling\_score} = \sum_{p \in (Players(match, Team_B)} \phi^p_{bowling\_score}$

9: $Relative\_strength_{Team_B/Team_A} = \frac{\phi^{Team_B}_{batting\_score}}{\phi^{Team_A}_{bowling\_score}} - \frac{\phi^{Team_A}_{batting\_score}}{\phi^{Team_B}_{bowling\_score}}$

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 $R^i_{ball\_remaining}$
- $RelativeTeamStrength_{Team_B/Team_A}$; 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.

5 Experiments and Results

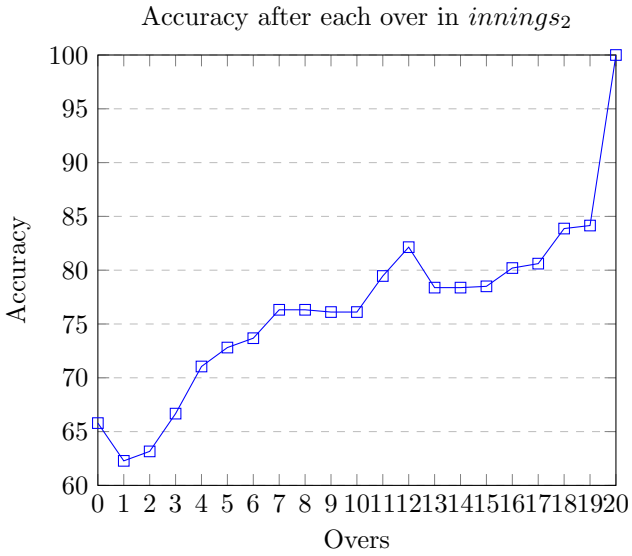
5.1 Datasets

The data used for this study can be categorized into: *historical data* pertaining to the career statistics and *ball by ball data* pertaining to various states during a

match. The dataset for *career statistics* was scraped from the *cricinfo* website [12] for seasons 3-10 of IPL. Similarly, for the *ball by ball* data for each match in seasons 3-10 of IPL, the dataset was scraped from the *cricsheet* website [13]. The dataset constitutes the match dynamics recorded after each *ball* such as; *runsscored*, *wicket lost*, *current batsmen*, *current bowler* along with other useful information like *winner of the match*, using which training data is labeled, *date of fixture* and *toss outcome*. We combined data from both these sources to make predictions using our model. IPL seasons 3-7 were used for training our model, season 8 for validation of parameters and seasons 9 and 10 were used for testing the accuracy of our model.

5.2 Weights

To select the weights used in equations 3 and 4, and the number of previous matches to be considered for the form (recent statistics) of a player, we used the data from seasons 3-8. Using exhaustive experimentation, we compared the estimated rankings of the batsman and bowlers before the start of a match (using the selected values of the two parameters) with the actual player rankings in that match. This process was repeated for all the matches in seasons 3-8. The minimum difference between the predicted and actual rankings was achieved when the previous 4 matches played a player were considered and the weight used in equations 3 and 4 equaled 0.8.



5.3 Results

Using the features listed in the previous sections along with the match outcome as label, we assessed various binary classifiers such as SVM, Random Forests,

kNN, Logistic Regression and Decision Trees through their scikit-learn [14] implementations. The *ParameterGrid* mechanism has been used to evaluate all possible combinations of parameters for all the above listed algorithms. The Random Forest algorithm with parameters: $n\_estimators = 28$, has yielded highest accuracy among the best models for all other classifiers.

From the plot in 5.2, 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 5.2, some fluctuations are observed around the middle overs which are due to the inherent unpredictability in the middle overs of a Twenty20 game because it has the potential to change the game outcome very quickly.

The overall prediction accuracy obtained regardless of the number of defined states is 75.68%. To the best our knowledge, this is the *highest* recorded accuracy reported for Twenty20 cricket. Our model started off at an accuracy 65.79% at the beginning of the second innings increased to 84.15% at the end of the 19th over.

There have already been a few works such as [7], [6] and [9] specifically addressing winner prediction problem in ODI and Twenty20 cricket. However, our study cannot be directly compared to the works stated above as we consider our analysis only from the beginning of the $innings_2$ and our model cannot be transformed to accommodate the analyses presented in the works above. Nevertheless, table 2 shows the accuracies of several different models.

- Model 1: Our model without the $Relative\_strength_{Team_B/Team_A}$ feature. This shows the significance of $Relative\_strength_{Team_B/Team_A}$ as a feature for making robust predictions.
- Model 2: model presented by [11] for IPL Season.
- Model 3 : model presented by [7] for ODI's.

Table 2. Comparison

Model	Accuracy
$OurModel$	75.68%
$Model1$	69.37%
$Model2$	69.64%
$Model3$	68-70%

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.

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. 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, and Riddhi Shah. Role of External Factors on Outcome of a One Day International Cricket (ODI) Match and Predictive Analysis."
7. Sankaranarayanan, Vignesh Veppur, Junaed Sattar, and Laks VS Lakshmanan. Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction. SDM. 2014.
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. 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.
10. 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
11. Deep C Prakash, C Patvardhan and Vasantha C Lakshmi. "Data Analytics based Deep Mayo Predictor for IPL-9". International Journal of Computer Applications 152(6):6-11, October 2016.
12. ESPN Cricinfo, http://www.espncricinfo.com
13. IPL data, http://cricsheet.org
14. Pedregosa, Fabian, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12.Oct (2011): 2825-2830.