Towards a time-lapse prediction system for cricket matches

by

Vignesh Veppur Sankaranarayanan

B. E., Anna University, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

May 2014

Abstract

Cricket is a popular sport played in over a hundred countries, is the second most watched sport in the world after soccer, and enjoys a multi-million dollar industry. There is tremendous interest in simulating cricket and more importantly in predicting the outcome of games, particularly in their one-day international format. The complex rules governing the game, along with the numerous natural phenomena affecting the outcome of a cricket match present significant challenges for accurate prediction. Multiple diverse parameters, including but not limited to cricketing skills and performances, match venues and even weather conditions can significantly affect the outcome of a game. The sheer number of parameters, along with their interdependence and variance create a non-trivial challenge to create an accurate quantitative model of a game. Unlike other sports such as basketball and baseball which are well researched from a sports analytics perspective, for cricket, these tasks have yet to be investigated in depth. The goal of this work is to predict the game progression and winner of a yet-to-begin or an ongoing game. The game is modeled using a subset of match parameters, using a combination of linear regression and nearest-neighbor classification-aided attribute bagging algorithm. The prediction system takes in historical match data as well as the instantaneous state of a match, and predicts the score at key points in the future, culminating in a prediction of victory or loss. Runs scored at the end of an innings, the key factor in determining the winner, are predicted at various points in the game. Our experiments based on actual cricket game data, shows that our method predicts the winner with an accuracy of approximately 70%.

Preface

The work presented in this dissertation was conducted in the Data Management and Mining lab under the supervision of Prof. Laks V.S.Lakshmanan and in collaboration with Dr.Junaed Sattar. I was the lead investigator in this work responsible for high level problem identification, data collection, methodology, analysis of results and manuscript composition. Dr. Junaed Sattar and Prof. Laks V.S.Lakshmanan were closely involved throughout and provided advice on all of the core aspects like formalization of the problem, methodology, and analysis and interpretation of results and also helped in manuscript composition and editing. Prof. Jim Little acted as a second reader and provided constructive feedback on the manuscript.

A jointly authored paper based on this work has been published in the conference proceedings of 2014 SIAM Conference on Data Mining [28]

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisor Prof.Laks V.S.Lakshmanan and co-supervisor Dr.Junaed Sattar for their support, valuable guidance and encouragement. I would like to acknowledge NSERC for funding this research.

I want to thank my parents for their sincere love and support through this journey. Words cannot do justice to their contribution. I would like to thank the stimulating environment of UBC Computer Science, Waran Research Foundation – specifically Prof.Venkateswaran, my friends, the special *one* and all others who have been part of this journey.

Last but not the least, I thank the almighty for providing me with the above!

Dedication

..to my over curious and naughty brother Prasanna..

Chapter 1

Introduction

Predicting game progression and its outcome has direct applications in devising strategies aimed at winning a game. It is also used to set odds in the betting industry. Despite the popularity of the sport, prediction in cricket has not been addressed in great detail as in other sports like baseball, basketball, soccer etc. This work addresses the problem of predicting game progression and outcome of One-Day-International cricket matches. This chapter provides the motivation for this work.

1.1 Importance of Sports Analytics

In today's world, professional sports are intensely competitive propositions. Motivated by prestige and huge financial rewards, sports professionals are engaged in fierce competition not only on the sporting turf, but also off it, pursuing perfection and the slightest of advantages over their opponents. Today's sports professionals include not only the sportsmen actively participating in the game, but also their coaches, trainers, physiotherapists, and in many cases, strategists. Coaches, captains and team managers leverage their expertise and make decision using their intuition. Such decisions can be biased by the human impressions and judgments of players and hence might overlook players' weakness. Moreover, interesting patterns in the game may elude the eye of the best tactician.

With massive advances in cheaper and reliable storage technologies, data about

every match is being stored in such a way that the entire chain of events could be replayed. With such advancements in technology and huge stake involved in sporting events, the in-game data recorded is analyzed and converted into actionable knowledge by teams to gain advantage over their competitors. The trend noticeable both in individual sports such as tennis and in team sports such as baseball and basketball is that this knowledge is used to determine pre-game strategy. Successful application of this pre-determined strategy often becomes the decisive factor towards victory. The difference between a win and a loss hinges on formulation of such strategies and extensive planning.

Effective formulation of strategies requires carrying out extensive analysis of past games, current performance in the game in progress, and numerous other factors affecting a game. Players and team management (collectively often referred to as the team *think-tank* in sports) perform as a "human expert system", relying on experiences, expertise and analytic ability to arrive at the best-possible course of action before as well as during a game. Vast amount of raw data and statistics are available to aid in the decision-making process, but determining what it takes to win a game is extremely challenging.

These amounts of raw data are also leveraged by broadcasters and game experts to analyze performances of individual players, their strengths, weaknesses and teams' performance. These facts and figures are presented to the viewers to add richness to the viewer experience.

Betting adds an extra dimension for the sport industry. Billions of dollars are being wagered on sporting events [21]. Gambling houses rely on statistical models that predict outcomes of various level of sporting events.

1.2 Cricket – Popularity & Formats

Cricket is played in more than 100 countries across the world[1]. However, focus areas, namely the Indian subcontinent, United Kingdom, Australia, South Africa and the Caribbean drive the revenues and commercial interest in the sport. This is attributed to the fact that these are the top performing teams and form the core members of International Cricket Council, the cricket governing body. It has the

[1] http://www.espncricinfo.com/ci-icc/content/story/209608.html

second largest viewership by population for any sport, next only to soccer, and generates an extremely passionate following among the supporters. It is also the fourth highest sport to be bet on, after Soccer, Tennis and snooker[21].

Cricket is a team sport played between bat and ball, and governed by an extensive and complex set of rules. The eventual goal of the game is for one team to score more *runs* than their opponents to be declared the winner. A pair of *batsmen* from a team are in the field at any given time, trying to score runs off the *balls* or *deliveries* thrown at them by the *bowlers* of the opposing team. The goal of the bowlers is to get all the batsmen out before they can accumulate a large score. The *fielders* (from the bowling team) assist the bowlers by stopping or catching the balls after they are hit by the batsmen, to stop them from scoring runs and getting them out, respectively. Once all the batsmen are out, it is the turn of the previous *bowling team* to bat, who must score more runs than their opponents to win. Each team gets a minimum of one turn, or *innings* in cricket parlance, depending on the format of the match. In a *Test* match scenario, the game is limited to at-most two innings each, held over a maximum of five days, whichever finishes first. Figure 1.1 shows an Indian batsman Sachin Tendulkar batting in a test match between India and Australia. The other two formats of cricket and arguably more popular ones are *One-Day International* (abbreviated as ODI) and *Twenty20* formats. In One-Day International matches (object of analysis in this work), each team gets to bat once, with an innings consisting of 50 *overs*, with each over being a collection of six legal balls *bowled* by a bowler. A ball is considered to be illegal if the bowler crosses the crease while delivering the ball. It is also illegal if the balls lands and bounces more than once on the pitch before reaching the batsman. In these and few more circumstances, the ball is not counted and a penalty run is awarded to the batting team. *Twenty20* matches limits the game to 20 overs per side, shortening the duration to approximately three hours a game. The Test match setting is unique in the sense that it allows for a game to be drawn, whereas the concept of a draw does not exist in the other two formats. If the two teams score the same number of runs using the resources allocated, then the match is said to be tied. only *Twenty20* format has a tie-breaker to determine the winner after a tie. Among these three formats, the one-day format is the most popular. Motivated by this, we focus our analysis in this paper on one-day cricket. In figure 1.2, India and Australia

play an ODI match on 9th January 2004 at the MCG cricket ground in Melbourne, Australia.

Figure 1.1: A test format cricket match between India and Australia. Sachin Tendulkar plays the ball. (Image by Pulkit Sinha, released under the Creative Commons Attribution-ShareAlike 2.0 Generic License (CC BY-SA 2.0).

The team strategists often are faced with making decisions when the predetermined strategy fails or the game unravels in an unexpected manner. Currently, a combination of personal experience, team constitution and seat of the pants "cricketing sense" is relied upon for making instantaneous strategic decisions. Inherently, the methodology employed by human experts is to extract and leverage important information from both past and current game statistics. However, to our knowledge, the underlying science behind this has not been clearly articulated. One of the key problems that needs to be solved in formulating strategies is *predicting the outcome of a game*. The focus of this work is to address the problem of accurately modeling game progression towards match outcome prediction. Predicting game progression and outcome involve leveraging data mining and machine learning techniques to learn the patterns from historical play data. Various techniques like regression, nearest neighbors, clustering, attribute bagging etc. are customized and used in this work.

Figure 1.2: A ODI cricket match between India and Australia (Image by Ricky212, released under the Creative Commons Attribution-ShareAlike 2.0 Generic License (CC BY-SA 2.0).

1.3 Machine Learning Models

This section serves as an introduction to the machine learning models used in this work. K-nearest neighbor is a non-parametric method that is used for regression and classification [11]. A test sample along with a set of training samples are given as input to the model. The model identifies the test sample's k closest point(s) in the training sample and assigns class membership to the test sample. The intuition behind this model is that, given a test sample, information can be borrowed from training data by finding the closest k points. Spearman [29], Jaccard [15], Manhattan (L1 norm), Cosine [13], Euclidean (L2 norm) are some of the distance metrics used to calculate distance between two samples.

Attribute bagging [7] is an ensemble learning method that has l individual classifiers operating on n features chosen at random. Majority voting is performed

among the l classifiers to pick the output class. This ensemble method can use any classification learning model. The number of classifiers and the size of each bag (n) is experimentally determined. By definition, random forests [20] (which employs decision trees as classifier model) is considered to be a special case of attribute bagging method. As explained in section 4.8, attribute bagging along with nearest neighbor classification method is used to predict *home* runs - one of the two prediction components in this work.

K-means algorithm [5] is a clustering technique that partitions n samples into k clusters based on the distance of samples from the cluster centres. In this work, k-means algorithm is used to cluster batsmen into five groups based on their ability as inferred by their performance statistics (elaborated in section 4.5).

1.4 Contributions

In this work, a model for one-day format games is learnt by mining existing game data. In principle, our approach is applicable towards modeling any format of the game; however, we choose to focus our testing and evaluation on the most popular and arguably the most important format, namely one-day international (ODI), for the reasons mentioned above. By using a combination of supervised and unsupervised learning algorithms, our approach learns a number of features like home/-away, team performance in the past, batsmen, bowlers etc. from a one-day cricket dataset which consists of complete records of all games played in a 19-month period between January 2011 and July 2012. Along with these learned *historical features* of the game, our model also incorporates *instantaneous match state* data, such as runs scored, wickets lost etc., as game progresses, to predict future states of an on-going match. By using a weighted combination of both historical and instantaneous features, our approach is thus able to simulate and predict game progression before and during a match. A paper based on this work has been presented and published [28] at the *SIAM 2014 International Conference on Data Mining* held between April 24 - 26 in Philadelphia, USA.

1.5 Outline

The organization of rest of the thesis is as follows.

- An overview of existing literature in sports prediction and also specifically for cricket is provided in Chapter 2.
- Chapter 3 provides an overview of the one-day format and its basic rules.
- In Chapter 4, we introduce the problem of modeling cricket to the data mining community. A key part of this modeling is the identification of the most important features of the game (Section 4.3 and 4.4). Furthermore, it explains our algorithm in detail.
- In Chapter 5 we describe the challenges in extracting and cleaning historical match data so it can be used for model learning. We discuss the results of the extensive experimentation we conducted on a historical dataset we crawled and cleaned. Not only do our results validate our approach, but they also show it significantly outperforms the state of the art in one-day cricket game score and outcome prediction.
- Conclusion and directions of future work are presented in chapter 6.

Chapter 2

Related Work

As mentioned in Section 1.1 sports analytics has direct applications in understanding and improving team performance, betting industry and match analysis by broadcasters. The complete set of data recorded for every game is proprietary to the broadcaster. But a handful of high level play-by-play data is accessible to the public through websites like www.espncricinfo.com (cricket), www.mlb.com (baseball), www.nba.com (basketball) etc. In Section 2.1 below, we discuss some relevant work in the direction of modeling, simulation and prediction of sports.

2.1 Sports Prediction

The problem of outcome prediction has been investigated in the context of basketball, baseball and soccer.

2.1.1 Basketball

Vaz de Melo et al. [32] model the league championship as a network of players based on their work relationships over the years and predict the league standing at the end of a season. Fewell et al. [10] define players as nodes and ball passes as links and examine different network properties such as degree centrality, clustering, and entropy to analyze and quantify a teams' strategy. Bhandari et al. [4] developed the Advanced Scout system for discovering interesting patterns from basketball games, which has is now used by the NBA teams. The key idea is that

by using a technique called Attribute Focusing, an overall distribution of an attribute is compared with the distribution of this attribute for a subset of data (*e.g.*, a single game, all away matches[1], entire season, etc.). If it has a characteristically different distribution for the focus attribute, it is marked as interesting. Such interesting patterns are discovered and provided to the domain expert to investigate further and gain insights. More recently, Schultz [23] studies how to determine types and combination of players most relevant to winning matches.

2.1.2 Baseball

In baseball, Gartheepan et al. [14] built a data driven model that helps in deciding when to 'pull a starting pitcher'. Pulling a starting pitcher is the act of replacing the pitcher who started the inning with another pitcher. This is considered to be an important decision in the context of baseball. In their subsequent work [12], the authors propose a model that could lead to better on-field decisions for the next inning. Bukiet et al. [8] use Markov chain method for evaluating the performance of teams and influence of a particular player on the team performance.

2.1.3 Soccer

Luckner et al. [22] have predicted the outcome of FIFA World Cup 2006 matches using live Prediction Markets. Palomino et al. [26] study soccer matches with a game-theoretic model and determine that teams' skill level, current score and home field advantage are significant explanatory variables of the probability of scoring goals. Kang et al. [17] build a mathematical model to quantitatively express performance of soccer players based on the trajectory of the ball passes among the 22 on-field players.

The work discussed above is developed with sport-specific intuitions. Both soccer and basketball are fundamentally very different from cricket which would render the work inapplicable to the sport of cricket. Baseball is probably the closest to cricket in terms of playing dynamics since both are bat and ball games with batters and bowlers and the objective is to score the maximum number of runs. But given the facts that it has very different rules, ground shape and dimension, play

[1] Matches played away from home.

structure, in-game economy etc., it would not be appropriate to apply the baseball-specific models to cricket. Moreover, none of the existing work models the sport to predict future match states of an on-going game. They are either used as a pre-match prediction tool or as a post-match analysis framework.

2.2 Research in Cricket

One of the earliest and pioneering work in cricket was by Duckworth and Lewis [9] where they introduce the Duckworth-Lewis or D-L method, which allows for fair adjustment of scores in proportion to the time lost due to match interruptions (often due to adverse weather conditions such as rain, poor visibility etc.). If the interruptions occur during the second innings, the team batting second will have less batting time and will face fewer balls. This affects the equilibrium and puts the second team at a disadvantage. To mitigate this, the target for victory has to be adjusted in proportion to the time lost. The D-L method is based upon a mathematical formulation that abstracts every ball and wicket of an ODI match into a single scalar called *resource*. Using this formulation, the number of overs lost is evaluated as runs and used to reset targets for the second team. This proposal has been adopted by the International Cricket Council (ICC) as a means to reset targets in matches where time is lost due to match interruptions. The method proposed in [9], and subsequently adapted by [25], for capturing the resources of a team during the progression of a match has found independent use in subsequent work in cricket modeling and mining [25][2].

One of the objectives in sports analytics is to rate and rank players. In cricket, some possible ranking criteria are statistics such as batting average and strike rate for batsmen (defined formally in Section 4.5) in determining most valued players. Lewis [19], Lemmer [18], Alsopp and Clarke [1], and Beaudoin [3] develop new performance measures to rate teams and to find the most valuable players.

Raj and Padma [27] analyze the Indian cricket team's One-Day International (ODI) match data and mine association rules from a set of features, namely toss, home or away game, batting first or second and game outcome. Kaluarachchi and Varde [16] employ both association rules and naive Bayes classifier and analyze the factors contributing to a win, also taking day/day-night game into account.

Both approaches use the same subset of high-level features to analyze the factors contributing to victory. Furthermore, they do not address score prediction, nor the progression of the game.

Brooker et al. [6] take into account the number of runs scored, wickets lost, balls remaining at any point in the game along with the ground conditions and estimate the end-of-innings score. This work fails to take the batsmen and bowler features into account. The authors note that it is not practical to learn a model for every single bowler and batsman that has played the game so far. Bailey and Clarke [2] use historical match data and predict the total score of an innings using linear regression. As data of a match in progress streams in, the prediction model is updated. Using this, they analyze the betting[2] market's sensitivity to the ups and downs of the game. Their model predicts total score as instantaneous match data is streamed in. This enables us to use their model as a baseline for this work. Swartz et al. [31] use Markov Chain Monte Carlo methods to simulate ball by ball outcome of a match using a Bayesian Latent Variable model. Based on the features of current batsman, bowler, and game situation (number of wickets lost and number of balls bowled), they estimate the outcome of the next ball. This model suffers from severe sparsity as noted by the authors themselves: the likelihood of a given batsman having previously faced a given bowler in previous games in the dataset is low. Simulating a match, based on team compositions and making use of a model built from historical match data and taking in the current match situation, is a key step in predicting the outcome of a match. While both [31] and [2] have built match simulators for ODI cricket, their models rely on games played over 10 years ago. ODI cricket has since undergone a number of major rule modifications. Important examples include powerplays, free hit after an illegal ball delivery, and the use of two new balls (as opposed to just one) in an innings. These changes significantly affect the team strategies, and essentially render old models a poor fit. This work focuses on the modern and current form of ODI cricket, incorporating all recent changes to the game with support for accommodating future rule modifications.

[2]There is a vibrant betting market associated with cricket. See, *e.g.*, http://www.betfair.com/exchange/en-gb/cricket-4/sp/.

Chapter 3

Rules of the sport

This chapter provides an overview of the ODI format of the game and review its basic rules as they pertain to the problem of modeling the game progression and score prediction. It serves as foundation to the contributions in this thesis.

3.1 Rules and Objective

The objective of the teams is to score as many runs as possible while batting and limiting the opponents from scoring while bowling. The team that has the highest number of runs at the end of the game is determined to be the winner. Figure 3.1 is a depiction of a cricket field and is used for representational purposes. A team consists of 11 players. Based on the team's strategy it can contain any number of batsmen and bowlers (typically 6 batsmen and 5 bowlers). Batsmen and bowlers can be left-handed or right-handed. Different batsmen have specific roles like opening the innings, consolidation (during overs 15 to 40), power-hitting (during overs 41 to 50). Bowlers can be specialist in pace bowling (bowling fast and enabling the ball to swing) or spin bowling (rotation of the ball while releasing that aids turning in direction after landing on the pitch)

Theoretically, all 11 players can bat or bowl in a match if needed. At any time, two players of the batting team (in blue) and eleven players of the bowling team are present on the field. The pitch is the pale yellow rectangular shape in the middle of the field. It is 22 yards (20.11 meters) in length. The playing area is enclosed by

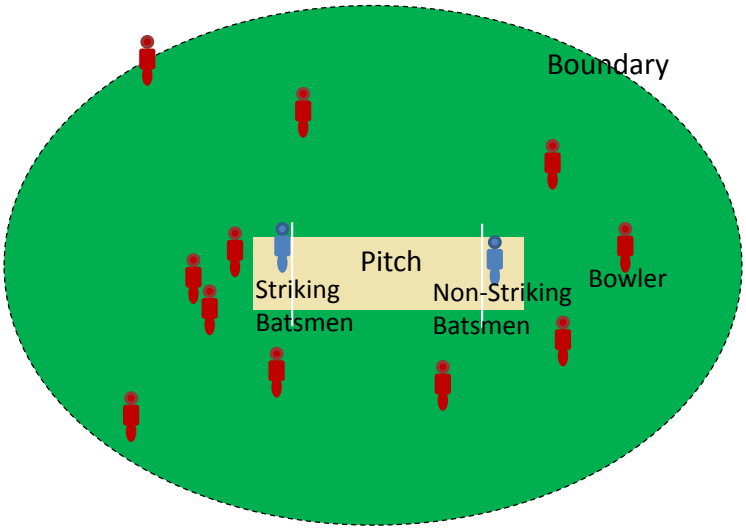

Figure 3.1: A depiction of a typical cricket field. It is not mandatory for the field to be oval in shape.

the boundary. 4 or 6 (*home*) runs are awarded based on whether the ball lands in the playing area and rolls over the boundary or flies past the boundary. After hitting the ball, the two batsmen exchange their position as many times as they can. These are called *non-home* runs. Players of the bowling team field the ball to minimize the number of exchanges the batsmen attain. At any time in the game, there are two umpires present on the field to officiate the match.

The most important components and terminologies in the game of cricket are described below.

3.1.1 Toss

Similar to a number of other sports, an ODI cricket match starts with a toss. The team that wins the toss can choose to bat first or can ask the opponents to bat first. This is an important decision in the context of the game. The teams take into account the nature of the *pitch* (Pitch is the 22 yard central strip of the cricket field

where bowlers bowl the ball for the batsmen to hit.), weather conditions and the strengths and weaknesses of their own and that of the opponents to arrive at the decision.

3.1.2 Over

Six consecutive legal delivery of balls by a bowler to the batsmen is called an *over*. An ODI game consists of two *innings* and each have 50 *overs* (300 legal deliveries).

3.1.3 Objective

In a game between $Team_A$ and $Team_B$, suppose $Team_A$ wins the toss and chooses to bat first. The period during which $Team_A$ bats is called $innings_1$, in which bowlers from $Team_B$ will have to *bowl* the ball to the batsmen of $Team_A$. $Team_A$ has 50 *overs* to score as many *runs* as possible, while $Team_B$ tries to minimize the scoring by getting $Team_A$'s batsmen *out* (more commonly referred to as taking *wickets*). Scoring can also be restricted by $Team_B$, by bowling balls that are difficult to play and by flawless fielding, where *fielders* stop hits by batsmen of $Team_A$ to deny them opportunities to score runs. $Innings_1$ comes to an end when $Team_A$ loses all its wickets or finishes its quota of 50 overs, whichever happens first. When a team loses all its wickets, it is termed as being *all-out*. Let $Score_A$ denote the number of runs accumulated by $Team_A$ at this point. When $Team_B$ comes in to bat in $innings_2$, it has the exact same number of 50 overs to play (not considering rain intervention), with the goal of scoring at least $Score_A+1$ runs; $innings_2$ ends when $Score_B$, the number of runs scored by $Team_B$, exceeds $Score_A$, or when $Team_B$ finishes its quota of 50 overs or loses all its wickets, whichever happens first. $Team_B$ is deemed the winner in the first case, and $Team_A$ wins otherwise. A third possibility is a *tie* when $Score_A$ and $Score_B$ are equal at the end of the game.[1]

3.1.4 Scoring

Teams can accumulate runs in two ways - home runs and non-home runs as described below. When a bowler commits a foul while delivering the ball, the bowling team is penalized by awarding run(s) to the batting team. The ball delivered by

[1] Currently, there are no tie-breakers in ODI the format.

the bowler is deemed illegal by the on-field umpires. Some of the main reasons for a delivery to be illegal are as follows:

- The bowler crosses the crease while releasing the ball
- The ball lands and bounces on the pitch more than once
- The ball does not land on the pitch and reaches the batsman directly above his hip.
- The ball falls very wide of the batsman making it unplayable.

These penalty runs are termed as *extras* and contribute a small fraction of the total runs. In our model, the *non-home* run category accounts for extras.

Home Runs

One way of scoring is to power-hit the ball outside the playing area. Based on where the ball lands while traveling past the boundary, *four* or *six* runs are awarded. *Four* runs are awarded if the ball touches the ground before rolling past the boundary of the playing area. If the ball lands directly outside the playing area (thereby not touching the ground within the playing area), *six* runs are awarded. Borrowing a term from baseball, for convenience, we collectively term runs scored this way as *home runs*. Home runs yield greater reward in terms of runs scored, but the batsmen have to take risks to hit them, which increases their chance of getting out.

Non-Home Runs

The other way of scoring is to hit the ball within the playing area and for the two batsmen to run and exchange their positions. In the mean time, the opponent players try to collect the ball to minimize the number of exchanges. Runs are awarded based on the number of times the batsmen exchange their positions before the ball is returned to one of the positions. There is theoretically no bound on the number of exchanges possible in a given ball but this value typically lies in the range *0–3* runs. This way of scoring has a lower risk of the batsman getting out but yields a lower number of runs. We term these *non-home runs*.

It is to be understood that batsmen have different intentions and mind-set when they try to score home runs and non-home runs. Hence their approach to scoring each of the two are also different. Since home runs involve high risks than scoring non-home runs, the number of non-home run scoring balls greatly outnumber the number of home run scoring balls as substantiated by our data. The decision to hit a particular ball to the boundary (home run hit) is taken by the striking batsman based on a combination of factors like team's score, his strengths, merit of the ball, merit of the bowler, fielders' placement in the ground etc. Understanding this minute yet significant dynamic of the game has helped to come up with separate models for predicting home runs and non-home runs (More on this in section 4.2)

3.1.5 Dismissal

There are eleven ways for a batsman to lose his *wicket*, commonly referred to as getting out or being dismissed. The common ways to get dismissed are the following

- Bowled: when the ball delivered hits the stumps[2]
- Caught by opponent fielders: When the ball hit by the batsmen is caught by the fielders before it touches the ground.
- Run-out: This happens when the batsmen try to score runs by exchanging their positions. A batsmen loses his wicket if he is found short of his position when the ball (fielded and thrown back by opponent fielders) hits the stumps.
- Leg Before Wicket (LBW): When the ball delivered hits the batsman's body parts and the umpire determines that the it would have traveled to knock the stumps had it not hit his body, he is deemed out.

There are a few other modes of dismissal which are uncommon. In our model, we do not distinguish between the different forms of dismissal.

[2]Stumps are the three vertical posts with two support bails that are placed in the pitch. A batsman guards them so that the ball delivered by the bowler does not knock them.

3.1.6 Target score

The number of runs accumulated by $Team_A$ at the end of $innings_1$ is $Score_A$. $Score_A+1$ run is set as the *Target*. This is the score that the team batting second tries to achieve or exceed in $innings_2$.

3.1.7 Resources

Overs and Wickets are collectively termed as *resource*. The batting team consumes the overs to accumulate runs and loses wickets in the process. A batting team has 50 overs and 10 wickets at their disposal at the start of an innings. This resource continually decreases as the game progresses.

The rules, terminologies and objective of a cricket game was explained in this chapter. Next chapter explains about the game modeling using relevant historical and instantaneous features and our algorithm in detail.

Chapter 4

Game Modeling

This chapter describes the problem formulation with relevant features in detail. Furthermore, it proceeds to explain our algorithm that predicts game progression and the winner.

4.1 Terms & Notations

A few very important terms and notation that will be used in our model are described below.

4.1.1 Segment

To predict the end-of-innings score of a team, a *segmented prediction* approach is taken. The batting period of a team is called an *innings* and it lasts till they run out of one of the *resources*. The 50 over innings is segmented into 10 intervals of 5 overs each, where each interval is referred to as S_i, $1 \leq i \leq 10$.

4.1.2 Runs in a segment

For a team T, R_i^T and W_i^T denote the the number of runs scored and the number of wickets lost in segment S_i, respectively. For a segment S_i, NHR_i and HR_i denote the non-home runs and home runs scored in that segment respectively. Together, they form the runs scored in that segment i.e., $R_i = NHR_i + HR_i$

4.1.3 Match state

The *Match state* at segment n, $0 \leq n < 10$ is defined as the pair of numbers consisting of the number of runs scored and the number of wickets lost so far, by the batting team. Notice that given a match state, the resources remaining at the batting team's disposal can be easily calculated: the number of balls remaining is $(10-n) \times 5 \times 6$ and the number of wickets remaining is $10-(\#\text{wickets lost so far})$.

4.1.4 End-of-innings score

The total number of runs scored by team T at the end of their innings is given by $R^T_{eoi} = \sum_{i=1}^{10} R^T_i$. The superscript T is dropped when the team is obvious from the context.

4.2 Problem Formulation

The main problem tackled in this work is, given the *instantaneous* match data up to a certain point in the game, predict the progression of the remainder of the game, and in particular, predict the winner. If the no instantaneous data is available since the match is yet to begin, predict the progression and outcome using available match information. This is a special case with $n = 0$ and dealt with by supplying $R_{known} = 0$ and $W_{known} = 0$. More precisely, given a match state associated with segment n, namely $(R_{known} = \sum_{i=1}^{n} R_i, W_{known} = \sum_{i=1}^{n} W_i)$, predict the number of runs $\hat{R}_i$ for the remaining segments i, $n+1 \leq i \leq 10$. Using these predictions, the total predicted score at the end of the innings, $\hat{R}_{eoi}$ can be obtained as

$$\hat{R}_{eoi} = R_{known} + \sum_{i=n+1}^{n} \hat{R}_i \tag{4.1}$$

To predict the number of runs scored in a segment S_i, both *historical data* as well as *instantaneous match data* available till segment S_{i-1} are used. The current state of the match are the *instantaneous features* that is used for game state prediction. Both sets of features are explained in further detail in the following sections.

If an innings has not commenced, as a special case, $n = 0$, $R_{known} = 0$ and $W_{known} = 0$. In this case, the task becomes to predict number of runs $\hat{R}_i$, for all

$i = 1, ..., 10$. The total predicted score is then

$$\hat{R}_{eoi} = \sum_{i=1}^{10} \hat{R}_i \tag{4.2}$$

This segmented prediction approach is followed to predict $\hat{R}_{eoi}$ for both $innings_1$ and $innings_2$. $Team_A$ is predicted to be the winner if $\hat{R}^A_{eoi} > \hat{R}^B_{eoi}$. $Team_B$ is predicted to be the winner if $\hat{R}^A_{eoi} < \hat{R}^B_{eoi}$. If both $\hat{R}^A_{eoi} == \hat{R}^B_{eoi}$, it predicted to be a tie.

The task of predicting the number of runs in the next segment S_{n+1}, given the match state up to segment S_n is broken down into two subproblems:

- (i) predicting non-home runs in S_{n+1} - $N\hat{H}R_{n+1}$
- (ii) predicting home runs in S_{n+1} - $\hat{HR}_{n+1}$

Hence, $\hat{R}_{n+1} = N\hat{H}R_{n+1} + \hat{HR}_{n+1}$

While it may seem counter-intuitive to use two different classes of techniques to predict the overall total score, this decision was driven by observing the inherent nature of the game itself, and has eventually been justified by our experimental results. As explained in section 3.1.4, scoring home and non-home runs take different intentions, approach, skill, risk level and consequently different outcomes (in terms of runs). Moreover, the number of non-home run scoring balls greatly outnumbers the number of home run scoring balls in a typical game. A linear-regression based approach to predict home runs thus runs into the problem of data sparsity. Attribute bagging, on the other hand, enables our system to find matches that have similar home-run scoring patterns, given the set of match features, and thus avoids the sparsity issue altogether. Our experiments have shown much degraded performance when using ridge regression for home run prediction.

Prediction of $\hat{HR}_i$ and $N\hat{H}R_i$ for a segment S_i is accomplished using two sets of features – *historical features* and the *instantaneous features*, described next. Of these, historical features are critical for predicting runs for the first segment, since by definition, no instantaneous match data is available before the first segment.

4.3 Historical Features

Our model consists of 6 *historical features* for each team in the dataset. They are mined from data across all matches played by a given team. The *historical features* of a team are as follows:

1. Average runs scored (by the team) in an innings
2. Average number of wickets lost in an innings
3. Frequency of being all-out[1]
4. Average runs conceded in an innings
5. Average number of opponent wickets taken in an innings
6. Frequency of getting opposition all-out

In what follows, N denotes the total number of matches in the training dataset. Recall, n denotes the segment up to which match state is known. The first feature is calculated by dividing the total runs scored by the given team across the number of matches it played.

$$AverageScore = \frac{\sum_{i=1}^{N}(\text{Runs scored in } match_i)}{N} \tag{4.3}$$

The subsequent five features are self-explanatory and are calculated similarly to (4.3). Out of the 6 features, the first three represent the team's batting ability, while the last three represent the team's bowling ability.

Table 4.1 presents the historical batting and bowling features for the teams considered in our dataset. Team India, the winner of World Cup 2011, is considered to have a stronger batting ability compared to other teams. Team South Africa ranks second and is known to possess one of the best bowling attacks. The batting and bowling features indicated in the table for these teams agree with these facts: *e.g.*, India's average score is the highest and South Africa's runs conceded and average number of wickets taken are both the lowest and frequency of getting opposition all-out is the highest.

[1] That is, losing all 10 wickets in an innings within 50 overs.

	Batting Strength			Bowling Strength		
Team	**Average Score**	**Wickets lost**	**all out**	**Runs Conceded**	**Wickets taken**	**all out**
Australia	238	7.3	0.27	227	7.7	0.49
Bangladesh	199	7.5	0.32	224	7.0	0.27
England	233	6.9	0.37	237	7.4	0.35
India	**247**	**7.0**	**0.32**	248	7.6	0.28
New Zealand	223	8.3	0.43	218	7.1	0.33
Pakistan	212	7.5	0.41	212	7.5	0.34
South Africa	237	6.9	0.24	**210**	**8.4**	**0.52**
Sri Lanka	227	7.3	0.35	231	6.9	0.31
West Indies	208	7.8	0.48	220	7.2	0.35

Table 4.1: Historical feature values for teams in the dataset

4.4 Instantaneous Features

In addition to the features mined from past game data, *i.e.*, the historical features, several instantaneous match features incorporated in our prediction model.

4.4.1 Home or away

This is a binary feature describing if the batting team is playing in its *home ground*. If the match is played in a neutral venue, this feature carries no weight for both teams.

4.4.2 Venue class

The match venue is incorporated as a multinomial feature called "Venue Class", which classifies the location of the venue into one of the following four continent clusters: (i) Australia/New Zealand; (ii) Indian Subcontinent; (iii) The British Isles; and (iv) The Caribbean/South Africa. This classification is significant since the pitch (*i.e.*, the area where the ball is bowled and pitched) and weather conditions across these region clusters are known to be substantially different. Pitches across

different continental plates are known to behave differently and offer different assistance to bowlers and batsmen. For example, it is well known that pitches in the Indian subcontinent are batsmen friendly and offer assistance to spin bowlers. Hence teams accumulate high scores when they play on the Indian subcontinent. Pitches in Australia are known to be bouncy and offer great assistance to pace bowlers. To capture this phenomenon, we group the venue based on continents.

4.4.3 Powerplay

Powerplay is a restriction on the number of fielders that could be placed by the bowling team outside a certain range from the batsmen (usually 30 yards, approx. 27.432 meters). This restriction enables the batsmen to hit the balls aggressively and try and score home runs, with a relatively reduced risk of getting out. The first 10 overs of the game are mandatory powerplays, with two more instances of powerplay periods arbitrarily chosen by the batting and bowling team each, to occur at any point in the game up to the 45^{th} over. For any segment, the powerplay can occupy between 0 and 5 overs of the segment. Consequently, the value of this feature ranges from 0 to 1 in increments of 0.2.

4.4.4 Target

The goal of the team batting second is to achieve the *Target Score*, ($Score_A$ +1 runs). This is used as a feature for $innings_2$ prediction.

4.4.5 Batsmen performance features

For any given segment S_n, we identify four performance indicators for each of the two currently playing batsmen. They are

- batsman-cluster (to be described in section 4.5)
- Number of runs scored till segment S_{n-1}.
- Number of balls faced till segment S_{n-1}.
- Number of home runs hit till segment S_{n-1}.

4.4.6 Bowler quality features

Using a few performance indicators from history, bowlers are categorized into three classes (explained in detail in section 4.6). The value of a feature indicate the number of overs bowled by a bowlers of that class in a particular segment. The three bowler-features always sum upto 5, which is the number of overs in a segment.

4.4.7 Game snapshot

This feature is a pair of game state variables, namely the current score and the number of wickets (*i.e.*, number of batsmen) left.

Batsmen performance features, bowler quality features and game snapshot features are are explained in detail in Sections 4.5, 4.6 and 4.7.

4.5 Batsmen Clustering

In our dataset, there are more than 200 players who have batted at least one ball. Given data corresponding to 125 matches, learning the features for each of the 200 individual players is fraught with complexity and extreme sparsity. To give an example, given a currently playing batsman b and a current bowler ℓ, the probability that b has faced ℓ in earlier matches can be quite low. Even when b has faced ℓ before, the number of such matches can be too small to learn any useful signals from, for purposes of prediction. To quantify, if in a dataset of M matches, the average number of matches played by player b is m_b, and by player l is m_l (where $M \gg m_b$ and $M \gg m_l$), even assuming independence, the probability that b and l played together is $\frac{m_b}{M} \times \frac{m_l}{M}$.

To overcome this sparsity, clustering using k-means algorithm, (un-supervised learning technique) is used to group batsmen with similar batting skills. The following four important features were considered.

1. Batting Average
2. Strike Rate
3. Home-run hitting ability
4. Milestone reaching ability.

Batting Average for a batsman is the ratio of the total number of runs he has scored across all matches, over the number of times he has gotten out. Strike Rate is the average number of runs scored per 100 balls, again calculated across all matches played. These two features are standard metrics used to report batsmen stats in cricket. Although they are used to express a batsman's quality, they do not quite capture his skill as observed by cricket experts and proved by [30] and [1]. Hence Features 3 and 4 are used to help capture the quality of batsmen more accurately.

4.5.1 Home-run hitting ability

Since a home run hit results in higher number of runs being scored, batsmen who can hit home runs frequently are considered to be skilled and strong. This parameter helps capture better batsmen and is measured as

$$HR\text{-}hittingAbility = \frac{\sum_{i=1}^{N} \#\text{home runs hit in match}_i}{\sum_{i=1}^{N} \text{balls faced in match}_i} \tag{4.4}$$

4.5.2 Milestone reaching ability

Scoring fifty runs or a hundred runs (commonly referred to as half-century and century) are considered batting milestones in cricket. Players who consistently and frequently reach these milestones are considered to be of very high caliber and bring value to the team. Milestone reaching ability (MRA) captures this quality and is measured as

$$MRA = \frac{\#\text{ of 50 \& 100 run scores in } N \text{ matches played}}{N} \tag{4.5}$$

Combining the Batsman Average and Strike Rate with Home-run hitting ability and Milestone reaching ability, batsmen are clustered into five clusters using the k-means clustering. The number of clusters (five) was chosen based on intuition and exploration. Generally, a team consists of opening batsmen, middle-order batsmen, all-rounders, wicket-keeper, and tail-enders (bowlers). Each have different roles and batting capabilities. A healthy combination of the *five* types add up to form the playing *eleven* members of the team.

Table 4.2 presents clusters of a few current players. Clusters 4 and 5 seem to

capture the players who are primarily bowlers and as a result do not have good batting credentials. The first 3 clusters capture the batsmen who, according to the rather vague informal terminology used in popular cricket press, are labeled as 'Great Solid Batsmen', 'Pinch Hitters' and 'Reliable Batsmen'.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Sachin Tendulkar	Virender Sehwag	Mohammed Hafeez	Nuwan Kulasekara	Andy Mckay
Ricky Ponting	Kieron Pollard	Matt Prior	James Pattinson	Stuart Meaker
Jacques Kallis	Shahid Afridi	Tim Paine	Vinay Kumar	Junaid Khan
Hashim Amla	Yusuf Pathan	Ajinkya Rahane	James Anderson	Suranga Lakmal

Table 4.2: Batsmen clustered according their ability

4.6 Bowler Classification

More than 180 bowlers have bowled at least 1 full *over* (collection of 6 continuous legal delivery of balls) in our dataset. As observed with batsmen (in section 4.5), constructing individual models for every bowler is also fraught with complexity and extreme data sparsity. Hence the bowlers were classified (supervised learning technique) into *three* classes based on their historical economy rate. *Economy rate* of a bowler is defined as the average number of runs conceded per over. More formally, assuming N to be total number matches played by a bowler, his

$$economy\ rate = \frac{\sum_{i=1}^{N} \text{runs conceded in match}_i}{\sum_{i=1}^{N} \text{overs bowled inmatch}_i} \tag{4.6}$$

Since the goal is to predict the number of runs conceded by bowlers in 5 over intervals, classifying the bowlers based on their economy rate proved to be useful indicator of their quality. Figure 4.1 shows the histogram spread of bowlers based on their historical economy rate. It could be observed that the mass lie between 4 to 6 runs. Bowlers with lesser economy rate are thrifty and considered to be of very good quality. Such bowlers, with $economyrate \leq 4.5$ runs were classified as

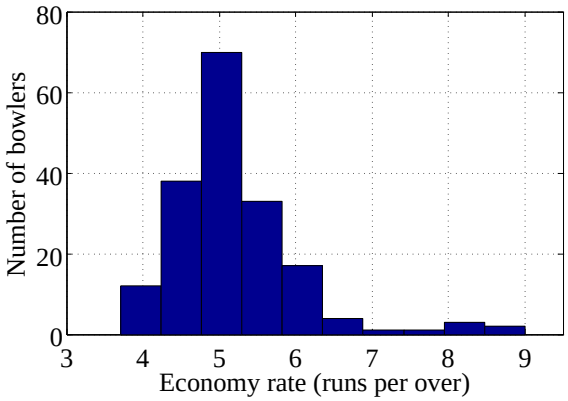

Figure 4.1: Histogram of bowlers' economy rate (average runs conceded per over)

class 1. 24 bowlers had economy rate less than 4.5 runs. Bowlers whose economy rate lay between 4.5 and 5.5 are of average quality and were classified as *class 2*. 112 bowlers were classified into the average class. Bowlers with $economyrate \geq 5.5$ are considered expensive were classified as *class 3* with 45 bowlers falling into this category. This group primarily had part-time bowlers (bowlers who are not specialists). The number of classes and class boundaries were arrived using a combination of cricket knowledge and exploratory analysis.

4.7 Game Snapshot

Recall that the problem is, given the match state data up to segment $n < 10$, *i.e.*, runs scored R_i and wickets lost W_i in segment i, $1 \leq i \leq n$, number of runs for segment $n+1$ is to be predicted. To facilitate this, information in segments S_1 to S_{n-1} are aggregated and the information in segment S_n is retained separately. More precisely,

1. Features from aggregated instantaneous information in segment S_1 to S_{n-1}
   (a) $R_{1:n-1} = \sum_{i=1}^{n-1} R_i$
   (b) $W_{1:n-1} = \sum_{i=1}^{n-1} W_i$
2. Features from instantaneous information in segment S_n

(a) R_n

(b) W_n

For example, let us take the task of predicting the runs in segment S_6 (overs 26 to 30). runs scored and wickets lost in segments S_1 to S_4 are aggregated and given by $R_{1:4}$ and $W_{1:4}$. Runs and wickets in segment S_5 are retained as such and given by R_5 and W_5. This quasi markovian approach provides the game information till segment S_{n-1} and the game information in segment S_n separately to the model. This provides a broader snapshot of match state and also gives more importance to the immediately preceding segment.

The learning algorithm, described in section 4.10, makes use of the aforementioned historical and instantaneous features up to a given segment to predict scores for subsequent segments and uses that to predict the overall score $\hat{R}_{eoi}$. When $n = 0$ (when game is yet to begin), game snapshot features don't exist. So, in addition to the historical features, available instantaneous features like home/away, Venue Class, Target, etc. are fed to the algorithm to make its predictions.

4.8 Home-Run Prediction Model

Using the historical and non-historical features discussed in chapter 4, the number of home runs $\hat{HR}_i$ for a segment S_i are predicted using attribute bagging ensemble method [7] with nearest-neighbor classification. Random subsets of features for n classifiers with l features each are chosen and the results are aggregated overall. Different sets of features corresponding to the previous states are chosen randomly and their nearest neighbors are identified from history, thereby leveraging the Markovian nature of segments. Number of features for every classifier is set to be the square root value of the total number of features [7]. The number of classifiers is experimentally determined. The intuition behind using nearest-neighbor algorithm is that information from similar match situations can be "borrowed" from the training dataset. Standardized euclidean distance metric is used as the distance metric. Standardizing attribute values involve subtracting the original value from its mean and dividing the difference by the standard deviation. Euclidean distance, also called the *L1* norm between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ in a two di-

mensional systems is given by

$$dist(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.7}$$

A number of alternate distance metrics like Spearman, Jaccard, Cosine, Manhattan, Euclidean etc. were explored. But Nearest Neighbor algorithm with *Standard Euclidean* distance metric performs the best in terms of mean absolute error.

After running n number of classifiers with l features each, the top 5 neighbors based on frequency counts are picked and the number of home run hits is the mean of these neighbors' home run hits. This number of neighbors, 5, has been determined experimentally.

4.9 Non-Home-Run Prediction

Using the same historical and instantaneous features, non-home runs of segment S_i, $\hat{NHR_i}$ is predicted by means of Ridge Regression [24]. Linear model was found to work best for predicting non-home runs.

Using the home run and non-home run prediction models, the iterative algorithm to predict the runs R_i of future segments and consequently, R_{eoi} of the whole innings is described in the next section 4.10,

4.10 Algorithm

The following describe the notation used in the algorithm.

- $n \leftarrow$ segment number till which match information is available
- R_i, Runs scored in segment S_i where $1 \leq i \leq n$
- $\hat{NHR_i}$, predicted non-home runs in segment S_i
- $\hat{HR_i}$, predicted home runs in segment S_i
- $\hat{R}_i$, predicted total runs in segment S_i
- $\hat{R}_{eoi}$, predicted end of innings score

- $\Theta \leftarrow$ historical features
- $\Delta_n \leftarrow$ instantaneous features till segment S_i

Θ is the set of historical features given in Section 4.3, which remain constant through the iterations since they are learned just once from the historical match data; n is the segment number till which, match state data, also called instantaneous features, Δ_n are available. At the start of the algorithm, features Θ and Δ_n are fed as input to the algorithm which proceeds iteratively to predict $\hat{R}_i$, for every segment i, $n+1 \leq i \leq 10$.

Algorithm 1: Prediction of Future Segments runs $\hat{R}_i$ & End of Innings Score $\hat{R}_{eoi}$

Input : Θ, Δ_n, n
Output: R_i for every $n+1 \leq i \leq 10$, $\hat{R}_{eoi}$

```
1  for i ∈ n+1 ≤ i ≤ 10 do
2      for j ∈ 1 ≤ j ≤ L do
3          Γ_j ← RandomSubspace(Θ, Δ_{i−1})
4          Φ_j ← NearestNeighbor(Γ_j)
5      end
6      HR_i ← MajorityVoting(Φ_{1:L})
7      NHR_i ← RidgeRegression(Θ, Δ_{i−1})
8      R̂_i ← HR_i + NĤR_i
9      Δ_i ← Update(Δ_{i−1}, R̂_i)
10 end
11 R̂_eoi ← Σ_{i=1}^{n} R_i + Σ_{i=n+1}^{10} R̂_i
```

Using Θ and Δ_{i-1} (instantaneous features of previous segment), home runs $\hat{HR}_i$ of segment i is predicted using attribute bagging algorithm as explained in section 4.8. This is explained in lines 2 to 6 of algorithm 1.

For every classifier j in L (Number of classifiers), a random subspace of features (Γ_j) is chosen from the overall feature space (Line 3). The nearest neighbor based on this subspace of features (Γ_j) is found out to be Φ_j (Line 4). Based on *majority voting* among the chosen neighbors ($\Phi_{1:L}$), the closest neighbor from the training set is found and home-run information is borrowed. (Line 6) Using the same features Θ and Δ_{i-1}, non-home runs $N\hat{HR}_i$ are predicted using Ridge Regres-

sion as mentioned earlier in this section (line 7). $\hat{HR}_i$ and $N\hat{H}R_i$ are added together to give $\hat{R}_i$, the predicted number of runs scored in segment S_i (Line 8).

It is to be noted that two of the instantaneous features (Section 4.4) namely Home/Away and Venue Class do not change through the course of prediction while Game Snapshot features are constantly updated based on predictions of the previous iteration as explained in Section 4.7. Number of runs for segment i, $\hat{R}_i$, predicted in this iteration, is added to the Game Snapshot features of Δ_{i-1}, thereby modifying it to Δ_i *(Line 9)*. This Δ_i is used to predict the $\hat{HR}_{i+1}$ and $N\hat{H}R_{i+1}$ in the next iteration.

The cumulative sum of all known runs R_i, $1 \leq i \leq n$ and all predicted runs $\hat{R}_i$, $n < i \leq 10$, is the predicted End-of-Innings Score, $\hat{R}_{eoi}$ *(Line 11)*.

4.11 Cold-Start

If prediction is initiated without any match information, *i.e.*, before the start of the actual innings, then $n = 0$, and the algorithm starts prediction from S_1 through S_{10}.

In the first iteration when $i = 1$, no game snapshot features are available. As the opening pair of batsmen are known before the start of a game, their cluster ID numbers (from the feature Batsmen cluster) are used. In order predict $\hat{R}_1$ accurately, we leverage the Home/away and Venue class information along with the other historical features Θ.

Chapter 5

Experiments & Results

The dataset consists of 125 complete matches played between January 2011 and July 2012 among the 9 full-time ICC teams of Australia, Bangladesh, England, India, Sri Lanka, Pakistan, South Africa, New Zealand and the West Indies who have played more than 20 matches each excluding all rain-interrupted and rain-abandoned games. We split the dataset into training and test set with 100 and 25 matches respectively. Ten-fold cross validation was performed on the 100 training matches to learn the regression parameters to predict *non-home* runs and the model was tested on the remaining 25 matches. The mean of the cross-validation trials are reported for *Non-home* run prediction. The data was crawled from http://www.espncricinfo.com, where ball-by-ball data on all the matches are available publicly. Team, batsmen and bowler statistics for mining historical features and batsmen clusters, bowler classes are also queried from their publicly-accessible statistics databases. Since ball-by-ball commentary data consists of transcription of the human interpretation of actual events, there are occasional missing values and errors. They were fixed either manually or by automated consistency checks with the end-of-over summary, scorecards, partnership information and other relevant game data. After the required data is gathered, they were aggregated and rolled-up to 5-over levels (since a segment is a collection of 5 overs) without loss of necessary information. Once the data is available in the processed form, running time for the model to learn the parameters across all the segments and testing by 10-fold cross-validation takes less than 5 seconds on MATLAB in a 4-core, 2.66

GHz machine with 8 GB RAM, running OpenSuse 12.3.

5.1 Non-Home Run Prediction Performance

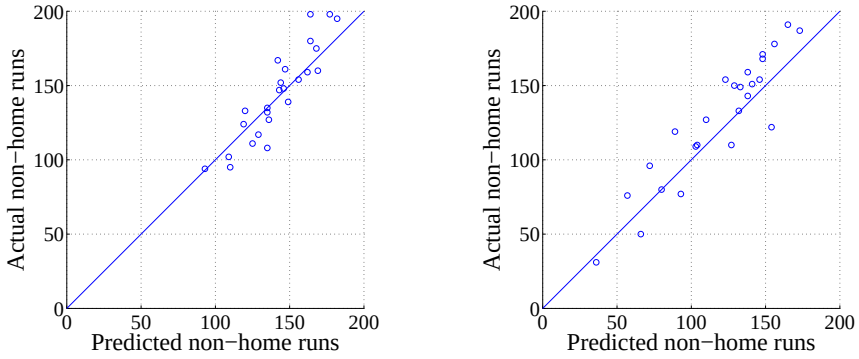

Figure 5.1: Total *non-home runs* scatter plot for $innings_1$ (left) & $innings_2$ (right)

Prediction of $N\hat{H}R_{eoi}$, which is the sum of individual $N\hat{H}R_i$ is shown in Figures 5.1 and 5.2. Figure 5.1 shows scatter plot runs for both $innings_1$ and $innings_2$ and demonstrates good agreement between the predicted and actual non-home runs.

Figure 5.2 shows the total non-home run prediction error distribution across all the matches. The plot on the left gives the Probability Density Function (*PDF*) and the one on the right gives the Cumulative Distribution Function (*CDF*). The median number of non-home runs in an innings in our dataset is 125. It can be seen that for $innings_1$, for 80% of the matches, the error margin is less than or equal to 10 runs. For $innings_2$ for more than 80% of the matches, the error margin is less than 20 runs. There is a fundamental difference in the economics of the first and second innings because of the availability of a "*target*" score in the latter. It is counter-intuitive that the availability of a target score makes the second innings more unpredictable.

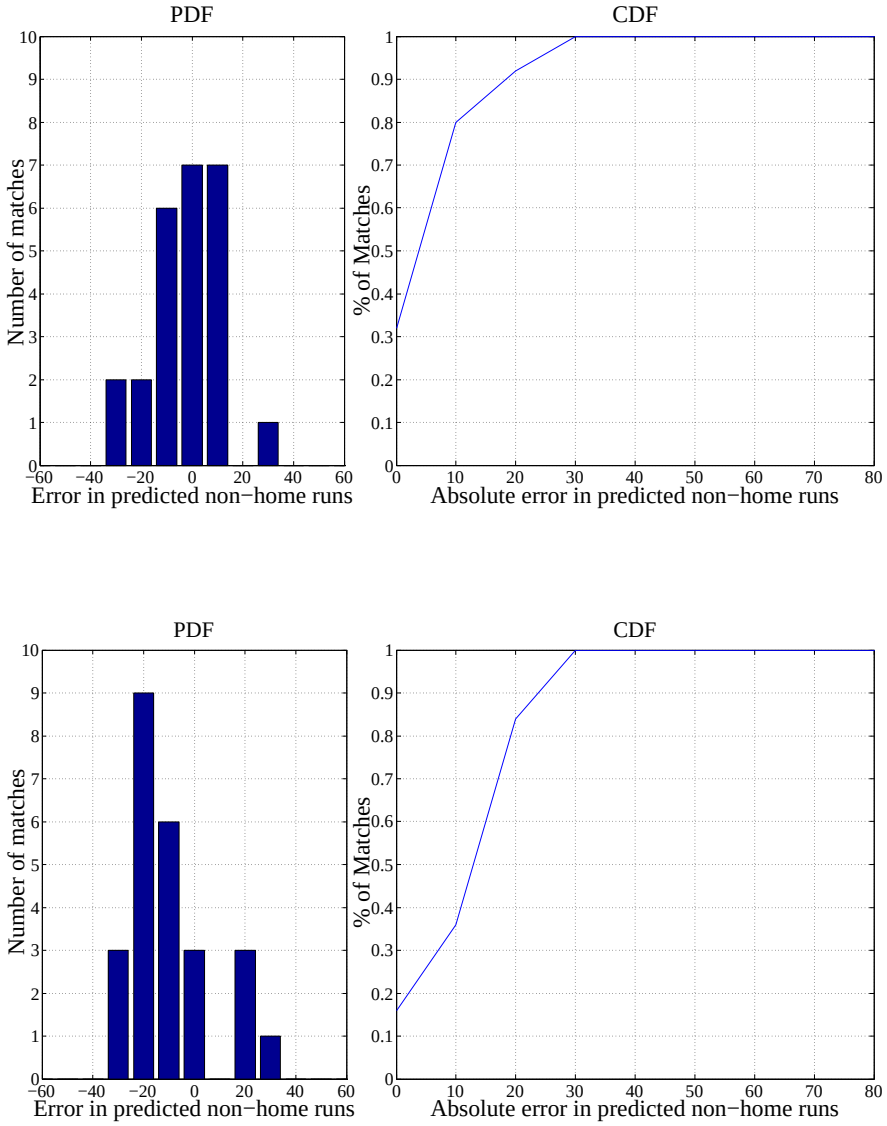

Figure 5.2: PDF and CDF of total non-home run prediction error for $innings_1$ (top) & $innings_2$ (bottom)

In $innings_1$, the batting team tries to accumulate as many runs as they can, regardless of their current score. They do not settle for lesser number of runs as long as there is an opportunity to score. Similarly, the team bowling in $innings_1$ tries to minimize the runs scored by the opponents. Even if history, team strategy or game intuition suggests that 300 is a winning total, they would continue scoring runs at the same rate till they run out of resources. The bowling team would want to give away as few runs as possible (by taking wickets or bowling dot [1] balls) and contain the score from growing, regardless of the current score.

But in the second innings, *target* is the ceiling and the game ends when the score reaches the target (batting team wins) or when one of the resources for the batting team is depleted (bowling team wins). There is no incentive to the batting team for using minimum amount of resources as opposed to consuming entire resources (scoring off the last ball or using the last wicket). Similarly, for the bowling team, the only requirement is to prevent the score from crossing the *target* score. There is no incentive to reduce the runs scored as long as it is practically impossible for the batting team to attain the *target* with the remaining resources. Therefore, availability of *target* directly influences the game plan and makes it slightly unpredictable. This explains the difference in prediction accuracy between $innings_1$ and $innings_2$ of home, non-home and total runs (observed in figures 5.2,5.6 and 5.10)

5.1.1 Segmentwise non-home run prediction performance

Figures 5.3 and 5.4 show the scatterplots for non-home runs prediction for every segment across $innings_1$ and $innings_2$ respectively. It can be observed that the first two and the last two segments have worse correlation compared to the rest of the segments. This is attributed to the unpredictable nature of these periods of the game. Unless the teams come out and play, their strategy, the behavior of the pitch, influence of weather conditions remain unknown. This property of unpredictable starts is observed in many other sports as well. In cricket, teams try to shift their scoring pattern and accelerate the scoring towards the end of their innings. This might lead to fall of many wickets in a short span of time or a steep increase in the scoring rate. Hence the final segments of an innings are also unpredictable.

[1] Balls in which no runs are conceded. The batsmen play the ball but do not score a run. These kind of balls are good for the bowling team

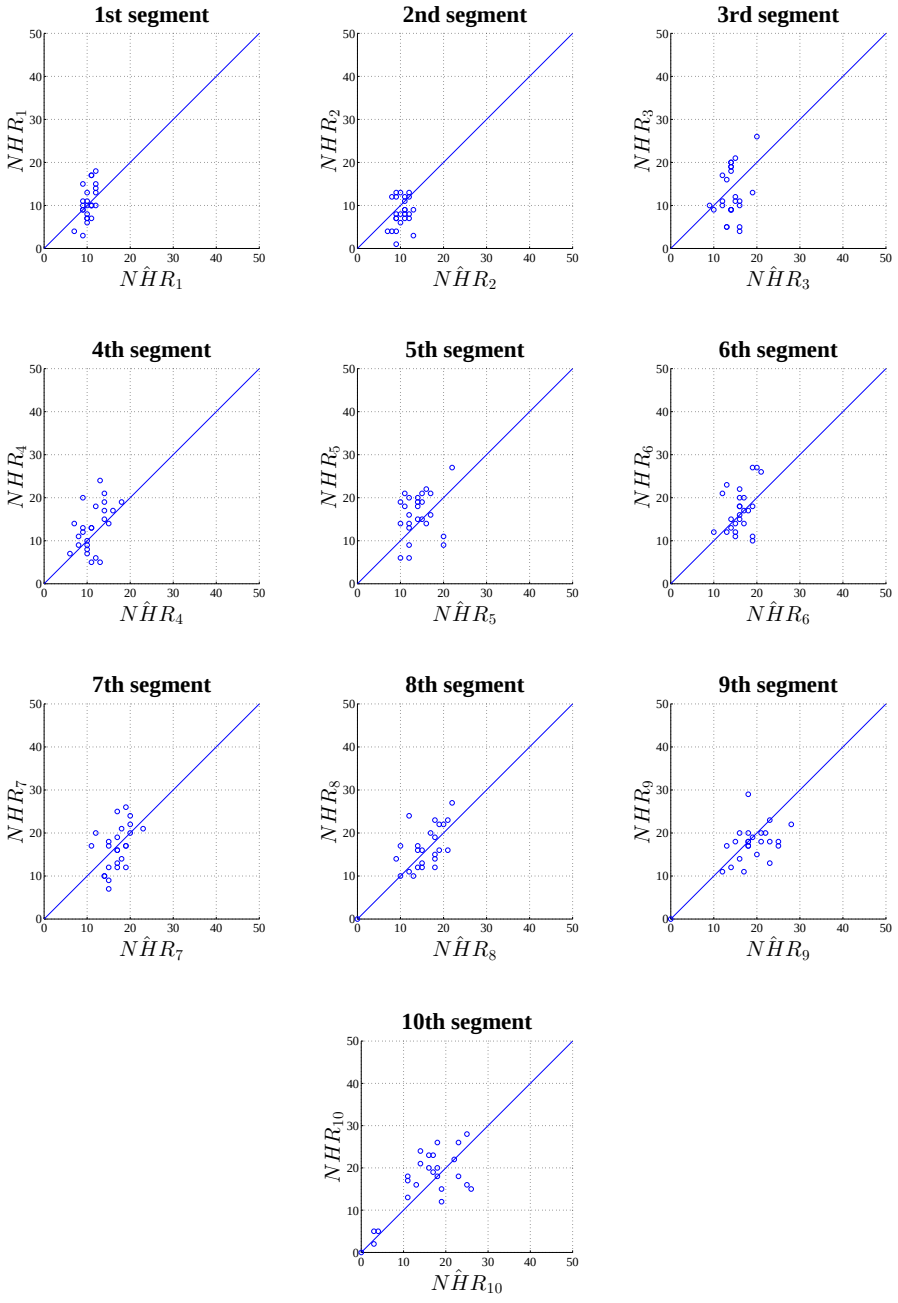

Figure 5.3: *Non-home runs* prediction scatter plot for every segment in $innings_1$

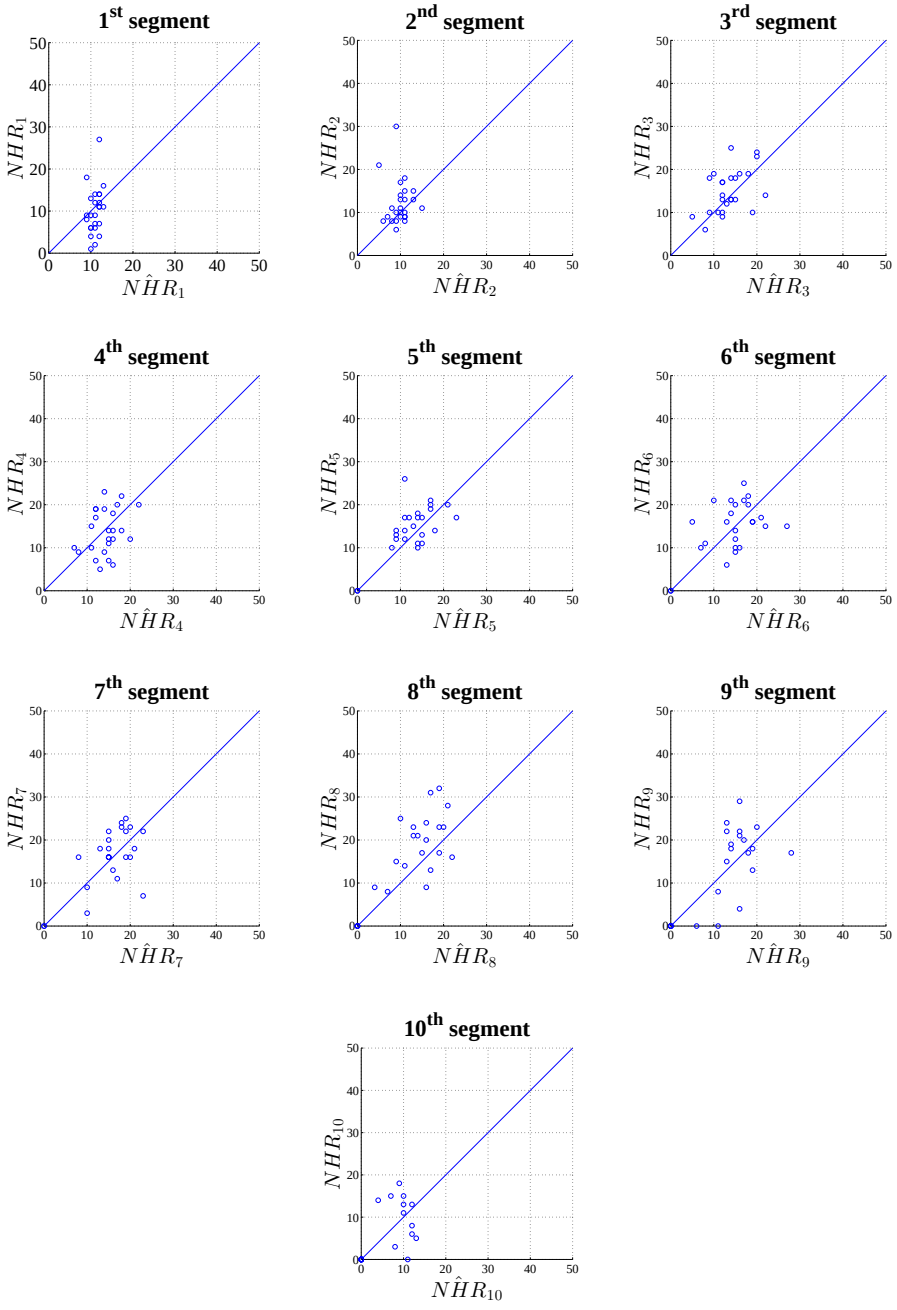

Figure 5.4: *Non-home runs* prediction scatter plot for every segment in $innings_2$

5.2 Home Run Prediction Performance

To predict *home* runs, nearest neighbor aided attribute bagging framework was used as described in section 4.8. Every classifier in this framework choses a subset of l features from the available feature space. This introduces a stochastic component in our algorithm and gives rise to high variance during prediction for the same sample across multiple trials. To mitigate this, we combine predictions made by multiple invocations of the attribute bagging framework. We invoke 200 iterations of the attribute bagging and take the mean of predictions from the iterations as our final predictionThis way the variance is averaged out and there is good degree of consistency across predictions for the same sample. The number *200* was determined to be a good number in terms of running time *vs* accuracy trade-off. A number of distance metrics (namely, Jaccard, Hamming, Spearman and Cosine measures) were tried and compared with the performance of Standard Euclidean distance metric, and the Standard Euclidean metric was observed to perform best (*i.e.*, have the lowest minimum average error).

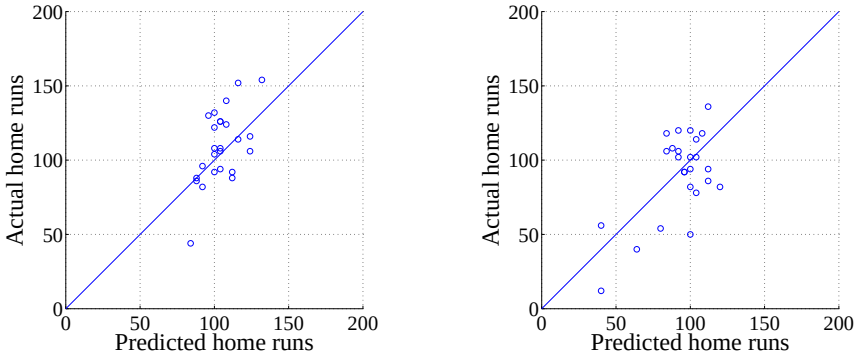

Figure 5.5: Total *home runs* scatter plot for $innings_1$ (left) & $innings_2$ (right)

Figure 5.5 shows a scatter plot between predicted and actual total home runs.

Figure 5.6 shows that 80% and 70% of the matches are predicted with an error margin of less or equal to 20 runs in $innings_1$ and $innings_2$ respectively. As men-

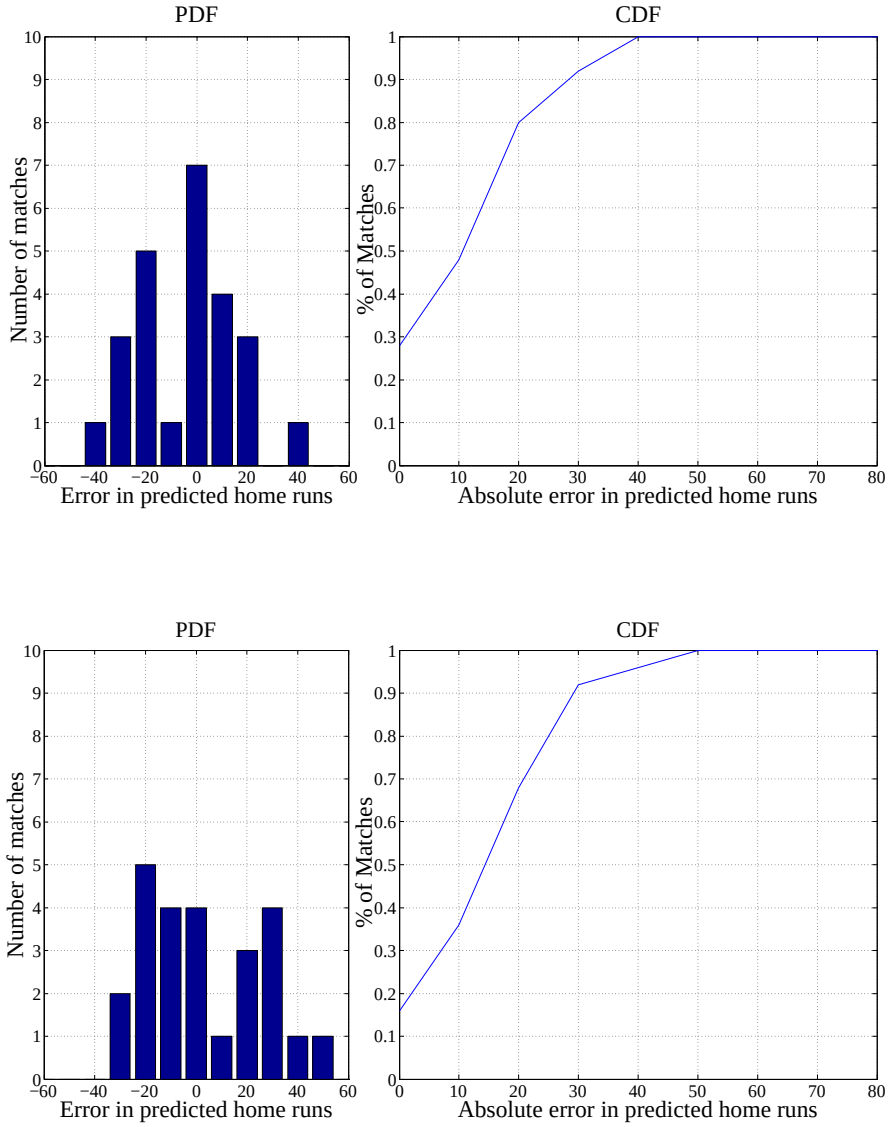

Figure 5.6: PDF and CDF of total home run prediction error for $innings_1$ (top) & $innings_2$(bottom)

tioned in Chapter 3, home runs are awarded either 4 or 6 runs based on where the ball lands. Hence, a single mis-prediction can induce a maximum error of 6 runs. It is also a more difficult problem to predict the number of runs scored through home

runs, with the uncertainty arising from the very nature of the game as described in section 3.1.4. This is reflected in Figure 5.5 and 5.6.

5.2.1 Segmentwise home run prediction performance

Figures 5.7 and 5.8 show the scatterplot for home runs prediction for every segment across $innings_1$ and $innings_2$ respectively. It can be observed that in comparison with segmentwise non-home runs prediction, segmentwise home run prediction has poor correlation. This is because of the fact that we predict number of home run hits in a segment as opposed to number of home runs itself. Then the hits are scaled by 4 units to predict number of runs (Majority of home run hits result in 4 runs). Hence a single mis-prediction results in an error of 4 runs minimum.

5.3 End-of-Innings Run Prediction Performance

Figure 5.9 shows the scatter plot for $\hat{R}_{eoi}$ for every match in the dataset. Figure 5.10 shows the total score error distribution across the all the matches in the dataset. It can be observed that, for 80% of matches, prediction error has a maximum of 20 runs in $innings_1$ and more than 55% for the same prediction error ceiling in $innings_2$. Again, the slightly poor prediction performance in $innings_2$ is attributed to unpredictability of runs in $innings_2$ (as elaborated in section 5.1)

5.3.1 Segmentwise run prediction performance

Figures 5.11 and 5.12 show the scatterplot for total runs predicted for every segment across $innings_1$ and $innings_2$ respectively. As mentioned in section 5.1.1, unpredictable nature of initial and final segments result in the mediocre runs prediction performance in these segments.

5.4 Runs in a Segment, $\hat{R}_i$

Given match data till segment S_{i-1}, prediction of runs scored in the next 5 overs, or the immediate next segment S_i alone is an interesting and important problem. As match data streams in, the model is updated (in five-over intervals) with ground truth and $\hat{R}_i$ is predicted for the next segment. Figure 5.13 shows the mean absolute

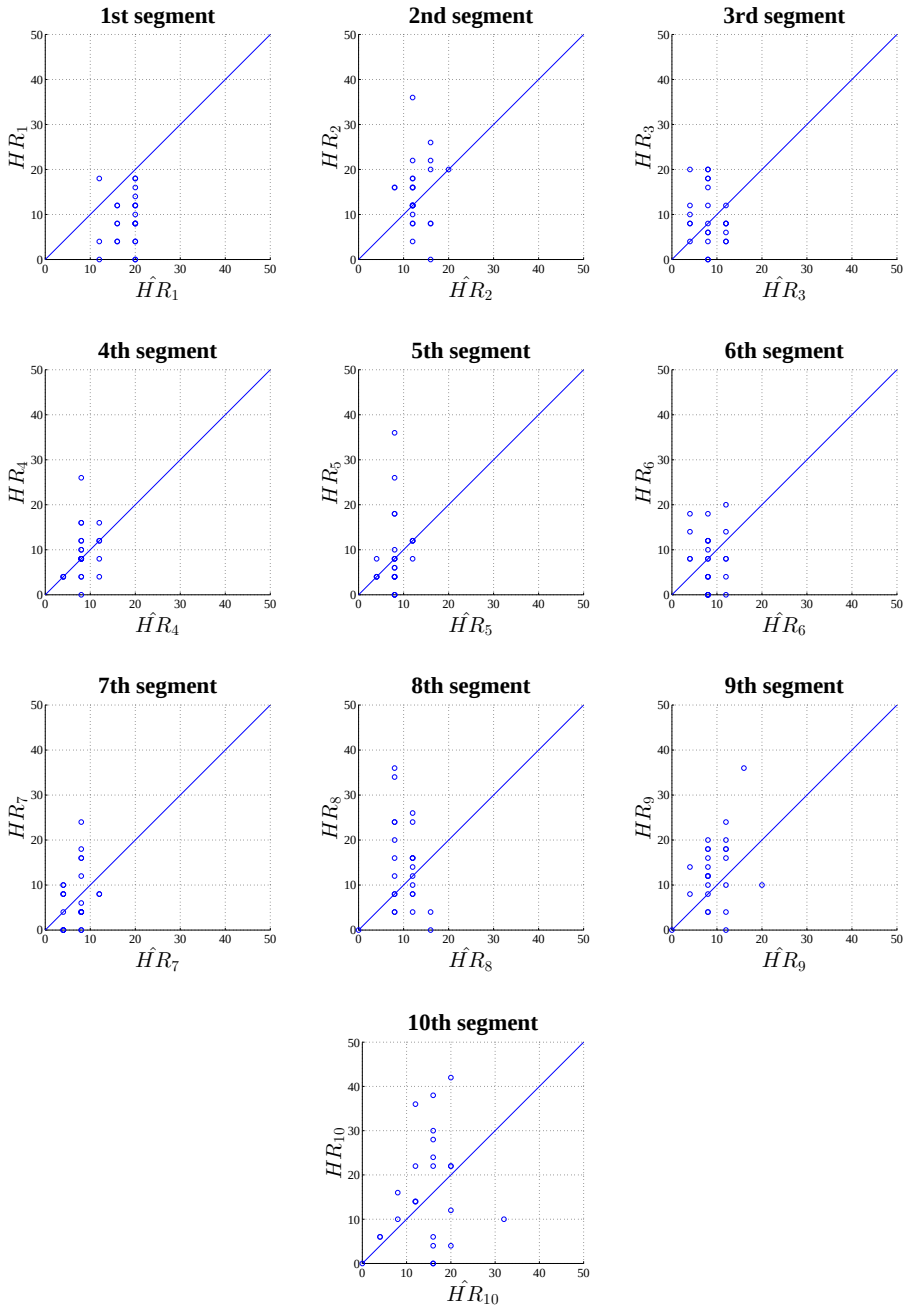

Figure 5.7: *Home runs* prediction scatter plot for every segment in $innings_1$

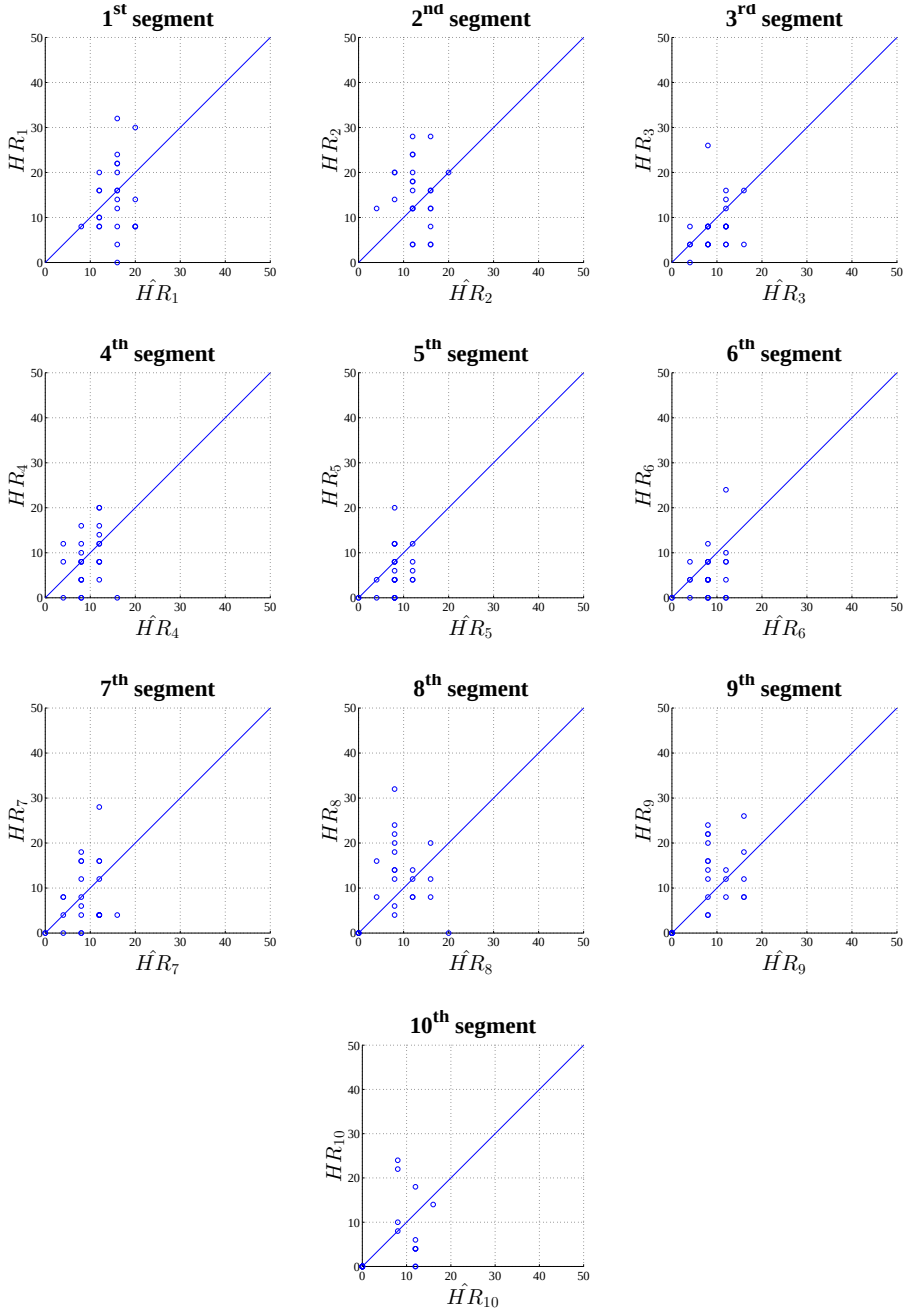

Figure 5.8: *Home runs* prediction scatter plot for every segment in $innings_2$

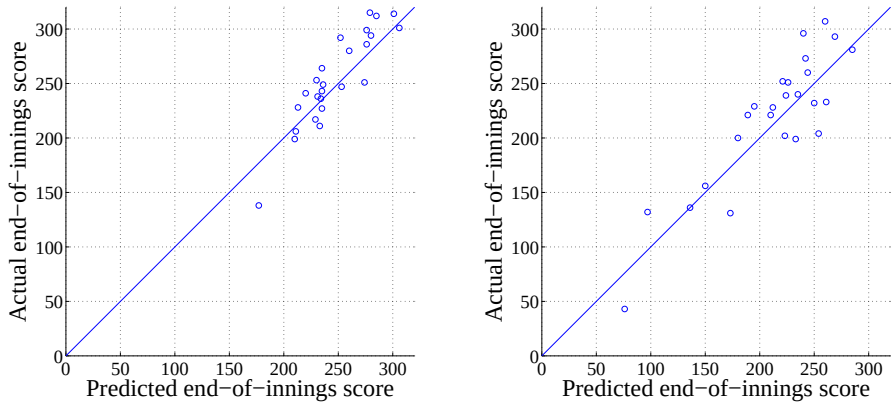

Figure 5.9: $\hat{R}_{eoi}$ scatter plot for $innings_1$ (left) & $innings_2$ (right)

error values and standard deviation for $\hat{R}_i$ scores across segments S_i, given match state data till segment S_{i-1}. MAE_i for both $innings_1$ and $innings_2$ lies within the range of 4 and 12 runs for all the segments, with errors increasing towards the later segments during the innings. Generally, until the middle overs (*i.e.*, up to over 35), teams are focusing on building a good foundation and consolidating their run scoring efforts. On the other hand, in the last 2 or 3 segments (from overs 35 to 50), it is common for the batsmen to try to hit most of the deliveries for home runs to maximize total runs; subsequently, a large chunk of the total score is accumulated in these last three segments. In doing so, batsmen take high risks and subsequently may get dismissed. Hence the match could turn in favor of any of the two teams with more or less equal probability. Because of such unpredictable nature of the game during these segments, it is difficult to estimate $\hat{R}_i$ with high accuracy. Accordingly, the performance of our algorithm suffers somewhat in these segments, as demonstrated by the plots.

5.5 Performance Comparison with Baseline Model

Our model is compared with two other baseline models.

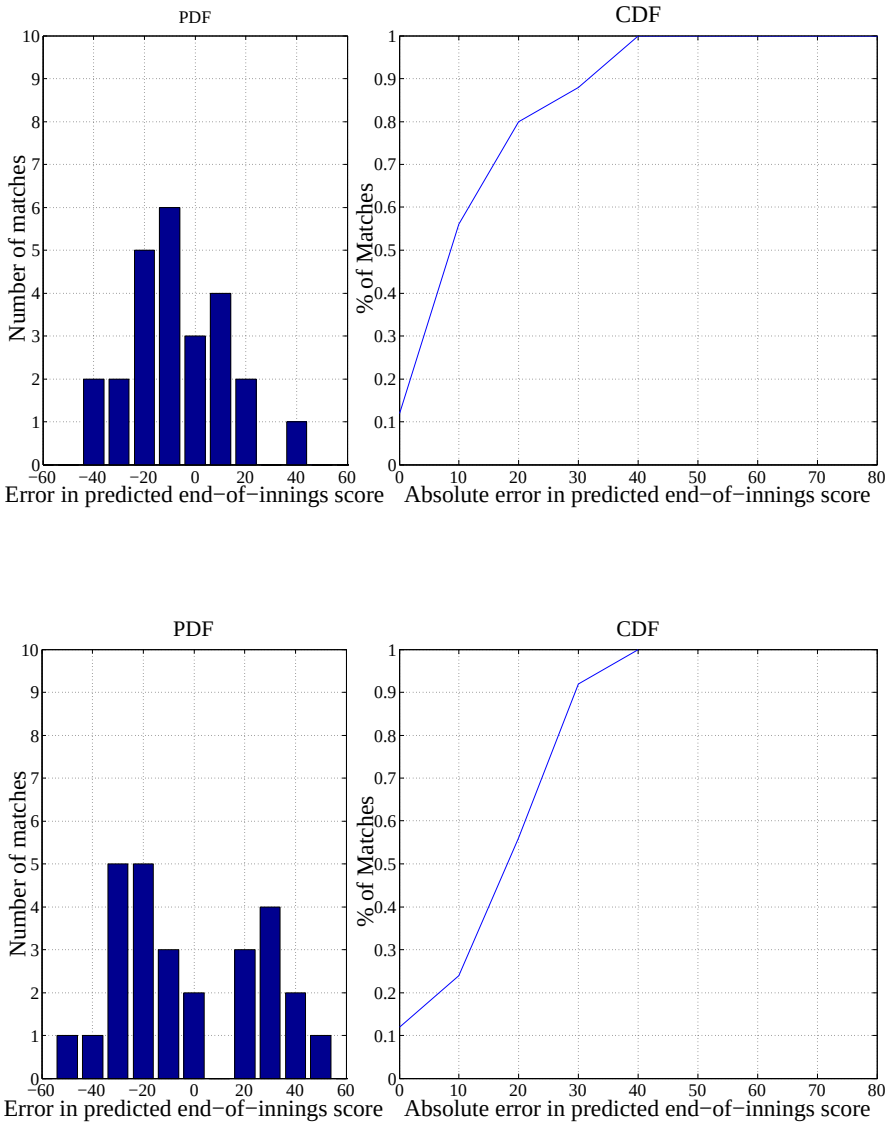

Figure 5.10: PDF and CDF of R_{eoi} prediction error for $innings_1$ (top) & $innings_2$ (bottom)

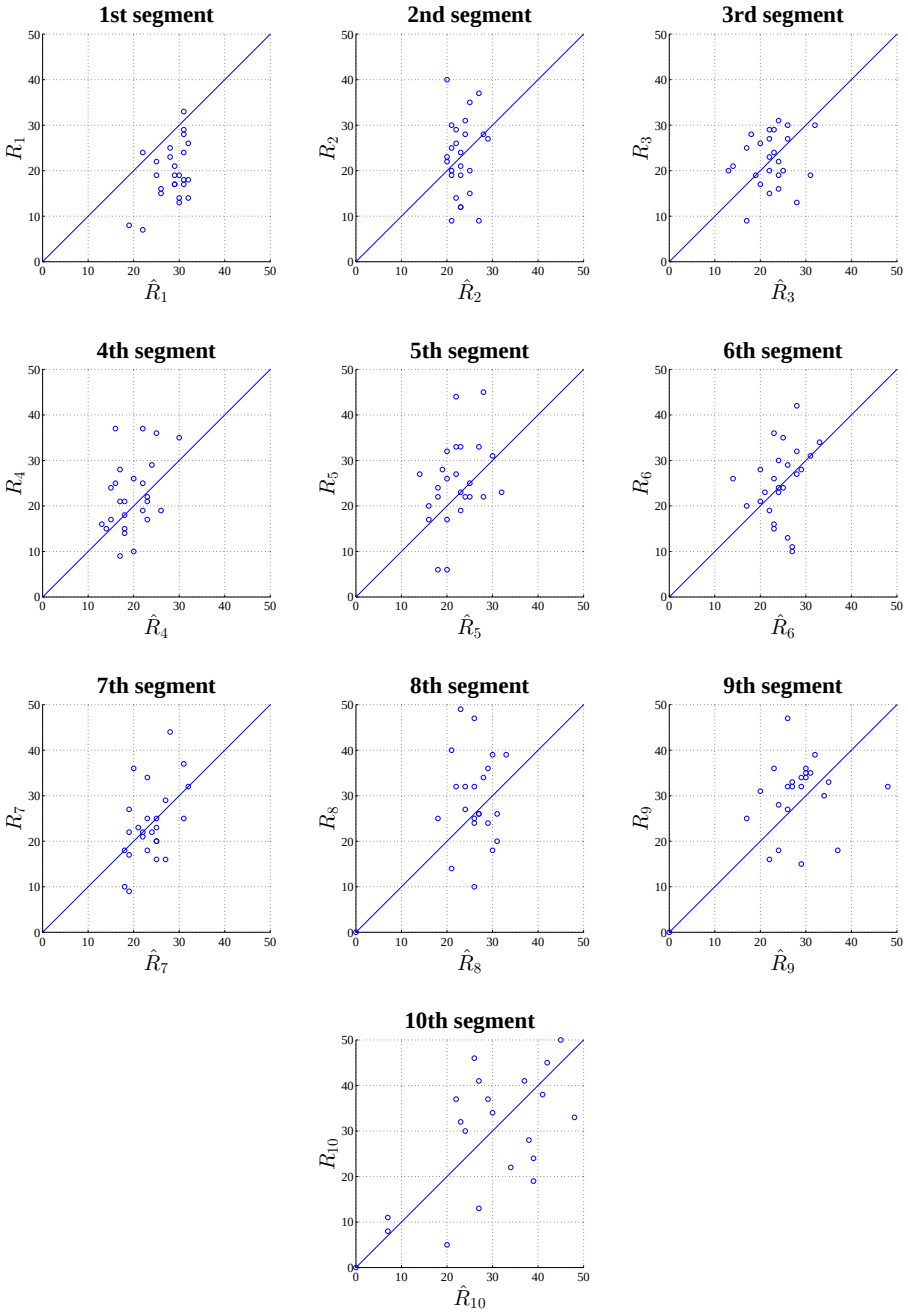

Figure 5.11: Runs in a segment prediction scatter plot for every segment in *innings*$_1$

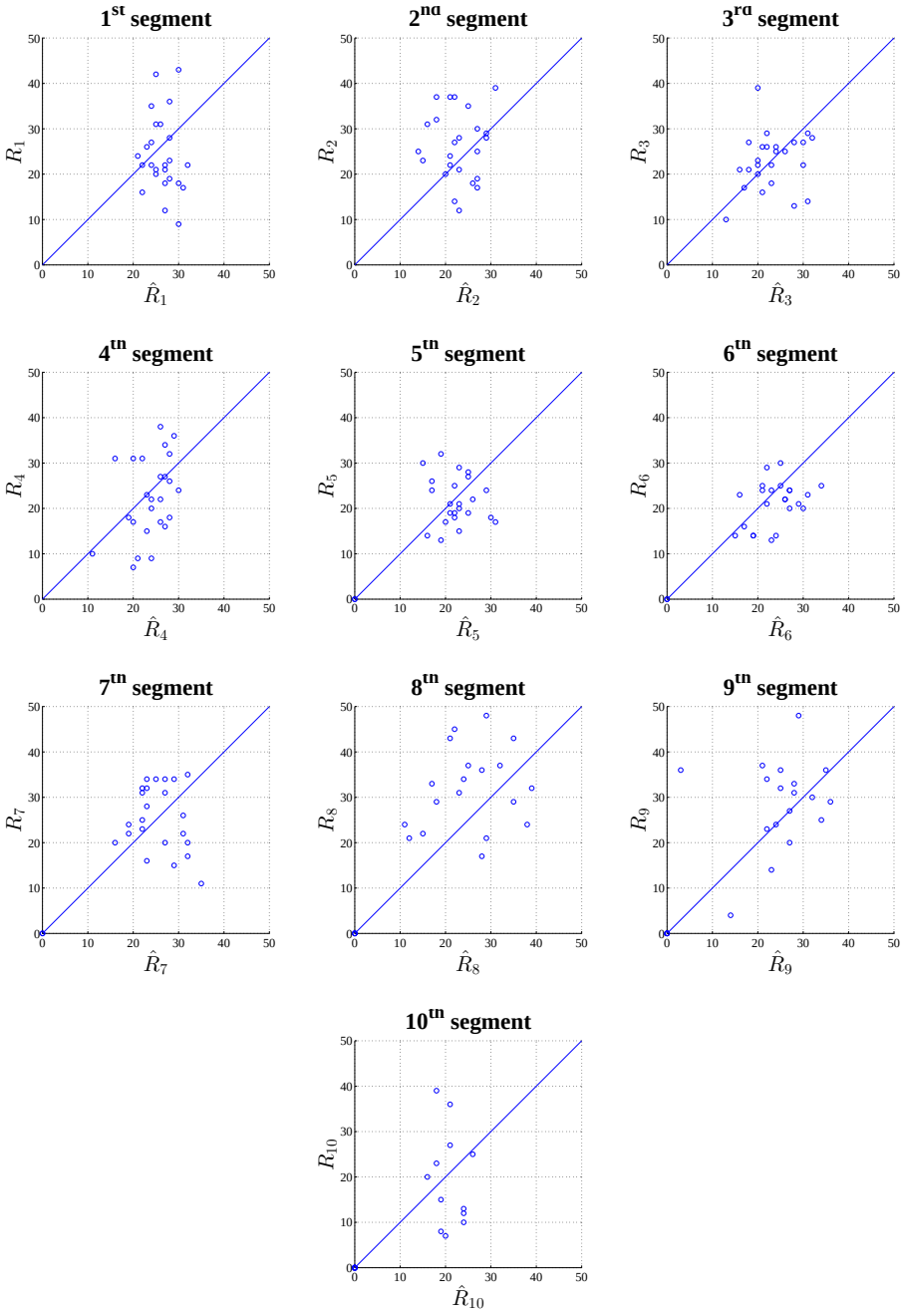

Figure 5.12: Runs in a segment prediction scatter plot for every segment in *innings*$_2$

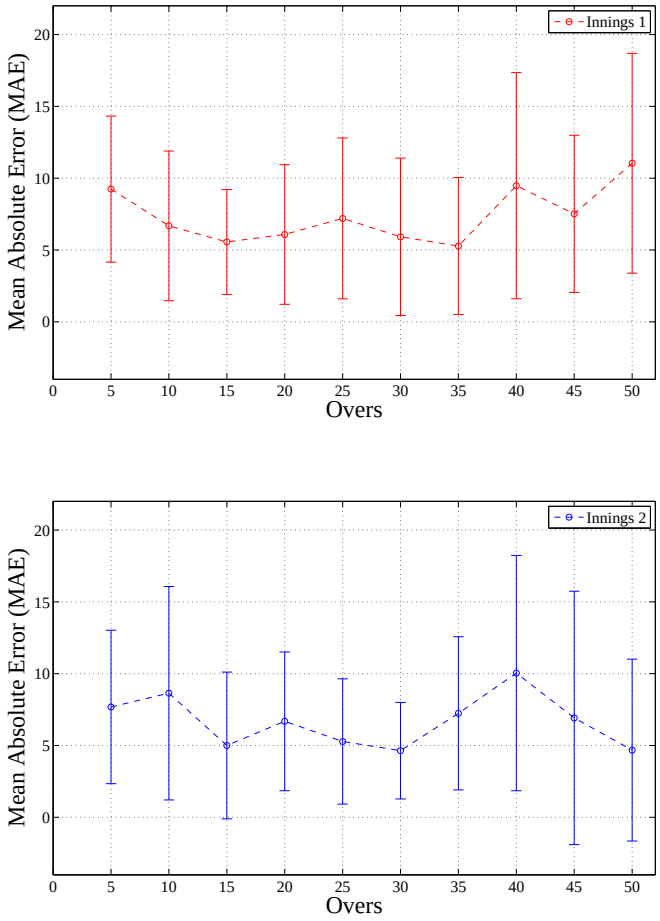

Figure 5.13: Mean absolute error and standard deviation for Runs R_i across each segments S_i, or, 5-over intervals for innings 1 (above) and innings 2 (below). Since the first and fore-most prediction R_1 for $i = 1$ gives the runs scored at the end of over number 5, the plots start from over 5.

5.5.1 Run prediction by Bailey et al.

Bailey et al. [2] propose a model that predicts the $\hat{R}_{eoi}$ of a game in progress which is used to analyze the sensitivity of betting markets. Although addressing a different requirement, their framework allows making $\hat{R}_{eoi}$ predictions at the end of each innings.

5.5.2 ICC projected score prediction model

The first is a proprietary model used by the broadcasters of cricket matches and is telecast to the viewers when the game is in progress. This simple model predicts the end of the innings score ($\hat{R}_{eoi}$) based on the run rate. Run rate of a team is the average number of runs scored per over and is formally defined as, given by,

$$\text{Run rate} = \frac{\text{runs scored till } over_n}{n} \tag{5.1}$$

This run rate is used to extrapolate the score at the end of the innings. The projected score for the innings (consisting of 50 overs) is given as,

$$\text{Projected Score} = \text{run rate} * 50 \tag{5.2}$$

This model implicitly makes the assumption that the teams continue to score at the same rate regardless of their score and the amount of resources left.

In Figure 5.14, we demonstrate the accuracy of our model considering the above two models as a baseline. At the end of each segment, $\hat{R}_{eoi}$ is calculated and compared with the actual R_{eoi} obtained at the end of innings from match data. As shown in the plot, all three models (Bailey et al., ICC Projected Score and ours) make better predictions as more data from the match in progress are input to the models. However, MAE for our model is significantly better for all the segments compared to Bailey et al. and ICC Projected Score. It can be observed that our model significantly outperforms the baselines for both the innings.

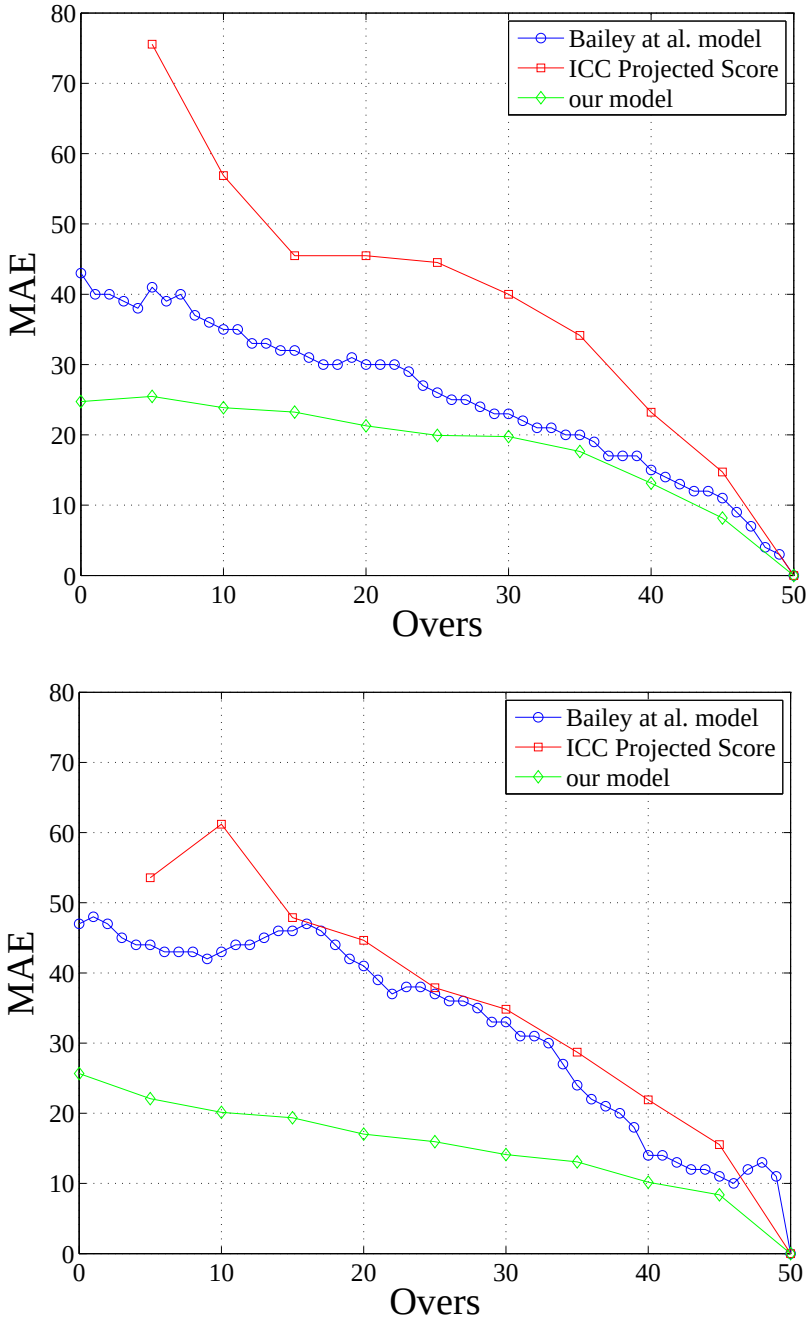

Figure 5.14: Mean absolute error in $\hat{R}_{eoi}$ prediction for innings 1 (top) and innings 2 (bottom) for Bailey et al. [2], ICC Projeted Score prediction and our model.

5.6 Winner Prediction

Our framework was used for predicting $\hat{R}_{eoi}$ for both innings, to predict the game winner. We found that the accuracy of this prediction is just above 70%, which is robust regardless of the number of known segments. To the best of our knowledge, this is the highest winner prediction accuracy reported for ODI cricket.

Chapter 6

Conclusion & Future Work

6.1 Future Work

A model to predict game progression and outcome using historical and instantaneous features was developed in this work. Needless to say, there are many future directions for this work.

6.1.1 Other formats of the game

Our model is tested on the One-Day International format of the sport. *T-20* and *Tests* are the other two formats in cricket where we can extend our model. These formats have slightly different rules (like powerplay, number of overs in an innings, number of innings in a game etc.). Hence teams devise different strategies and pack their team with different kinds of players for each format. Hence it would present, within itself, a challenge to model the format in our algorithm.

6.1.2 Strategy recommendation

Given the historical and the instantaneous match features, our model predicts the progression of remainder of the game. This provides us with a framework for a search space where key decision-making variables like runs to-be-scored, when to take the powerplay, which bowlers to bowl, which batsmen to play etc. of a team predicted to lose the game, could be tweaked continuously till we predict them

to win. This way, our algorithm could recommend strategies automatically and throughout the game as it unfolds.

6.1.3 Wickets prediction

Our model currently predicts the number of runs scored in the innings. A natural direction, though quite challenging one, would be to extend the model to predict the number of wickets lost in the process. However, the sparsity of wicket data on a *per*-segment basis presently poses a significant challenge. A team has ten wickets at its disposal in an innings. When compared with the number of home runs in an innings (which is already difficult to predict due to sparsity), this number is very less. Furthermore, teams rarely lose all their wickets in an innings. Typical values tend to lie between five to eight. Hence extensive investigation is required in order to create a robust and accurate approach to predict wicket falls.

6.2 Conclusion

The main goal of this work is to learn a model for predicting game progression and outcome in one-day cricket. Separate models were developed for home runs and non-home runs using historical features as well as instantaneous match features from past games. Ridge Regression and attribute bagging algorithms are used on the features to incrementally predict the runs scored in the innings. ODI cricket data was collected and the quality and accuracy of our predictions was compared with an extensive set of experiments. In addition to predicting runs for future segments, our winner prediction accuracy is by far the highest reported in ODI cricket mining literature.

Bibliography

[1] P. E. Allsopp and S. R. Clarke. Rating teams and analysing outcomes in one-day and test cricket. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 167(4):pp. 657–667, 2004. ISSN 09641998. URL http://www.jstor.org/stable/3559882. → pages 10, 25

[2] M. Bailey and S. R. Clarke. Predicting the match outcome in one-day international cricket matches while the game is in progress. *Journal of sports Science and Medicine*, 5(4):480–487, 2006. → pages ix, 10, 11, 48, 49

[3] D. Beaudoin. *The best batsmen and bowlers in one-day cricket*. PhD thesis, Simon Fraser University, 2003. → pages 10

[4] I. Bhandari, E. Colet, and J. Parker. Advanced Scout: Data mining and knowledge discovery in NBA data. *Data Mining and Knowledge Discovery*, 1(1):121–125, 1997. → pages 8

[5] H.-H. Bock. Origins and extensions of the -means algorithm in cluster analysis. *Journal lectronique d'Histoire des Probabilits et de la Statistique [electronic only]*, 4(2):Article 14, 18 p., electronic only–Article 14, 18 p., electronic only, 2008. URL http://eudml.org/doc/130880. → pages 6

[6] S. R. Brooker. An economic analysis of ability, strategy and fairness in odi cricket. *Theses and Dissertations*, 2011. URL http://hdl.handle.net/10092/5886. → pages 11

[7] R. Bryll, R. Gutierrez-Osuna, and F. Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36(6):1291–1302, June 2003. doi:10.1.1.83.9733. → pages 5, 28

[8] B. Bukiet, E. R. Harold, and J. L. Palacios. A markov chain approach to baseball. *Operations Research*, 45(1):pp. 14–23, 1997. ISSN 0030364X. URL http://www.jstor.org/stable/171922. → pages 9

[9] F. C. Duckworth and A. J. Lewis. A fair method for resetting the target in interrupted one-day cricket matches. *The Journal of the Operational Research Society*, 49(3):pp. 220–227, 1998. ISSN 01605682. URL http://www.jstor.org/stable/3010471. → pages 10

[10] J. Fewell, D. Armbruster, J. Ingraham, A. Petersen, and J. Waters. Basketball teams as strategic networks. *PLoS ONE 7(11): e47445.*, 2012. doi:doi:10.1371/journal.pone.0047445. → pages 8

[11] K. Fukunaga and P. M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *Computers, IEEE Transactions on*, 100(7): 750–753, 1975. → pages 5

[12] G. Ganeshapillai and J. Gettag. A data-driven method for in-game decision making in mlb. In *MIT Sloan Sports Analytics Conference*, 2014. → pages 9

[13] E. Garcia. Cosine similarity and term weight tutorial. *Information retrieval intelligence*, 2006. → pages 5

[14] G. Gartheeban and J. Guttag. A data-driven method for in-game decision making in mlb: when to pull a starting pitcher. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 973–979, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi:10.1145/2487575.2487660. URL http://doi.acm.org/10.1145/2487575.2487660. → pages 9

[15] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901. → pages 5

[16] A. Kaluarachchi and A. Varde. CricAI: A classification based tool to predict the outcome in ODI cricket. In *Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on*, pages 250–255, 2010. doi:10.1109/ICIAFS.2010.5715668. → pages 10

[17] C.-H. Kang, J.-R. Hwang, and K.-J. Li. Trajectory analysis for soccer players. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, ICDMW '06, pages 377–381, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2702-7. doi:10.1109/ICDMW.2006.160. URL http://dx.doi.org/10.1109/ICDMW.2006.160. → pages 9

[18] H. H. Lemmer. An analysis of players' performances in the first cricket Twenty20 World Cup series. *South African Journal for Research in Sport, Physical Education and Recreation*, 30:71–77, 2008. → pages 10

[19] A. J. Lewis. Towards fairer measures of player performance in one-day cricket. *The Journal of the Operational Research Society*, 56(7):pp. 804–815, 2005. ISSN 01605682. URL http://www.jstor.org/stable/4102181. → pages 10

[20] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL http://CRAN.R-project.org/doc/Rnews/. → pages 6

[21] S. Longley and M. Gipon. In-play betting report. analysis of sports bettings newest battleground by gamblingdata. Technical report, GamblingData, 91 Waterloo Road,London, SE1 8RT, September 2011. URL http://www.gamblingdata.com/files/Final%20In-play%20report_0.pdf. → pages 2, 3

[22] S. Luckner, J. Schröder, and C. Slamka. On the forecast accuracy of sports prediction markets. In *Negotiation, Auctions, and Market Engineering, International Seminar, Dagstuhl Castle*, volume 2, pages 227–234, 2008. ISBN 978-3-540-77553-9. doi:10.1007/978-3-540-77554-6_17. URL http://www.springerlink.com/content/t0156067312n5116/. http://www.odysci.com/article/1010112984064393. → pages 9

[23] D. Lutz. A cluster analysis of NBA players. In *MIT Sloan Sports Analytics Conference*, 2012. → pages 9

[24] D. W. Marquardt. Ridge regression in practice. *The American Statistician*, 29(1):3–20, February 1975. ISSN 0003-1305. doi:10.1080/00031305.1975.10479105. → pages 29

[25] I. G. McHale and M. Asif. A modified Duckworth-Lewis method for adjusting targets in interrupted limited overs cricket. *European Journal of Operational Research*, 225(2):353–362, March 2013. ISSN 03772217. doi:10.1016/j.ejor.2012.09.036. → pages 10

[26] F. Palomino, L. Rigotti, and A. Rustichini. Skill, strategy, and passion: an empirical analysis of soccer. Econometric Society World Congress 2000 Contributed Papers 1822, Econometric Society, Aug. 2000. URL http://ideas.repec.org/p/ecm/wc2000/1822.html. → pages 9

[27] K. Raj and P. Padma. Application of association rule mining: A case study on team India. In *International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–6, 2013. doi:10.1109/ICCCI.2013.6466294. → pages 10

[28] V. Sankaranarayanan, J. Sattar, and L. Lakshmanan. Auto-play: A data mining approach to odi cricket simulation and prediction. In *Proceedings of the SIAM International Conference on Data Mining*, pages 1064–1072, 2014. doi:10.1137/1.9781611973440.121. → pages iii, 6

[29] C. Spearman. The proof and measurement of association between two things. By C. Spearman, 1904. *The American journal of psychology*, 100 (3-4):441–471, 1987. ISSN 0002-9556. URL http://view.ncbi.nlm.nih.gov/pubmed/3322052. → pages 5

[30] T. B. Swartz, P. S. Gill, D. Beaudoin, and B. M. deSilva. Optimal batting orders in one-day cricket. *Comput. Oper. Res.*, 33(7):1939–1950, July 2006. ISSN 0305-0548. doi:10.1016/j.cor.2004.09.031. URL http://dx.doi.org/10.1016/j.cor.2004.09.031. → pages 25

[31] T. B. Swartz, P. S. Gill, and S. Muthukumarana. Modelling and simulation for one-day cricket. *Canadian Journal of Statistics*, 37(2):143–160, 2009. ISSN 1708-945X. doi:10.1002/cjs.10017. URL http://dx.doi.org/10.1002/cjs.10017. → pages 11

[32] P. O. S. Vaz de Melo, V. A. F. Almeida, A. A. F. Loureiro, and C. Faloutsos. Forecasting in the NBA and other team sports: Network effects in action. *ACM Trans. Knowl. Discov. Data*, 6(3):13:1–13:27, Oct. 2012. ISSN 1556-4681. doi:10.1145/2362383.2362387. URL http://doi.acm.org/10.1145/2362383.2362387. → pages 8