Inferring Tennis Match Progress from In-Play Betting Odds : Project Report

James Wozniak
4th year Information Systems Engineering MEng
jhw107@ic.ac.uk

Supervised by Dr. William Knottenbelt

June 23, 2011

Abstract

This project examines the way in which tennis matches are modelled to obtain the probabilities of a each player winning a match in progress. By taking live market data from the betting website Betfair, the live odds are analysed and compared against those generated from the derived tennis model. Based on this comparison, an algorithm is described to infer the scoreline of the match from the Betfair data. This algorithm is then improved by detecting breaks in play, and using them to correct any errors in the score prediction.

Acknowledgements

I would like to thank Dr. William Knottenbelt for providing the idea behind the project and for providing valuable insight, guidance and motivation. I would also like to thank Professor Peter Harrison for his suggestions and help in reviewing the progress of the project. Lastly, I would like to thank my parents for their financial and morale support throughout my education.

Contents

1 Introduction

Over the last few years, the growth and widespread use of the Internet has helped revolutionise the betting industry. The Internet offers a more efficient and convenient way to wager money, as well as opening the door for large scale betting exchanges and 'in-play' markets. As a result, large volumes of money are now wagered continuously throughout the duration of many sporting events.

Analysing the betting exchange 'Betfair', a significant singles match of professional tennis can often see several millions of pounds traded between users[1]. One match analysed in this report from the 2011 French Open Final, saw over 25 million pounds traded. With the large volume of bets being placed, this project primarily aims to investigate how sensitive and adaptive the market is to the current progress of a tennis match. The overall goal is to see if the betting market has grown to the point whereby the score of a match can be inferred from the odds offered on a particular exchange.

In order to relate the fluctuations in the live betting odds to the progress of a tennis match, a mathematical model is developed under the assumption that each point played is independent and identically distributed (iid). The model provides a way of calculating the probability of a player winning the match from any possible scoreline based on a measure of the players relative skill level. Also discussed how is the 'in-play' odds are collected and what information is available for analysis.

The main investigation of the project is then performed with the aid of the tools created. The odds extracted from Betfair are compared against those generated by the model for three different matches. Based on these observations several methods are suggested to try and infer the score. These methods are described, tested and evaluated, giving reasons for any successes or failures.

2 Background

2.1 Tennis Scoring System[2]

Tennis has a rather unusual and complicated scoring system and the exact scoring rules vary between different competitions and with the gender of the players. The basic system consists of a hierarchical model, with points required to win games, games required to win sets, and sets required to win the match.

2.1.1 Service

A player is allowed a single fault per point when on service. Performing two faults on the same point forfeits the point. If a player loses the game on his service, then it is referred to as a 'break'. This is considered to be a significant event due to the advantage of having service.

2.1.2 Games

Scoring within a game goes incrementally from 0, 15, 30, 40 and then game. If both players reach 40 it is referred to as 'deuce'. From deuce, winning a point gives a player 'advantage'. Winning a point with the 'advantage' wins the game, whilst losing with 'advantage' returns the game to deuce. The server remains constant throughout any given game and alternates between games. The server of the first game of the match is determined by a coin toss.

2.1.3 Sets

The winner of the set is based upon the total number of games won by each player during the set. Generally speaking, the set is won if a player has won six or more games, and is at least two games clear of his opponent. If both players reach six games all, then a tie-break occurs to determine the winner of the set. If both players can win the match by winning the current set, then it is known as the final set.

In accordance to the rules of the particular tournament or competition, professional tennis matches are played either as the best of three sets (first player to win two sets) or the best of five (first player to win three sets). There are also varying rules about the final set, and whether or not the match can enter a tiebreaker. In the case where there is no final set tiebreaker (common in major tournaments), the final set will continue past 6-6 as normal until a two game advantage is achieved.

2.1.4 Tie-breaks

A tie-breaker requires a player to score seven or more points, and have a two point lead against the opponent. For example, a tie-break can be won by 7-5, 7-2 or 9-7. The winner of the tie-break determines the winner of the set by a score of seven games to six. Service is rotated between both players. The player who served the first game of the set is given service for the first point. The next two points are served by the other player, and service is swapped every two points until a winner emerges.

2.2 Betting odds

The odds for a particular event or outcome are formulated as a ratio of two integers, usually written as

$$n/m$$

where n and m represent the relative chance of the outcome not occurring and occurring respectively. A bet of m currency units will then give a profit of n units if successful. For example, the odds of getting a six on a dice roll are 5/1 and if £10 is staked, the profit will be £50.

It is also common for betting exchanges, such as Betfair, to use a decimal representation of odds. Given the probability of an event occurring, p, the decimal odds are calculated as follows

$$Decimal\,odds = \frac{1}{p}$$

For example, if the probability of an outcome is 0.25, then the decimal odds will be 4. Given this figure, it is easy to calculate the return of the bet

$$return = units\,staked * decimal\,odds$$

The profit can be determined by subtracting the stake itself from the returns, i.e. by subtracting 1 from the decimal odds and multiplying. In this case, you would win 3 times the stake if successful.

2.2.1 Sources

Traditional bookmakers are a source of up to date and 'in-play' betting odds. However, they do not offer what can be termed, 'actual' odds. That is, their odds are not representative of the true probability of a certain outcome. Although arguably the probability of a certain outcome is subjective and can never be defined, they will deliberately try and offer slightly worse odds than what is being predicted to statistically ensure a profit. Without knowing or estimating their desired profit margins, these odds are of little use.

Fortunately, as mentioned in the introduction, the amount of people gambling on-line has allowed betting exchanges to grow rapidly in popularity. A betting exchange allows users to place bets amongst each other, with users able to assume the role of a bookmaker. As the exchange odds are decided by the users, we can assume that they more closely represent the 'actual' odds and the general consensus of the user base as to a particular outcome. In this project, the betting exchange 'Betfair'[1] will be used due to its popularity and wide spread use, especially for tennis betting. It also offers an API service to aid the collection of market data.

The API[3] uses the Simple Object Access Protocol (SOAP). The API both sends and receives XML documents over an HTTP connection. The structure of these XML documents is defined by the SOAP protocol. Betfair offers sample code in Java, with libraries able to manage the HTTP connections and handle the construction of the XML documents in accordance to the appropriate specifications. For this reason, Java will be used to extract information from Betfair. The free version of the API service will allow 60 requests for the latest odds in a minute, which will provide a good enough resolution to keep accurate track of the live odds.

2.3 Statistical Modeling

Unlike many sports, tennis lends itself well to statistical modelling. This is because a match of tennis essentially involves the repeated situation of one player serving a point against the other. This makes it much easier to analyse the performance of a player from a quantitative perspective. As a match essentially consists of many single points, one way to quantify the relative skill levels between two players is to estimate the probability of a player winning a point on service when playing against the opponent.

From Wimbledon data, referenced by Magnus and Klaasen [11], an average men's match (best of five sets) consisted of roughly 230 points. With a large number of tournaments being played throughout the year there is a large amount of statistical data that can be analysed to make predictions. There are also other indicators of performance which can be used to infer the relative strength of each player. For example, the official tennis world rankings[4] and seedings for a particular tournament.

2.3.1 Stochastic Markov chains

A common assumption used in modeling a game of tennis is that each point is independent and identically distributed (iid) [6][8][7]. That is, the chance of a player winning a point is not in any way dependent on the outcome of the point beforehand and the probability of a player winning a point on his serve can be assumed constant throughout the duration of a match.

The validity of this assumption was analysed by Klaassen and Magnus[12]. Certain phenomenon were shown to occur which can be justified from a psychological perspective and which violated the iid assumption. For example, if the previous point was won, it gave a positive effect on the current point through psychological momentum. However, the conclusion was that variation from the iid assumption was small, and the idd assumption can still be utilised to calculate the chance of a player winning a match with good effect.

Under the iid assumption, a game of tennis can be modelled as a stochastic Markov chain. A Markov chain has the property that the next state depends only on the current state, and thus conforms to the independence property. Given that the probability of a player winning a point on his serve is assumed to be constant (identically distributed) a Markov chain can be constructed, with different states representing different scorelines. Figure 1 shows a Markov chain representing the outcome of a particular game in which the server has a probability, p, of winning a point on his serve. *Note: some states are statistically equivalent and have been merged.*

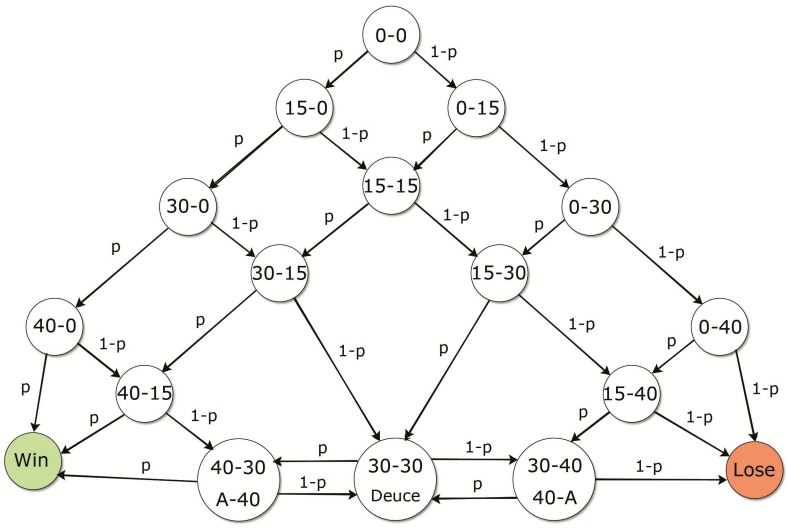


Figure 1: A Markov Chain of a tennis game

2.4 Tennis formulas

2.4.1 Game formula

Given a probability of a player winning a point on service, it is possible to create a formula to give the chance of winning the game based on the Markov chain assumptions. To calculate the overall win probability, you can take the sum of the probabilities that the game is won 4 points to 0,1 or 2 and then the probability of reaching 3-3 and then going on to win from deuce. Coefficients are then required to capture all the different possible traversal paths.

It can be shown that the probability of winning a game on service, $G(p)$, is given by the following formula as derived by O'Malley[8], where p = the probability of winning a point on service:

$$G(p) = p^4 + 4p^4(1-p) + 10p^4(1-p)^2 + 20p^3(1-p)^3 \times p^2 \sum_{i=3}^{\infty} 2p(1-p)^{i-3} \quad (1)$$

Note - when the game is at deuce, it can theoretically go on for a infinite amount of time. It is necessary to sum the probabilities of winning with each possible outcome i.e. winning deuce after 10,100 or 1000 points. Therefore, an approximation of a summation to infinity is used.

$$G(p) = p^4 + 4p^4(1-p) + 10p^4(1-p)^2 + \frac{20p^5(1-p)^3}{1-2p(1-p)} \quad (2)$$

Using similar assumptions and input parameters, Barnett, Brown and Clarke[9]

utilised the properties of a Markov chain to derive a recursive formula to calculate the probability of winning from any state within a game. Essentially, the probability of winning from any state in the Markov chain can be achieved by summing the probabilities of reaching either of the two child nodes, multiplied by their own probability of reaching the 'win' state. The formula given is as follows, where a and b represent the game score of player A and B.

$$P(a,b) = p\,P(a+1,b) + (1-p)P(a,b+1) \tag{3}$$

With the following boundary conditions:

$$P(a,b) = 1 \quad if \quad a = 4, b \leq 2$$
$$P(a,b) = 0 \quad if \quad b = 4, a \leq 2$$
$$P(3,3) = \frac{p^2}{p^2 + (1-p)^2}$$

2.4.2 Set and tie-breaker formula

As the scoring system in tennis is a hierarchical one, the above formula can be used in a calculation to determine the tennis formula for winning a set, and similarly, the set formula can be used to calculate the probabilities of winning a match. Calculating the probability of winning a set is complicated due to the rotation of service. To obtain the probability of winning a set, the easiest way is to use a similar recursive approach, again outlined by Barnett, Brown and Clarke[9]

- Let $P_A^S(c,d)$ and $P_B^S(c,d)$ be the probability of player A/B winning a set from game score (c,d) where player A/B is serving
- Let P_A^G and P_B^G be the probability of player A/B winning a game on service
- Let P_A^T and P_B^T be the probability of player A/B winning a tiebreaker.

The probability of winning a set is then given by

$$P_A^S(c,d) = p_A^G P_B^S(c+1,d) + (1-p_A^G)P_B^S(c,d+1) \tag{4}$$

With the following boundary conditions:

$$P_A^S(c,d) = 1 \quad if \quad c = 6, 0 \leq d \leq 4 \;\; or \;\; c = 7, d = 5$$
$$P_A^S(c,d) = 1 \quad if \quad d = 6, 0 \leq c \leq 4 \;\; or \;\; d = 7, c = 5$$
$$P(6,6) = P_A^T$$

Using a similar approach, formulas can be derived for the case whereby the final set cannot go to a tiebreaker, and for the tiebreaker itself. The tiebreaker formula, however, requires the win on serve probabilities rather than the game win probabilities. Thus, to fully determine the probability of winning a set which could include a tiebreaker, both probabilities are required.

2.4.3 Match formula

A Markov chain can also be constructed to describe the process of winning a match from different set scores. O'Malley uses the same approach to the game formula in deriving the following match formulas. Given the probability of winning a set, p^s, the following formula is given for the chance of winning the match for the best of three sets.[8]

$$p = (p^s)^2 + 2(p^s)^2(1 - p^s) \tag{5}$$

and for the best of five sets

$$p = (p^s)^3 + 3(p^s)^3(1 - p^s) + 6(p^s)^3(p^s)^2 \tag{6}$$

This is a slight oversimplification, as the probability of winning a set can vary in the final set depending on whether or not a tiebreaker is played. This is easily changed by considering that when the set score reaches either 1-1 in the three set or 2-2 in the five set equation, a new value for p^S should be used to model the probability of a set being won without being able to go to a tie-break.

To calculate the chance of winning from any set scoreline, as with the other formulas, there is a recursive equation described by Barnett, Brown and Clarke [9] In this case, the final set is assumed to be without a tiebreaker.

- Let $P^M(e, f)$ be the probability of player A winning the match from set score (e,f)
- Let p^{sT} be the probability of player A winning a tiebreaker set
- Let p^s be the probability of player A winning a non-tiebreaker final set

The probability of winning a match from score (e,f) in a five set match is then:

$$P^M(e, f) = p^{sT}P^M(e + 1, f) + (1 - p^{sT})P^M(e, f + 1) \tag{7}$$

With the following boundary conditions:

$$P^M(e, f) = 1 \quad if \quad e = 3, f \leq 2$$
$$P^M(e, f) = 0 \quad if \quad f = 3, e \leq 2$$
$$P^M(2, 2) = p^s$$

2.5 Independence of points

There has been significant research into the validity of the iid assumption. Intuitively it is difficult to imagine that the probability of a player winning a point is identically distributed, even though we can say that it is a good assumption over the course of an entire match. In a paper by Magnus and Klassen[11], many hypothesis are tested using data from the Wimbledon Championships.

Examples such as the 'first game effect' suggest that a player is less likely to have his serve broken in the first game of any match, a direct invalidation of the iid assumption. In fact, there are many reasons why we might expect to see a variation in the win on serve percentages. For example, in a long match, as the players tire the impact of their serve may decrease. The ability of a player to cope with pressure on high importance points could also be another issue.

Another common phenomenon often seen in sport is the idea of psychological momentum, which implies winning previous points can have a positive mental effect on the match. To test the idd assumption, Brown, Barnett and Clarke[9] investigated the mean number of games played for each match played at the 2003 Australian Open. The results suggested that the idd assumption gave a 7 percent overestimate of the number of games you would expect to see for a 5 set match, and a 7 percent underestimation for a three set match.

A revision to the model was added to increase the chance of a player winning a point having already won the previous point. By capturing the momentum of a player, the mean number of games more closely matched the actual results seen and hence suggests that by altering the probability of a player winning on service whilst the match is in play can offer a more realistic prediction. Overall, we can expect the odds taken from Betfair to vary according to such factors.

3 Obtaining in-play betting odds

As mentioned previously, the Betfair website is different to a lot of other bookmakers in the sense that it acts as a mediator for users to place bets with each other. It is therefore essential to explicitly state how much money is being offered, and at what odds. Their business model also makes it attractive to offer other useful pieces of information which can inform betting decisions and help increase market volume.

In this section, we look at what betting information is available and how it is queried and processed. We also detail the exact information that is recorded in order to make score predictions.

3.1 Getting data from Betfair

The information displayed on the Betfair website can be queried using the Betfair API. Although both paid and free versions are available, the free version is sufficient for the purposes of this project, allowing 60 requests per minute for each type of request that will need to be issued. An application was created around the Betfair API sample code to easily record the status of the betting market for a particular match throughout its duration.

3.2 Information Available

There were two main function calls that were used to retrieve the necessary data. These were *getMarketPricesCompressed()* and *getMarketTradedVolume()* and each was called every second whilst data was being recorded.

getMarketPricesCompressed() provided, amongst other things, the current bet delay (used to tell when the market has gone in-play - there is typically a 5 second bet delay for all events in-play), the total amount matched on the event, the odds at which the last money was traded (known as the last price matched) and a list of how much money is being offered to back/lay[1] a player and at what odds.

getMarketTradedVolume() gave, for each player, how much money had been traded at each odds boundary.

3.2.1 Virtual bets and cross matching

When looking at the amount of money being wagered, it is noticeable that for almost all matches there is a disproportionate amount of money being traded for each player. In fact, the majority of money traded is done by people backing or laying the favourite. This is a common trend for almost all events on Betfair.

Based on this, it would seem that it would be difficult to place bets with reasonable odds on the non-favourite to win or lose. However, Betfair employs a cross matching system which creates the opportunity to place 'virtual bets'. By considering virtual bets, we get a more accurate view of the current market. Unfortunately, this feature is not included in the data obtained from the Betfair API, although cross-matching is used on their website. Cross-matching calculations were therefore carried out on the raw data.

For tennis, there are only two outcomes of a match so cross-matching is simple. This is due to the fact that placing a bet for a certain player to win is logically equivalent to placing a bet for the other player to lose. If, for example, there is £100 offered to back player A at odds of 1.5, then this is equivalent to offering

[1] 'backing' a player corresponds to placing a bet for the player to win. 'laying' a player refers to placing a bet that the player will not win

to lay player B. To get the odds which will be offered for player B, we can use the following formula.

$$odds = \frac{1}{1 - (1/originalodds)} = \frac{1}{1 - (1/1.5)} = 3 \tag{8}$$

To get the amount of money that should be offered as these odds, we consider that £100 will give a profit of £50 at odds of 1.5. Therefore, this intuitively maps to someone placing a £50 bet on player B, with the chance to win £100. Therefore, a £50 'virtual bet' will be offered at odds of 3 should someone wish to back player B.

Further work is required if the inversed odds lie outside of the accepted range of values allowed by betfair. For example, odds of 1.41 are allowed, but will map to 3.439 which is not. In this case, it appears that Betfair offer a bet at worse odds (i.e. 3.4) and keep the extra money that should come from the slightly higher odds if the virtual bet is taken.

The same principle can also be used to create a more accurate last price matched value. The Betfair API returns the last priced matched value for each player, but this can be misleading if the majority of money is being wagered over one player. Therefore, cross-matching is used again to get the most up-to-date and a more realistic last priced match for each player.

3.3 Odds logging

The information that was logged consisted of the following, all inference of the score performed later is done based on the data presented here.

Time - The exact time the odds were retrieved
Total matched - The total amount matched on the event
Bet delay - The delay faced when placing bets. This is always 0 before the match has started, and is commonly 5 seconds whilst the match is in-play. This is used to infer the moment the match went in-play.

Matched volume differences (for both players) - By looking at differences between the data retrieved from successive *getMarketTradedVolume()* calls, if there has been any money wagered over a player, the odds at which it was traded and the amount traded is recorded. The last price matched for each player can be inferred from this information. When there are several possibilities, the odds which saw the most money traded are used.

Example log entry: [29.68@1.17|] [130.9@6.8|58.3@7.0|]

(£29.68 traded at odds of 1.17 on player one, £130.9 and £53.3 traded at odds 6.8 and 7 on player two since the last update)

Best odds available (for both players) The best available back/lay odds for a player are recorded, along with the amounts offered for each case. Only the best three odds available for backing and laying a player are taken.

Example log entry for player one:

[35389.21@6.8|41.59@6.6|3771.65@6.4|][657.99@7.0|6079.99@7.2|39278.16@7.4|]

(£35389.21 offered to lay player one at odds of 6.8, £657.99 offered to back player one at odds of 7 etc.)

3.4 Projected Odds

For a given match, each player will have a pair of odds consisting of the best available back and lay price. That is, the highest odds you could get if you are willing to back a player, and the lowest odds you could get if you wanted to lay the player. Typically, any money traded will be over these values.

When the market is stable, i.e. before the match begins. The projected odds are used to estimate the 'actual' odds, that lie between the pair of values mentioned. Assuming that if the 'actual' odds of a player winning are, say, 4.03, we can imagine that many people will want to try and back the player with odds of 4.1, while less will want to offer to lay the player at odds of 4.0. The projected odds are calculated by taking into account the amount offered at the boundaries, and are used to more accurately specify the odds right before the match starts.

4 Tennis Model

We know already that tennis lends itself well to statistical modeling. In much of the literature on match prediction, tennis models are created using the assumption that all points are independent and identically distributed. It is likely that such models are utilised on Betfair and will be influencing the live change in odds. It is therefore essential to derive our own tennis model using similar principles. This section describes how a model was created to provide an estimation of the odds that you would expect to see for a particular scoreline within a match.

4.1 Inputs to the model

As a match consists of the repeated action of one player serving against the other, the input parameter for each player is the probability of a player winning

a point on service (PWOS) against the other. This figure encompasses the relative skill levels of each player and is assumed to remain constant throughout the remainder of the match. These inputs can be changed as the match progressed to capture effects such as momentum.

The other inputs to the model are the details about the match type. Clearly, the chance of winning a match will vary if the match is played as the best of three or five sets. In fact, the longer matches favour the stronger opponent. Similarly, it is important to know if the final set can go into a tiebreaker, although the impact of this is more limited.

4.2 Game formula

The game formula aims to calculate the probability of a player winning a game on service, from any possible game score. To calculate the probability from any state in the Markov chain, it is possible to use a recursive formula by Barnett, Brown and Clarke, described in the background section. However, to avoid the computational overhead associated with recursion, an iterative formula was derived based on the method used by O'Malley.

The Markov chain shown in background section (Figure 1) shows all possible states within a game. The formula given by O'Malley captures all possible paths from state 0-0 to the 'win' state or to the 'deuce' state. If the starting score was in fact 30-0, then we can redraw the Markov chain showing only the reachable states (Figure 2) and a similar formula could be derived to give the probability of winning from 30-0.

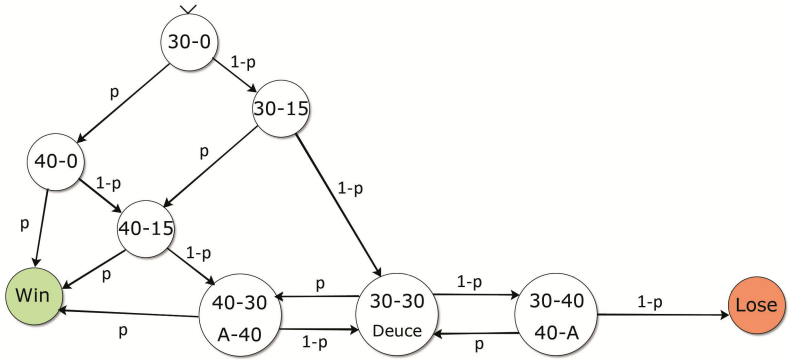


Figure 2: A Markov Chain of a tennis game, with a starting score of 30-0

Essentially, to calculate the chance of winning from any state, you need only consider the reachable part of the chain, and the remaining pathways to the 'win' or 'deuce' states. A formula for any starting state (n,m) can be given

as follows. It is then easily implemented using a for loop in Java and can be optimized by pre-calculating the coefficients.

$$P^G(n,m) = \sum_{i=0}^{2-m} \frac{(3-n+i)!}{(3-n)! \times i!} p^{4-n} q^i + \frac{(6-n-m)!}{(3-n)! \times (3-m)!} p^{3-n} q^{3-m} p(d) \quad (9)$$

Where $p(d)$ is the probability of winning from deuce, p is the probability of winning a point on service and $q = 1 - p$. It is also necessary to specify n and m in terms of the number of points won (eg 1-0, 3-3), and not the traditional tennis scores of 15, 30, 40 etc. Also, A-40 and 40-A are equivalent (analytically) to scorelines 3-2 and 2-3 and the latter must be used.

An example case where player A has a win on serve probability of 0.6227 yields the following probabilities.

Score	0	15	30	40	game
0	0.781	0.629	0.417	0.177	0
15	0.873	0.757	0.563	0.284	0
30	0.944	0.875	0.732	0.456	0
40	0.986	0.962	0.899	0.732	-
game	1	1	1	-	-

Table 1: A table showing the probabilities of a player winning a game from any possible scoreline. The opponents score is shown on the horizontal axis

4.3 Set formula

Creating an iterative formula for the set probabilities is more complicated. This is due to the fact that service alternates between the two players. For example, from the state 0-0 games each, there are many ways to reach, for example, state 6-4. Player one could break once, and player two not break at all, player one could break twice and player two once etc. To cover all possibilities you would then need to consider the permutations for each scenario (i.e. player breaks in the first game, second game and so on).

For simplicity, the recursive approach outlined by Barnett, Brown and Clarke was used for calculating set probabilities[9]. To help speed up the computation, memoization was used. This involved caching already calculated results in an array to remove the need for any recursive function calls. This required initializing all absorbing states in the Markov chain with either 0 or 1 (depending on if the scoreline represented the player having already won or lost the set) and iterating the Markov chain from bottom to top such that the probabilities of both the child nodes were calculated before the parent node.

There was also a special case for state 6-6. As mentioned earlier, the rules of tennis are that the set will be decided by a tiebreaker (provided it is not the final set of a match specified not to have a final set tie-break). The probabilities for a tiebreaker were calculated in a similar manner to those of a set due to their resemblance and these had to be calculated before the set probabilities. The Markov chain for the tiebreaker was made finite using an infinite summation approximation as used similarly in the game formula for the deuce state.

For a match between two players A and B, with probabilities of winning on serve as 0.6227 and 0.6359 respectively, the chances of player A winning a set (Table 2) and tie-break (Table 3) from different score-lines are as follows (assuming player A serves the first game of the set).

score	0	1	2	3	4	5	6
0	0.456	0.232	0.178	0.05	0.026	0.002	0
1	0.518	0.459	0.213	0.153	0.032	0.011	0
2	0.764	0.528	0.462	0.187	0.12	0.014	0
3	0.83	0.799	0.54	0.466	0.149	0.072	0
4	0.961	0.871	0.844	0.555	0.47	0.092	0
5	0.987	0.984	0.925	0.907	0.576	0.473	0.093
6	1	1	1	1	1	0.58	0.478

Table 2: A table showing the probabilities of a player winning a set from any possible game score. The opponents score is shown on the horizontal axis.

score	0	1	2	3	4	5	6	7
0	0.478	0.334	0.251	0.173	0.076	0.022	0.006	0
1	0.566	0.48	0.387	0.232	0.108	0.052	0.016	0
2	0.716	0.641	0.481	0.307	0.206	0.115	0.025	0
3	0.845	0.739	0.587	0.483	0.365	0.169	0.04	0
4	0.91	0.831	0.769	0.688	0.484	0.247	0.11	0
5	0.959	0.939	0.91	0.811	0.628	0.486	0.303	0
6	0.993	0.989	0.97	0.922	0.877	0.806	0.486	0.177
7	1	1	1	1	1	1	0.673	0.486

Table 3: A table showing the probabilities of a player winning a tiebreaker from any possible score. The opponents score is shown on the horizontal axis.

4.4 Match formula

A formula for deriving the probability of winning a match from any state can be derived using similar principles to those used in the game formula and hence avoids the need for recursion or memoization. Whilst the probability of winning a game, set or tiebreaker is independent of the match type, the match formula requires extra inputs. Specifically, the number of sets required to win the match and whether or not the final set will go to a tiebreaker. From a set score of (n,m), the formula is as follows.

- Let k be equal to 1 in the case of no final set tiebreaker and 0 otherwise
- Let p^{sT} be the probability of the player winning a tiebreaker set
- Let s be the number of sets required to win the match (2 or 3).

For the case $k = 0$ the following formula gives the full result.

$$P(n,m) = \sum_{i=0}^{s-2-m+k} \frac{(1+s-1-n)!}{((s-n-1)! \times i!)} (p^{sT})^{s-n} (1-p^{sT})^{i} \tag{10}$$

If $k = 1$, then the result from the previous formula is added to the result of the following formula (to deal with the differing probabilities of winning a tiebreaker set and a non tiebreaker set).

- Let j be equal to 4 if $s = 3$ and 2 otherwise
- Let p^{s} be the probability of the player winning a non tiebreaker set

$$\frac{(j-n-m)!}{(s-1-n)! \times (s-1-m)!} (p^{sT})^{s-n-1} (1-p^{sT})^{s-1-m} p^{s} \tag{11}$$

Although the formulas look complicated, this is largely due to number of conditions required when working out the permutation coefficients (i.e. number of possible paths to the 'win' state). Essentially, you take the number of sets player one requires to win, the number of sets player two can win without winning the match, and sum the probabilities of all possible combinations that result in player one winning the match.

Table 4 shows some example probabilities for a best of 5 set match involving players with win on serve probabilities 0.65536 and 0.6452. The probabilities are given without a final set tiebreaker and with (without/with).

score	0	1	2	3
0	0.553/0.552	0.357/0.355	0.148/0.147	0/0
1	0.729/0.728	0.543/0.542	0.28/0.278	0/0
2	0.895/0.895	0.778/0.777	0.531/0.528	0/0
3	1/1	1/1	1/1	-/-

Table 4: A table showing the probabilities of a player winning a match from any possible set score.

4.5 Service faults

On service you are allowed one 'fault' if the first serve breaks any rules. If a player receives two faults on a single point, then the point is forfeit. For this reason, tennis players commonly play two types of serve. On their first serve, a stronger but less reliable serve is attempted. If it fails, then a second serve, which is far more reliable but generally easier to return is performed.

The probability of winning a point on a first serve is increased due to the difficultly of returning a more powerful serve, and the increased possibility of an ace. The receiving player can often be put immediately on the back foot with a good serve. A second serve is commonly easier to return, and so some of the advantage gained by having service is lost.

Until now, we have assumed a single value for the percent chance of winning a point on service. However, if a fault occurs this is not always the case and the model should take this into account. To do this, the input parameters to the model are changed to the following.

- $P(noFault)$ -The probability a player's first serve not faulting
- $P(win|noFault)$ -The probability a player will win the point if the first serve does not fault
- $P(win|fault)$ -The probability a player will win the point on his second serve

The probability of winning a point on service is therefore given by

$$P(win) = P(noFault)P(win|noFault) + (1 - P(noFault))P(win|fault) \quad (12)$$

For an example player with $P(noFault) = 0.70$, $P(win|noFault) = 0.64$ and $P(win|fault) = 0.55$, then

$$P(win) = (0.70 \times 0.64) + (1 - 0.70)0.55 = 0.613$$

If the first serve faults, however, this becomes

$$P(win) = 0.55$$

A service fault can therefore have a moderate effect on the probability of winning the match. For more important points, we can speculate that service faults should be significant enough to warrant a noticeable change in the betting odds. By incorporating faults into the model, we can estimate how the odds would change if a fault occurred and potentially infer service faults from the live market odds.

4.6 Implementation of the Model

As the scoring system of tennis is a hierarchical Markov chain, to calculate the probability of a player winning the entire match requires using a combination of the aforementioned formulas. This typically involves the following procedure

- Let $P(cGame)$ is the probability of winning the current game
- Let $P(cSet)$ is the probability of winning the current set

$$\begin{aligned} P(cSet) &= P(cGame)P(win\ set \,|\, current\ game\ won) \\ &\quad + (1 - P(cGame))P(win\ set \,|\, current\ game\ lost) \\ P(win\ match) &= P(cSet)P(win\ match \,|\, current\ set\ won) \\ &\quad + (1 - P(cSet))P(win\ match \,|\, current\ set\ lost) \end{aligned}$$

There are several variants to this formula, for example where the match is in a tiebreaker. However, the same general principle holds. We first consider the effect of the next point, and then consider how the point can affect the game/tiebreaker, set, and match. With the notification of a service fault, then the probability of winning the next point is simply changed to $P(win|fault)$.

Once the model was finished, a GUI (Figure 3) was created to provide an easy to use interface and a possible gambling tool.

4.7 Simulator

Under the same assumptions as the mathematical model, a less elegant way to calculate the odds from a particular scoreline would be to 'simulate' the outcome of a match using random number generation. Given the same input parameters (match type, current scoreline and probability of the players winning a point on their respective serve) it is possible to compare the probabilities of a player winning a point against a uniformly distributed random variable in order to determine which of the two players wins the current point. This can be repeated many times, keeping track of the current score until the match is eventually

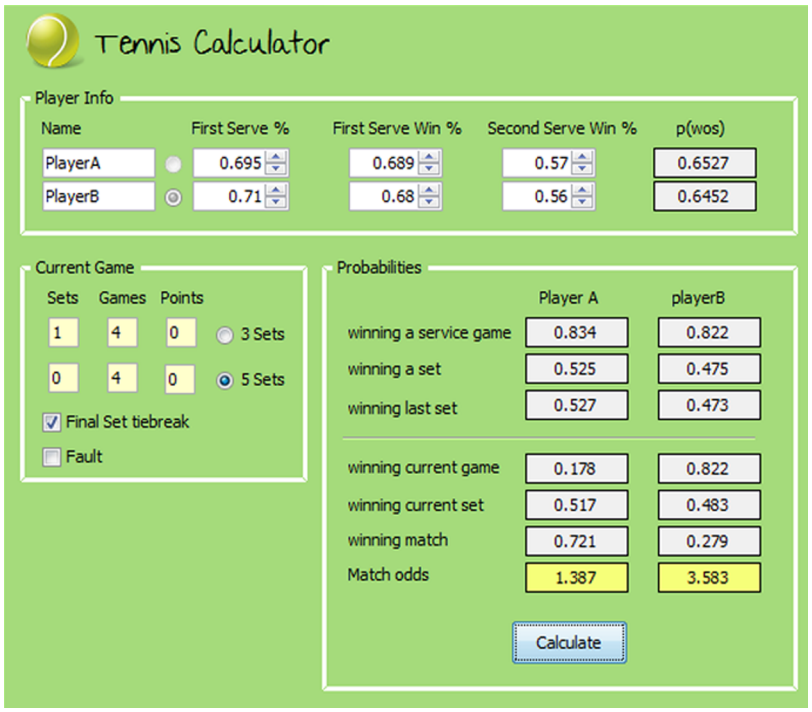


Figure 3: Tennis Calculator GUI

finished.

If multiple matches are simulated, you can get an accurate estimation for the proportion of matches you would expect a player to win from any starting scoreline or input parameters. However, there are obvious disadvantages to this method. The main problem is the large amount of computation required to get accurate results. Given that the accuracy of the result is important, one could suggest that tens of thousands of matches should be simulated from each possible starting scoreline. In such a time critical application where knowing how the odds will change is important, it does not make practical sense to simulate every result and a faster way to generate probabilities on the fly is necessary.

A simulator, however, was created and used to some capacity when testing the system. Due to the nature of the program, testing was extremely important as an incorrect figure could result in mis predictions in score and even erroneous bets. The complex nature of the tennis scoring system and mathematical formula meant it was difficult to guarantee that the mathematical models were in fact giving the desired results. Fortunately, creating a simulator was more straightforward and was thus used to compare results against those derived from the mathematical model.

The simulator itself can be configured to any type of tennis match (3/5 sets, final set tiebreaker etc.) and for any input parameters. For a given set of input probabilities, the simulator iterates through every reachable state during a tennis match and performs a specified amount of match simulations. Several simulations are run from each state to cover all possible match types (e.g. the inclusion or omission of a final set tiebreaker) and the situations where there has or has not been a service fault on the current point.

It is safe to assume that if the results are correct for one set of plausible input probabilities, then the model can be verified for all possible variations of the input probabilities. After changes were made to the mathematical model, the output was compared to the simulated results and it was ensured that they remained similar to within an arbitrarily small number. This process helped eliminate several minor errors such as missed boundary conditions and thus proved to be useful. It also gave a necessary level of confidence in the model.

4.8 The importance of a point

The importance of a point is considered a measure of the extent to which the point will influence the chance of each player winning the match. Due to the unique scoring system of tennis, the importance of points can vary massively throughout the duration of a match depending on the current scoreline. This obviously has a direct influence on the odds expected for each player after the point is decided.

For example, in a match where both players are tied at two sets and four games all, any point could be considered critical. If the score of the current game was *30-40, then the server(indicated by the asterisk) would be facing a break point. In the event of losing the point, it would seem very likely that the entire match would be lost. If we assume both players are identically matched (same win on serve percentage of 0.65) then the derived model would speculate odds of 2.964 before losing the point, and 12.283 afterwards. Clearly, this is a high importance point and we would expect to see a large jump in the odds from Betfair.

The other extreme would be if a player was already two sets, and four games up against the opponent. The match is, for all intents and purposes, already over. The chance of the other player getting back into the game is so low that in most cases bets would no longer be taken on the event. With a similar scenario to that above, the odds would change from 1.013 to 1.026. This would seem to make inferring the score more diffcult, as the change in odds would be extremely small.

5 Comparison between Model and Live Odds

In this section, several matches from the 2011 French Open are examined. The point by point score of each game was recorded, and fed into the derived mathematical model to determine the odds for each player as the match progressed. These odds were then graphed against the last priced matched data retrieved from Betfair and comparisons were made to see how closely the derived model matched the market data.

In each case, the win on serve percentage input parameters of the model were estimated such that the initial odds generated by the model were as similar as possible to the projected odds calculated from the Betfair data at the very start of the match. It is then also taken into account that the probability of winning a serve is generally lower for womens matches and higher for mens. These parameters remain constant throughout the entire match.

5.1 Schiavone V Li

The first match discussed is the 2011 Womens French Open final between Schiavone and Li, in which Li beat Schiavone two sets to zero. Going into the match, it was deemed to be a very close encounter. The projected odds before the match began were 2.144 for Schiavone and 1.875 for Li, making Li the favourite with a 53.33 percent chance of victory. During the match over 15 million pounds was traded making this a fairly highly traded match and potentially a more stable market.

The model used assumed that the probability of Schiavone winning a point on service was 0.582, and similarly 0.5884 for Li. To see a greater variation in odds, we plot Schiavone's last priced matched against time. We also ignore the last few points of the match, where the odds spike upward and distort the graph. (Figure 4 overleaf)

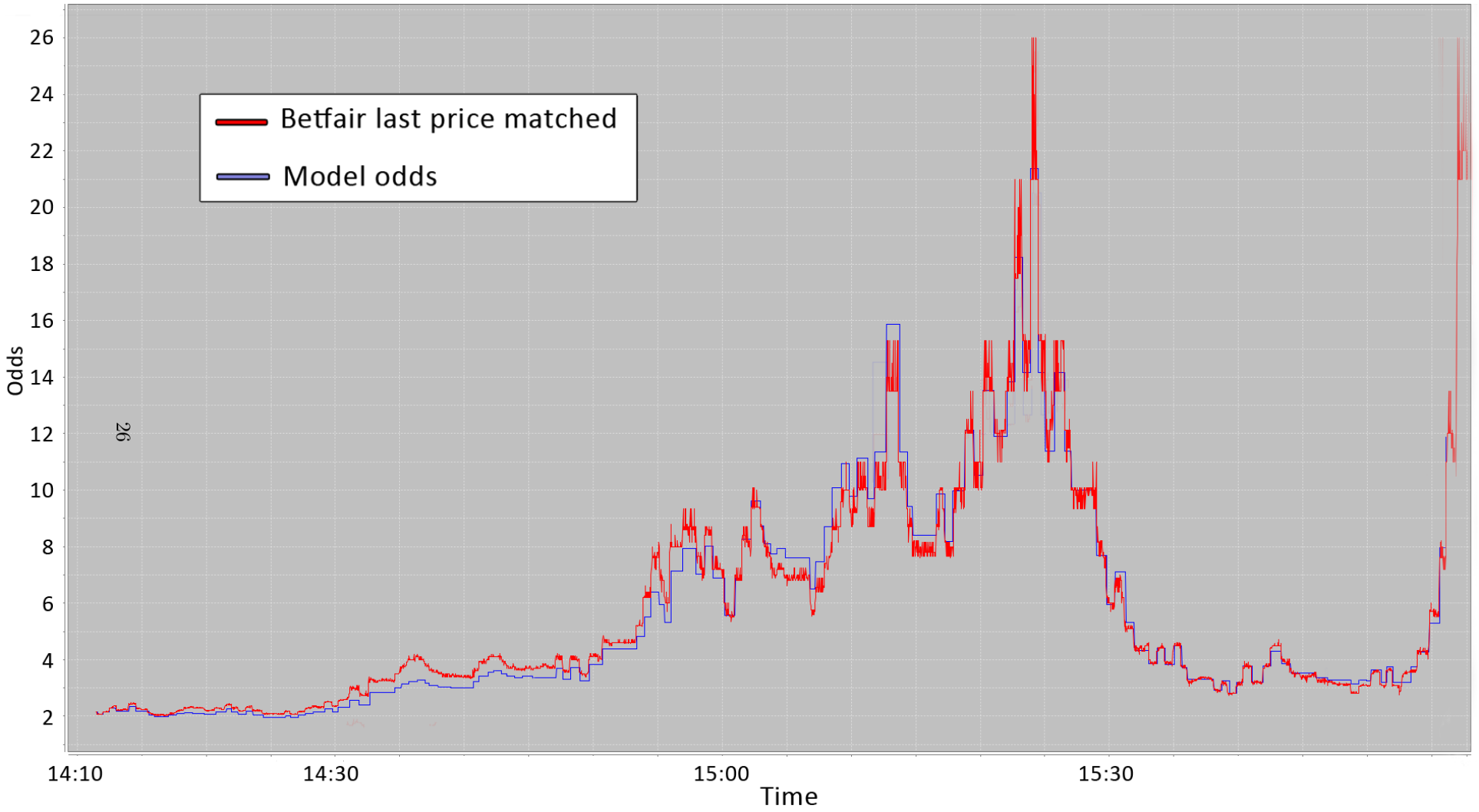
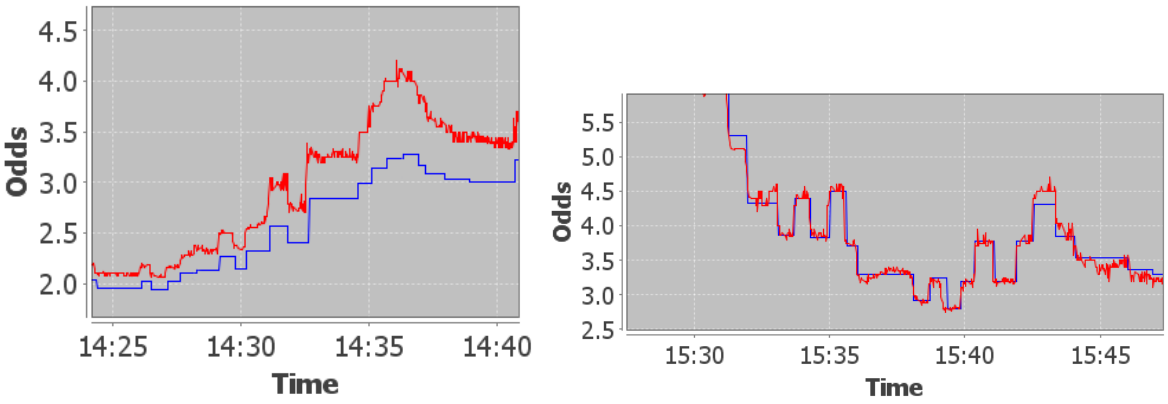


Figure 4: Li v Schiavone - Schiavone's last price Matched against time

From the graph, it is immediately clear that there is a strong similarity between both sets of data, with significant overlap between the two lines. Based on this, it would seem likely that other Betfair users are using a similar model to predict the outcome of a match and are placing bets accordingly. However, there are still areas where the odds differ quite dramatically, even though the basic patterns of movement are similar.

In particular, the first set (roughly before 15:00) has a fairly big discrepancy between the two lines. Figure 5a Looks more closely at a period during the first set, we can see that although the same crenelated pattern can be seen, the Betfair odds are noticeably higher, and appear to be gradually increasing further and further from the model odds. Such a change cannot be accounted for by assuming our initial probabilities of winning on serve were unrepresentative as no such input combination would yield the results seen. In fact, it appears that the probabilities of winning on serve are changing as predicted in the background section.



(a) close up from the first set

(b) close up from the second set

Figure 5: Close ups from Figure 4

From analysing the progress of the actual match, we can suggest reasons for this. Li performed well in the first game, which Schiavone was serving. The game was close, but eventually won by Schiavone after going to deuce. Li then went on to win three points in a row on her serve, before winning the game 40-30. In general, Li had a much better start to the match over the first set. It is therefore no surprise that the odds for Schiavone to win increased beyond that of what would be expected. One could be justified in claiming Li was having a good run of form, and Schiavone a bad one.

Nearer the end of the match, Schiavone made a come back to break Li's serve and forced the match into a tie-break for the second set. Figure 5b shows a close

up of the graph when the current score is roughly five games all. The model and Betfair odds in this case are remarkably similar suggesting that Schiavone's resurgence has altered the current win on serve probabilities to closely match our original prediction. The importance of each point near the end of the game is also increased, and the changes in odds are much more apparent as the match nears it's completion.

From this graph alone, is is easy to give a general outline of the match. Schiavone must have lost the first set and was broken in the second to create the spike around 15:20. She then will have broken back causing the second set to finish closely with a tiebreaker sealing the victory for Li. Overall, there are many areas were a jump in odds is clearly visible, indicating that inferring the score should be possible to at least some extent.

5.2 Schiavone V Bartoli

The next match is the 2011 French Open womens semi-final between Bartoli and Schiavone which saw roughly 14 million pounds traded. This match was another two set to nil victory, this time in Schiavone's favour. Going into the match, it was deemed to be more one-sided with odds of 2.68 for Bartoli and 1.591 for Schiavone. The same graph as before is plotted (figure 6)

Again, the same general pattern can be seen, with both lines appearing similar. We would therefore also expect to be able to determine the score for much of the game. The slight difference being that perhaps the model varies more greatly from the actual Betfair odds. In a similar manner to the Schiavone/Li match, we can explain the changes in odds from a point by point analysis of the match.

Bartoli had good form on service winning her first three service games: 4-0, 4-1 and 4-0. This would easily explain the gradual lowering of the odds below those of the model during the first half of the graph. This gap changed to an extent when she was broken and eventually lost the first set. Nearer the end of the match, the odds are instead significantly higher that the model. We can account for this as Bartoli was a set and break down due to a run of poorer form. As well as being non-favourite to begin with, it is easy to understand why not many would be willing to back Bartoli to make a comeback from this situation.

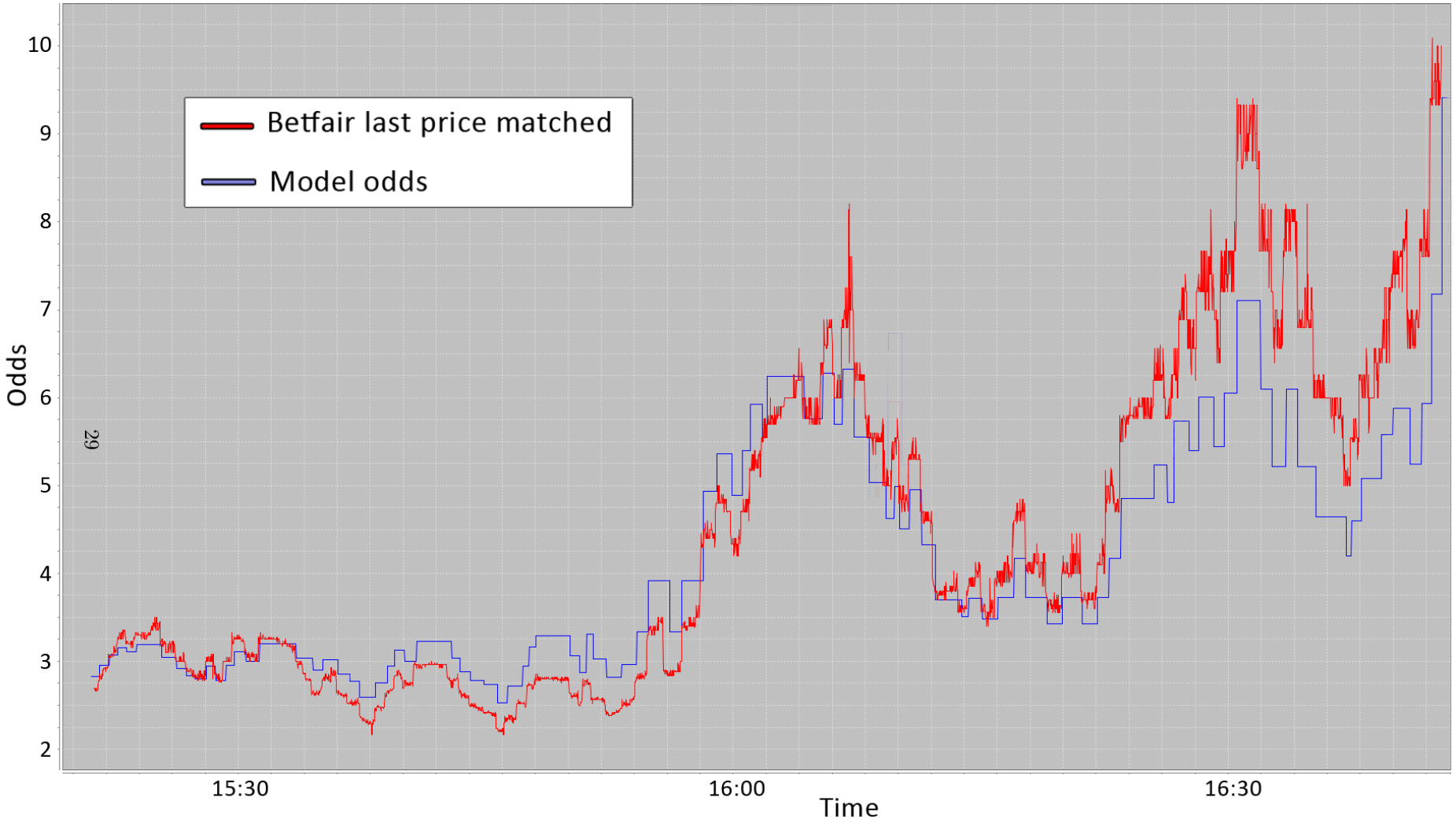


Figure 6: Bartoli v Schiavone - Bartoli's last price Matched against time

5.3 Murray V Nadal

This match is the men's semi-final from the 2011 French Open. Compared to the previous games this match was far more one sided from the start, saw more money traded (25 million) and was also a best of five sets instead of three. Murray, the eventual loser in straight sets, was only given roughly a 0.143 chance of winning against Nadal. Parts of the last priced match graph for Murray are shown in Figures 7 and 8.

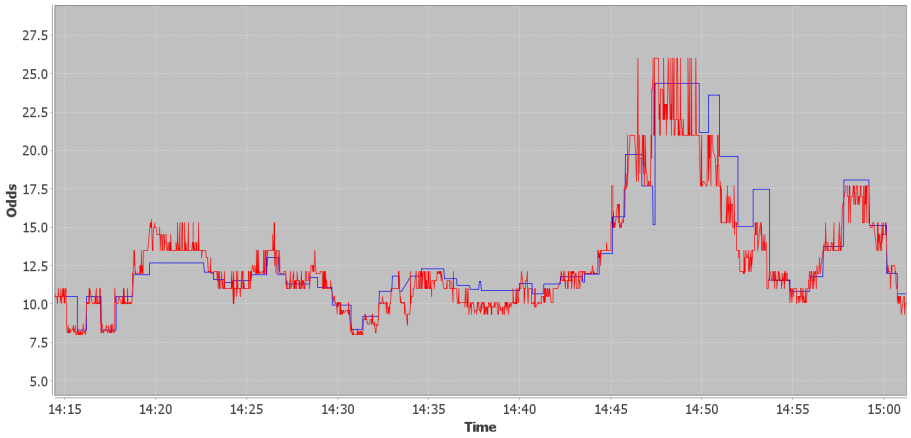


Figure 7: Murray last priced match graph - middle period of match

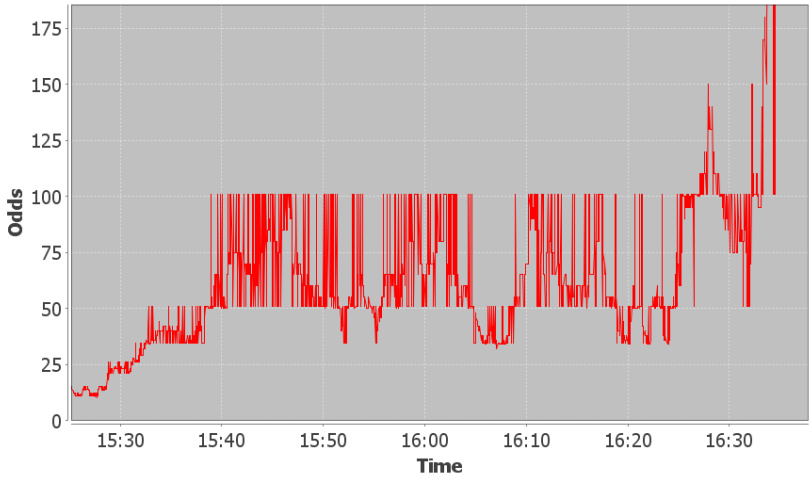


Figure 8: Murray last priced match graph - end period of match

These portions of the graph are far more noisy than the previous matches. Figure 8 especially is almost completely useless for inferring any sort of detail about the match at hand. There are two reasons for this. One reason is simply that the odds are in general much higher than those seen in the previous matches. As the odds increase, so does the boundary between values allowed on the Betfair exchange.

The next reason is due to the cross-matching algorithm described previously. Whilst cross-matching is useful, and theoretically gives us the true last priced matched, it would seem in this case to be a hindrance. Figure 7 shows that you can see a lot of fluctuation between the odds of 101 and 51. These values map to odds of 1.01 and 1.02 matched on Nadal. In this case, we can see that the money matched on Nadal is interfering with the last priced match for Murray.

By plotting the highest back price available for Murray, we see a completely different picture. Figure 9 gives a much more detailed and informative picture on how the odds are changing. This suggests that when the odds increase beyond a certain level, the last priced matched should not be used and instead the odds and amounts available on each player should be considered.

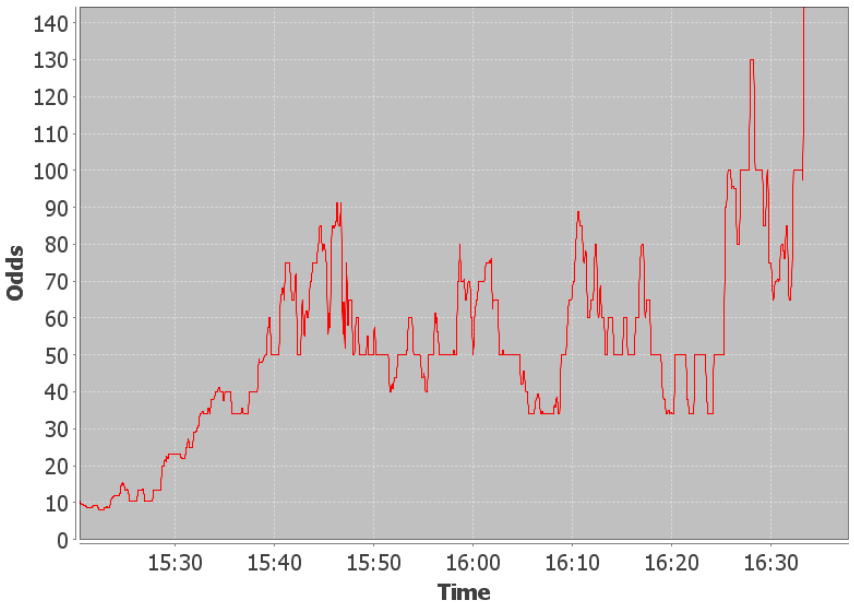


Figure 9: Murray highest back odds available - end period of match

5.4 Determining PWOS probabilities from 'in-play' odds

In the tennis model, the PWOS probabilities determine the players chance of winning from the current state of the match. Therefore, by finding where the odds have settled after a point, we can estimate what the PWOS must currently be to offer the current match odds, given the latest odds and score-line. In this section, the PWOS probabilities are calculated from the Betfair data whilst the recorded score feed is used to determine when points have been won.

After each point, the average of the last priced matched (or highest available price for high odd values) was taken for a short period after the point was resolved. This value was then used for the recalibration. If the point was won by a player, it is assumed that their PWOS will not decrease, but instead might increase. Conversely, losing a point should not raise the players PWOS value as for either case this would contradict the idea of momentum.

Given the current estimate for the PWOS probabilities, and based on the above logic, the current estimates were slightly increased or decreased by a small factor and placed into the tennis model. The combination of PWOS probabilities that predicted the odds closest to their actual value then replaced the current estimations. The graph showing the results from this analysis on the Schiavone/Li match is shown in Figure 10.

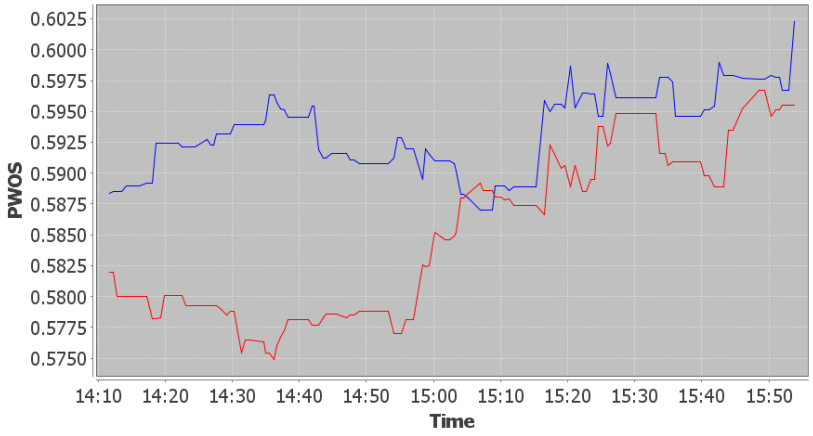


Figure 10: Schiavone V Li - estimated PWOS values throughout the match

At the start of the match, we know that Li was the better player, dominating much of the first set. Therefore, it is no surprise that the initial period of the graph shows the PWOS values separating apart. Relating this to the comparison graph (Figure 4), this would explain the model odds being an underestimate of the actual odds. It is also no surprise that the nearer the end of the match, when Schiavone made her comeback, that the PWOS values are much closer to how they were at the beginning of the match.

However, during the middle of the match, the PWOS values do not vary as expected, with Schiavone's PWOS increasing rapidly. This also coincides with a period during which Schiavone's odds were high. In this case, going on the highest available back price for higher odds would generally gave a slightly lower value than expected. This would account for the increase in Schiavone's PWOS as the lower odds would seemingly improve Schiavone's predicted chance of winning.

The accuracy of this PWOS method would therefore seem to drop when the odds are high, but offers a good approximation at lower odd levels. For higher odds, it is difficult to know exactly where the true odds would lie. This is because as the odds increase, the boundary between accepted odds values becomes larger and the amount of money traded decreases rapidly. If for example, the true odds were 15.4, the nearest odds boundary allowed by Betfair would be 15. When calculating PWOS values this would have a significant effect. Essentially, it becomes almost impossible to estimate the true odds in this case.

6 Inferring the score

From all of the comparison graphs, it is clear that to a large extent the model odds are very similar to the market data retrieved from Betfair. Based on this, it would seem likely that other Betfair users are using a similar model to predict the outcome of a match and are placing bets accordingly. It should therefore be possible to infer the score from the live 'in-play' odds. However, it is also clear that the derived odds vary at times to the Betfair odds due to the idea of momentum and possibly other factors.

In this section, several algorithms for trying to determine the score of a match from the live betting odds are suggested. The reasoning behind each approach is explained and each is tested on a variety of matches. The results are then analysed to see why the method was successful or unsuccessful and based on the results, the algorithm is refined in each step.

7 Non-corrective bounds checking

7.1 Method

The first, most basic algorithm attempts to follow the progress of the match from start to finish by making an assumption as to the current score, and by predicting how the odds will vary once the next point has been resolved. Figure 11 illustrates the general approach, with lines indicating the predicted odds if the point is either won or lost. To see a greater variation in the odds, the probabilities of the current non-favourite are examined.

The task becomes how to determine that there has been a significant increase or decrease in the odds, and how the change relates to the predicted values for either outcome of the point. As seen from the last price matched graphs, there is often a degree of fluctuation in the odds during periods where you would expect the odds to be constant. It is also important to filter out bets that were seemingly placed against the current trend.

To do this, we search through the odds data and examine all values that occurred within the last 20 seconds. Each value is then examined to see whether or not it lies within a close distance to the predicted boundaries. If a certain percentage pass the boundary check, then the point is assumed to be won or lost according to which boundary the odds are nearest. There is also a check to ensure that a point cannot be won until at least 15 seconds have passed since the previous point.

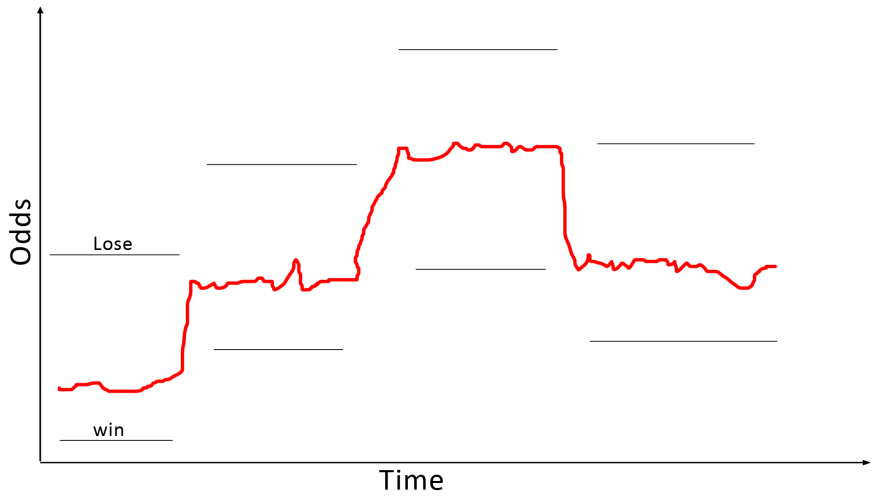


Figure 11: Boundary checking method illustration

7.2 Dealing with non-iid points

From the earlier comparison graphs, it becomes apparent that the derived model does not follow the Betfair odds exactly due to influences such as momentum and possibly other non-obvious factors. Therefore, to adapt the model to the current market after each point is played, the Betfair data is examined and used to create the next set of boundary values.

Once it has been determined that a point has been won or lost, the first step is forming an estimation of the 'new' odds once the market has settled. When enough values meet the boundary check, then the time of the earliest value to enter the boundary is recorded to try and give the exact time of the point. The algorithm then waits for several seconds and takes an average of the values for the period following the change. This attempts to create a more true value when there is variation in the odds.

It is also not possible to assume that the win on serve percentages (WOSP) will remain constant as shown earlier. To combat this and to make the algorithm more robust, code was created to generate a wide range of possible win on serve percentage (WOSP) combinations that would give the same odds for a given scoreline. The list of possible WOSP combinations was then iterated and each entered into the tennis model to get the expected odds if the next point was won or lost.

By taking the minimum and maximum values for each case, essentially, given a current scoreline, there were four values determining the ranges over which we

would expect the odds to lie in-between after the current point had completed. Figure 12 illustrates the updated method. It was then possible to say with a fair degree of certainty that the odds would lie between either of the two boundary areas when the point was resolved.

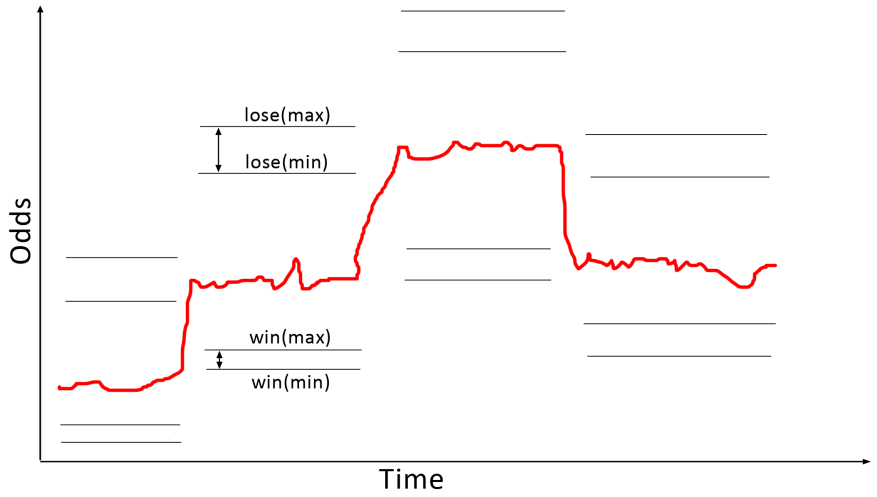


Figure 12: Boundary checking method illustration

However, there is also the possibility that the odds will not vary according to the model, even if all possible PWOS combinations are considered. Whilst this is unlikely for the majority of points, there is nothing to stop the odds not changing as much as they should, or vice versa, for a particular point. A possible reason would be that the predicted odds lie in-between a set of a odds values allowed by Betfair (i.e. 4.4 and 4.3 are allowed, but 4.33 is not). Because of this, there are varying degrees of leniency allowed in the boundary check.

Given the predicted current odds, using the lose(min) and win(max) values, the minimum expected increase and decrease is calculated. Typically, if the odds being checked are at least 0.80 times the expected change then the value is accepted as being inside the boundary. At higher odds, this value is relaxed slightly due to the greater distance between allowed odd values. For this version of the algorithm, if the value lies above the lose(max) or below win(min) then it is assumed to be within the boundary.

7.3 Odds information

So far we have not specified what data is being used when referring to the 'odds'. The comparison section highlighted that depending on the value of the odds, different approaches might be necessary. Theoretically the last price matched

should give the most accurate representation as money being traded is the most powerful indication of the odds being at a particular value. However, we have seen that the last priced match value becomes distorted by cross matching at high odd levels.

The algorithm therefore uses the last price matched value, with cross-matching, as long as the odds remain under a certain value, i.e. 5. Once the odds go over this value, to get a more realistic overview of the market, a moving average total of the last priced matched is used. This is an attempt to make up for the higher boundaries between the odds and potential fluctuation between these boundaries.

If the odds increase above 10, the highest back prices available are examined to determine the odds. As the odds rise, the amount traded over the player generally goes down, so even without cross-matching, the last priced matched over only the one player would still not be sufficient.To avoid distortion in the highest back price available, the amounts available are also taken into account.

In general, the highest odds are ignored if the amount available is small, and significantly less than the amount offered at the next best odds. This is in an attempt to capture the activity of the automated traders, who will more likely be using a similar model.

7.4 Results

The algorithm, as described in this section, was tested on the following matches. The results are displayed on graphs showing the odds used by the algorithm (last price matches, moving average or highest back price) with annotations pointing to the areas where the score is believed to have changed.

7.4.1 Bartoli V Schiavone

The graph (Figure 13) displays the first few games of the Bartoli/Schiavone match. The score prediction is given with Bartoli's score first in each case and appears initially to be accurate. Whenever there is a an upward shift in the odds, this is registered as a point for Schiavone, and the reverse is taken as a point for Bartoli. In actual fact the only correct score-lines are the first five points. The area circled in green causes a problem, and derails the algorithm.

From the graph, one could easily infer that a point has been lost, taking the score to 0-15. This is what the algorithm detects, however, no such point was won and so it would be expected for the odds to remain the same until the point was won by Bartoli. As the current score prediction is now wrong, the min/max values are also invalid meaning the algorithm cannot always continue to good effect.

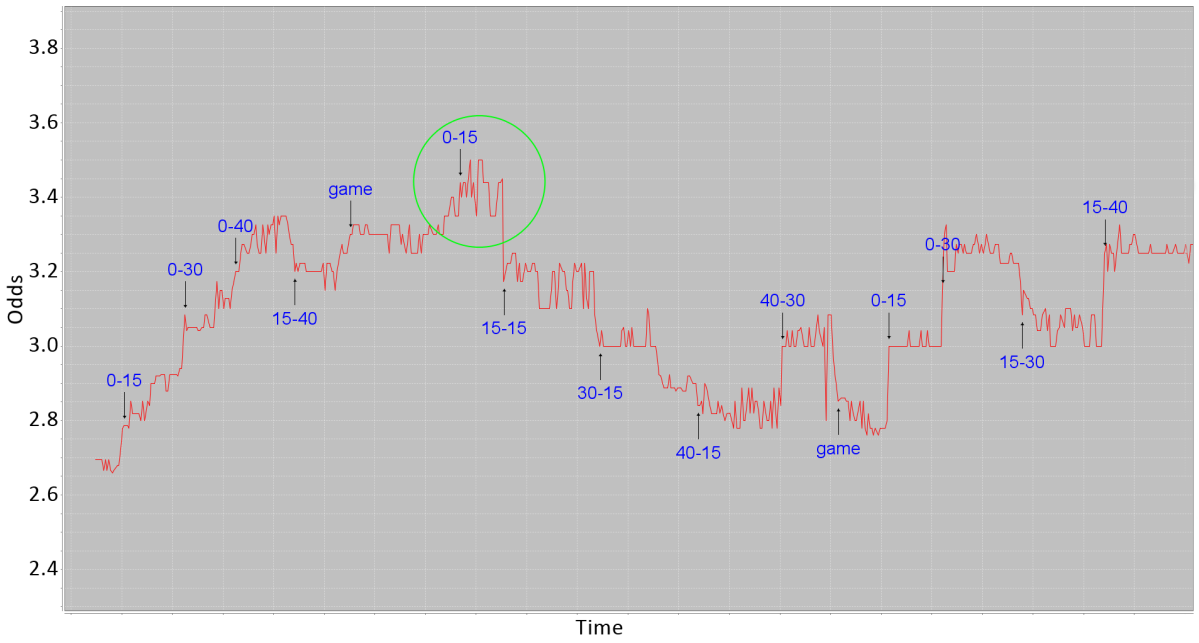


Figure 13: Bartoli last priced matched - score prediction

It is difficult to know what caused this unexpected change in odds. One possible reason can be obtained from the particular line of commentary on the match itself. 'Schiavone comes to the net but sends an easy volley into the net' [5]. Perhaps from watching the game, users expected Schiavone had already won the point as the volley presented itself and decided to take a quick bet before the odds changed. A strategy that obviously would have backfired.

Another possible cause would be acknowledging the fact that the match was still in its infancy. Typically, we can expect that at the beginning of the match the odds will vary quite a bit from the behavior of a model. It is likely that at the beginning of a match you would see a higher amount of users placing bets without any modelling software. It is also possible that those using modeling software are recalibrating based on the current activity or perhaps still waiting to begin trading any money.

For instance, the first game could be used to gauge who is serving first, such information is critical to the model and is not readily available. This could be inferred by looking at how the odds changed during the first game. Alternatively, the amount by which the odds changed could also be used to work out a rough estimate for the PWOS figures in order to begin trading. The fluctuations around the gap after the second game (to the right of where the score is predicted as 40-15) strengthens this theory.

7.4.2 Schiavone V Li

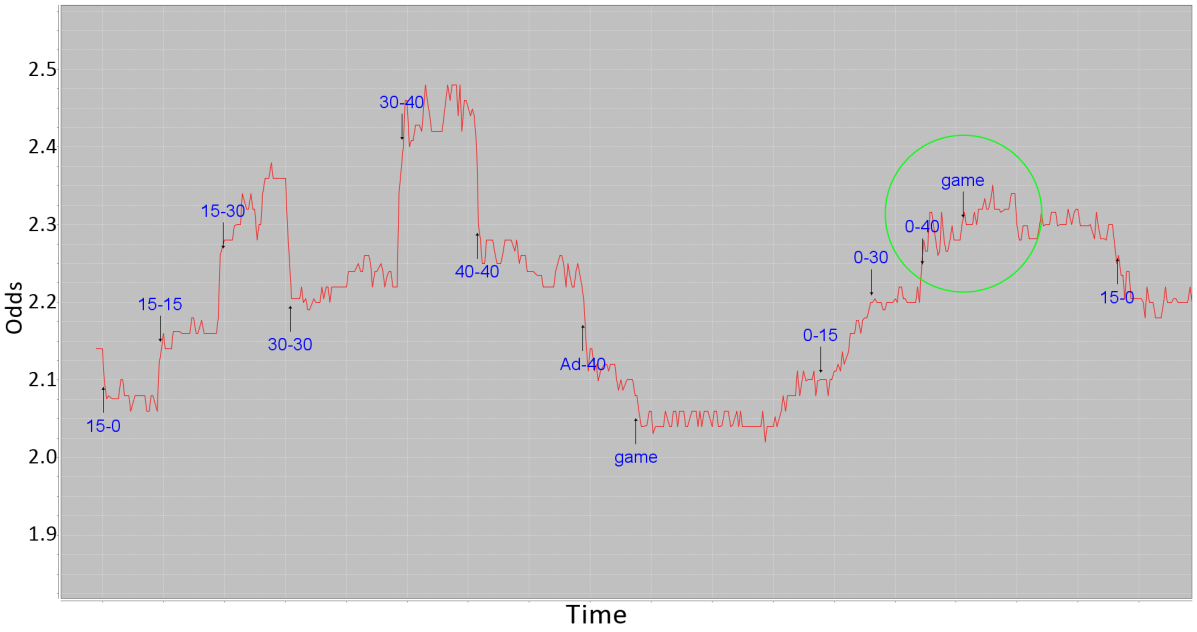


Figure 14: Schiavone last priced matched - score prediction

The graph showing the last priced matched for Schiavone shows the first two games of the match. Similar to before, the main changes in the odds are correctly labeled. In this case, all scores up until the green circle are correct. The algorithm faulters when the end of the game is wrongly predicted due to an apparent rise in odds. (A close up of this is shown in figure 15). After the mis-prediction, the next drop in odds is missed due to the inaccurate boundary values.

The noise in the odds is exacerbated by the low importance of the particular point. As Li already leads 40-0, the game is almost decided anyway and the rise in odds is enough to trigger a point. Essentially, we can infer that since the score became 40-0 to Li, the last price matched varied between values 2.26 and 2.358. It is difficult to justify why this would be the case, as in fact Li double faulted the next point. The graph suggests that after the first fault, Schiavones odds actually rose slightly when we would expect a decrease.

A suggestion would be that as Li was winning the game comfortably, and Schiavone struggling in the first game, the PWOS values were most likely changing quite dramatically at the start of the match. Schiavones odds could then rise as a result of recalibration. This behaviour could be expected if a large amount of money was placed on Li to win/Schiavone to lose, however, analysing the

amounts matched shows this is not the case.

The opening game in this case was fortunate to be a closely contended one, causing a large variation in odds for each point. However, it is still clear that the first game is rather erratic with the odds changing significantly between points. Although at 30-30 the slight rise could be attributed to a Schiavone service fault, this does not explain the large variation after the score goes to 15-30 and 30-40.

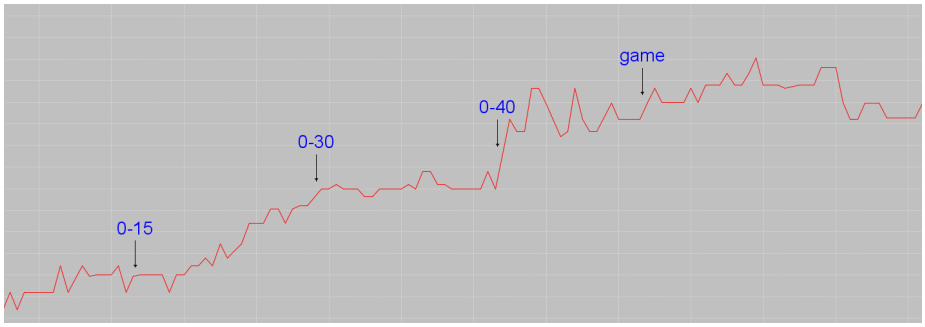


Figure 15: Schiavone V Li score prediction - close up of error

7.4.3 Murray V Nadal

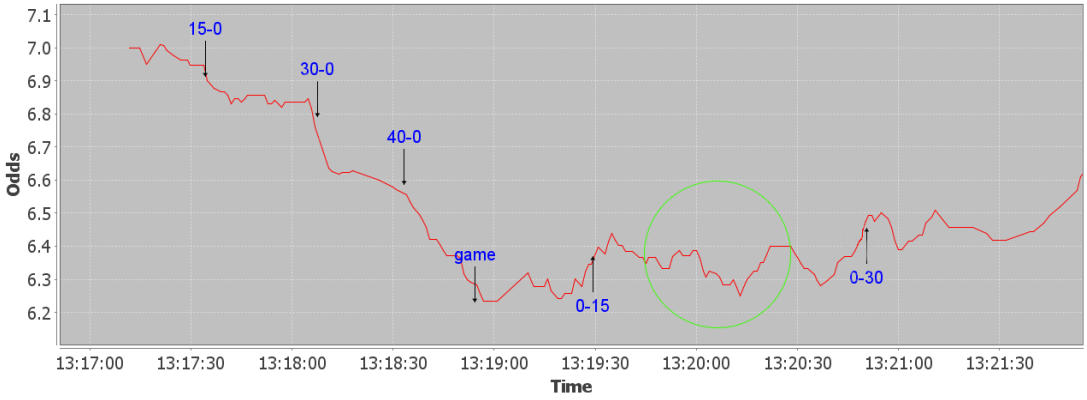


Figure 16: Murray V Nadal beginning of match

Figure 16 is different to the other graphs as the odds are much higher, meaning a moving average of the last priced match is used. The actual end of the first game is indicated by the green circle, with the 0-15 prediction before that referring to a point won by Nadal when the game was 40-0 to Murray. The odds then proceed to drop after Murray has won the game, but surprisingly rise again inside the green circle when the players were changing ends.

Looking at other indicators, such as the highest back price for Murray, shows that when Nadal won a point to take the first game to 40-15 there was no change at all. In fact, the period just before and after the game is won by Murray is rather inconclusive even when looking at all the information available. The last price matched value appears to vary from 6.2 to 6 throughout, whilst the highest back prices indicate that the odds of Murray winning also rise shortly after Murray wins the game.

7.4.4 Total amount matched

To analyse the amount of money traded during the Murray/Nadal match, the differences in the total amount matched on the event were taken between successive calls to the Betfair API every second. The resulting graph (figure 17) shows the amount traded at each particular point during the Murray/Nadal match. Other matches also gave a similar picture

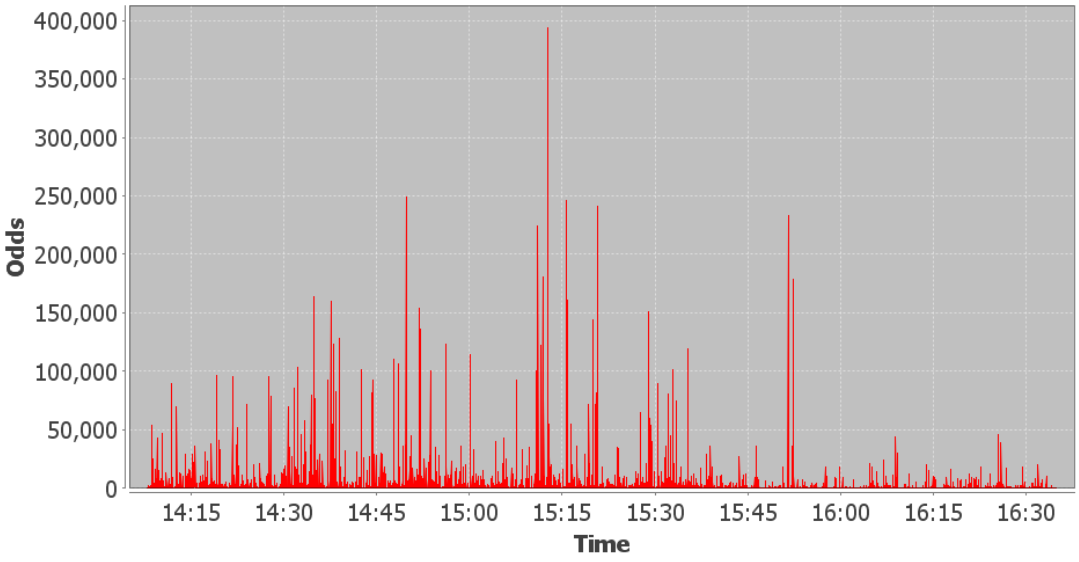


Figure 17: Murray V Nadal amount matched analysis

What is immediately apparent is that during certain areas of the match, an extremely large amount is traded very suddenly. This would seemingly be the result of a single user deciding to make a very large bet. For example, at around 5:12, over £380,000 is placed on Nadal to win. Due to the sheer volume, there was simply not enough money being offered at the highest available odds of 1.11. Therefore some of the money was traded at the odds of 1.11, some at 1.1 and the rest at 1.09.

A similar spike can also be seen nearer the beginning of the match. In fact, a large £200,000 bet was placed on Nadal during the first game, the same at which Nadal won his first point that was missed on the previous graphs. In general, it is easy to see that such a pattern of bets will directly influence the odds and may cause brief periods of unusual activity as the market reacts to such an event. This may offer an explanation for the seemingly anomalous movement in the odds.

7.5 Conclusions

As shown in the score prediction graphs, the algorithm is largely successful at predicting when a point has been won or lost. However, due to the scoring system of tennis, even one missed or erroneous point can have a huge effect on the prediction of the next points. If the end of a particular game is not tracked correctly, then the error carries through and the algorithm derails, becoming ineffective.

It does show, however, that it is possible to avoid the need to add non-idd affects into our model by extracting the information from the bets placed by other Betfair users. Given the odds and score, we can infer roughly what the probabilities of win on serve should be and hence re-adjust the boundary values for the next point. This is advantageous, as there may be factors that cannot be inferred from the pattern of points played alone, such as: a player may have picked up a slight injury concern, is serving poorly, or if there is a gradual increase in the number of people backing a certain player.

Overall, the algorithm highlights the essential ability for error correction for accurate long term score prediction. Several potential sources of anomalies in the odds were suggested. These may arise at any time and must be dealt with. For example, those introduced by the calibration of modelling software, a large amount of money being traded at once, or points where the player is largely expected to win but in fact makes a mistake.

8 Error correction - Breaks in play

In this section, a method of detecting and correcting an erroneous score prediction is detailed. The aim being to utilise scheduled breaks in play to perform score 'check-pointing'. Firstly, a brief background on the rules and regulations of rests within a tennis match is given. The reasoning behind the check-pointing idea is then explained, and the algorithm described in detail. Using this level of error correction, the previous matches are revisited and the affects of the algorithm are examined.

8.1 Rests

Tennis matches can often last for several hours. It is therefore essential the players have a rest at certain intervals, typically when the players change ends. During the match, the players change ends at the end of every first, third and every subsequent odd game. If the set finishes, and the number of games played is odd, the players change ends at the end of the set. Otherwise, they change ends after the first game of the set.

The International Tennis Federation rules state that players may rest during this change in ends in accordance to the following passage.[2]

> When the players change ends at the end of a game, a maximum of ninety (90) seconds are allowed. However, after the first game of each set and during a tie-break game, play shall be continuous and the players shall change ends without a rest. At the end of each set there shall be a rest break of a maximum of one hundred and twenty (120) seconds.

8.2 Explanation

It is inevitable that during a match, there will be many such periods of rest. Given that these break periods are from the moment that a point finishes until the next serve is performed, we can expect that during this period there should be no reason for the odds to vary significantly. Therefore, we can aim to detect these gaps in play and use them to our advantage.

To test for these gaps in play, the time difference between the last registered point and the current time is taken. If this time is large (i.e. 85 seconds or more), then the odds during this period are iterated over. As the boundary values could be wrong, the odds are checked for any significant change that might indicate a point having occurred. Figure 18 shows an selection of the Schiavone/Li odds graph, identifying two break periods.

Another check is performed by the break in play predictor, which keeps track of the time of the last check-point (or start time of the match if no check point has

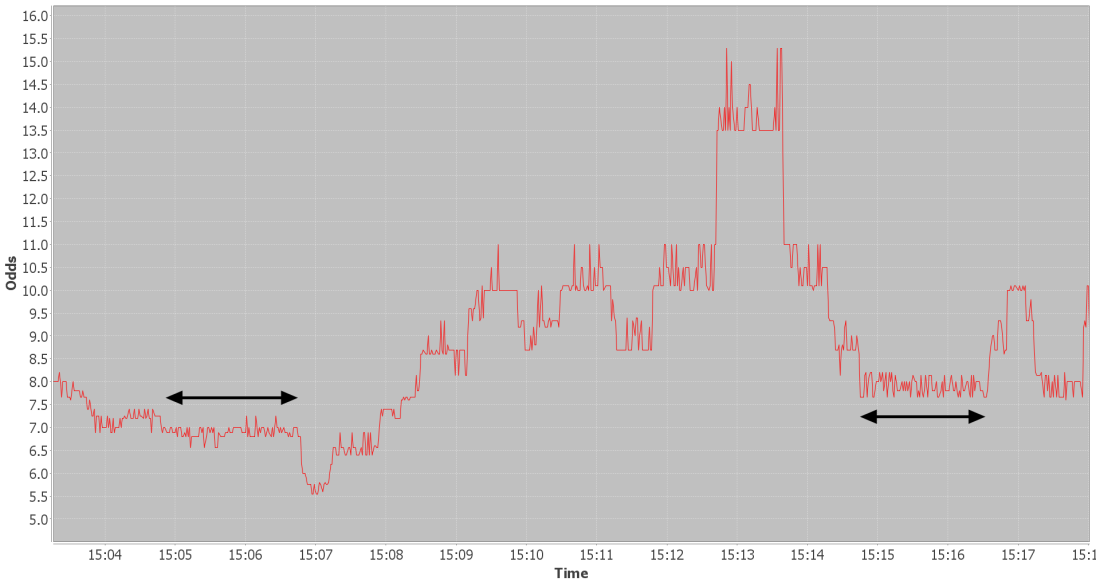


Figure 18: Schiavone V Li - breaks in play

occurred). If the time difference between successive breaks is not long enough to allow a two games to be played, then the break is rejected. If a check-point is detected, and a player is able to win the set by winning the next game, then this condition is reduced to one game. Similarly, if a new set is detected it is increased to three games to accommodate the extra game before a rest.

Given the pattern of rests, it is possible to assume that the total number of games completed in the current set should be odd if a rest is detected. The first break of the set should come at three games, then five, then seven and so on. The only exception is when a set has finished, and the score should be zero games all.

The algorithm essentially works on the basis that each check-point provides an exact prediction of the scoreline. Beginning with the start of the match, at each check point the score and current odd information is recorded. At the next checkpoint, this information is revisited and used to verify the scoreline that is currently predicted, or help correct the score if it is wrong.

If all checkpoints can be detected, then essentially you can 'refresh' the score every few games. In this case, the algorithm should be very effective in keeping track of the game score. This is due to there only being a few possible scenarios that can occur between each check-point. For example, consider a period of the match in which two games are played before another rest. The players can either both hold serve (and the odds will remain similar to the previous check-point), or one player could break the other causing a much greater shift in the odds.

At each checkpoint, the odds and score from the previous check-point are entered into the model. The current odds taken from Betfair are then compared against the model's prediction for the different possible game score combinations at the current check-point. For example, if the previous check-point was at two games to one, then we can determine which of the possible score-lines (4-1, 3-2 or 1-4) matches the current odds the closest.

So far, it has been implied that every check-point can be detected. The problems with this method arise when the majority of check-points cannot be detected. As the number of games between detected check-points increases, the accuracy of the method depletes due to the dynamically changing PWOS. Some score-lines also have very similar odds (eg, 2-3 ,3-4, 4-5) and could easily be confused.

Due to the fact that there is a large variation in the possible lengths of a game (i.e. winning to nil compared to going to deuce several times) it is also difficult to tell how many games should have been played by only looking at the time elapsed. For example, if the last check-point was 40 minutes ago, then there scope for a large variation in the number of games played, creating another difficulty.

To try and help determine the number of games played between checkpoints, the predicted score is used. In general, from the original bounds checking algorithm, when a mistake occurs, errors begin to manifest as new points are missed. This tends to slow the predicted progress of the match. Therefore, one technique is to round up to the next scoreline which would yield a break in play. If the nearest prediction using the model is far off, then the number of games is increased further.

8.3 Results

8.3.1 Schiavone V Li

The Schiavone/Li match was largely successful under this method. A total of seven checkpoints out of a possible ten were correctly identified which can be attributed to the success of the predictions. The end result was that the majority of the match was predicted correctly on a point by point basis, and the final score of each set was correct.

Out of the seven check-points, only one had to alter the score. The error was created due to the current score of the match causing the odds to change rather insignificantly. Since the previous checkpoint, the first game was served and won by Li 40-0 and the second game was won 40-15, after reaching 40-0, by Schiavone. This particular scenario is difficult to track from the odds. Once the score reaches 40-0, even if either player wins the point the odds will change only a very small amount.

The scoring prediction when the error correcting check-point was reached was three games to four, two points all. This was rightly changed to four games to five by the error correction algorithm. In fact, a few other points were missed from similar scenarios but in many cases the errors managed to hide and did not effect the future score prediction after the current game had been resolved.

The entire first set is shown in several stages in figures 19 & 20. Check-points are marked as CP, followed by the predicted game score.

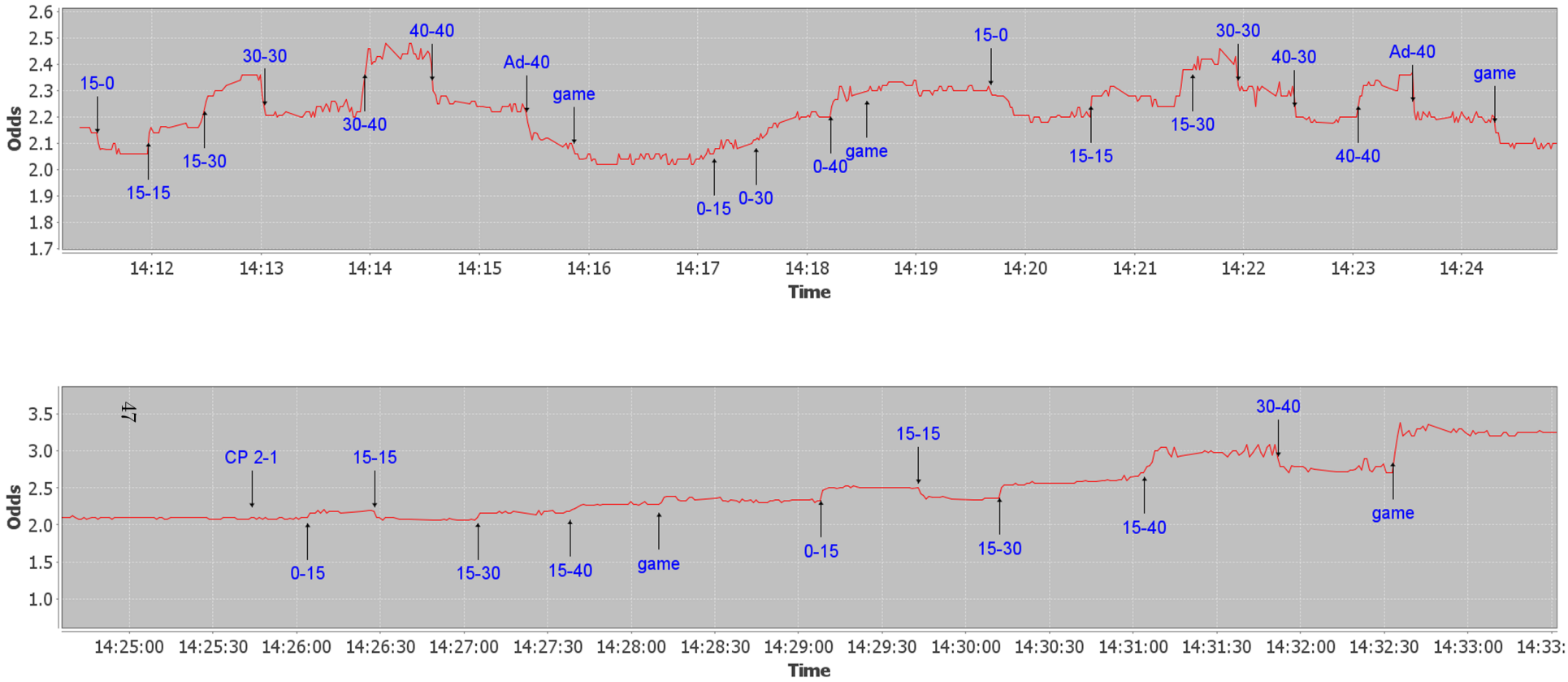


Figure 19: Schiavone V Li - first set score prediction with check-pointing. The odds plotted correspond to those used by the algorithm (i.e. moving averages when the odds are over 5 etc.) PART ONE

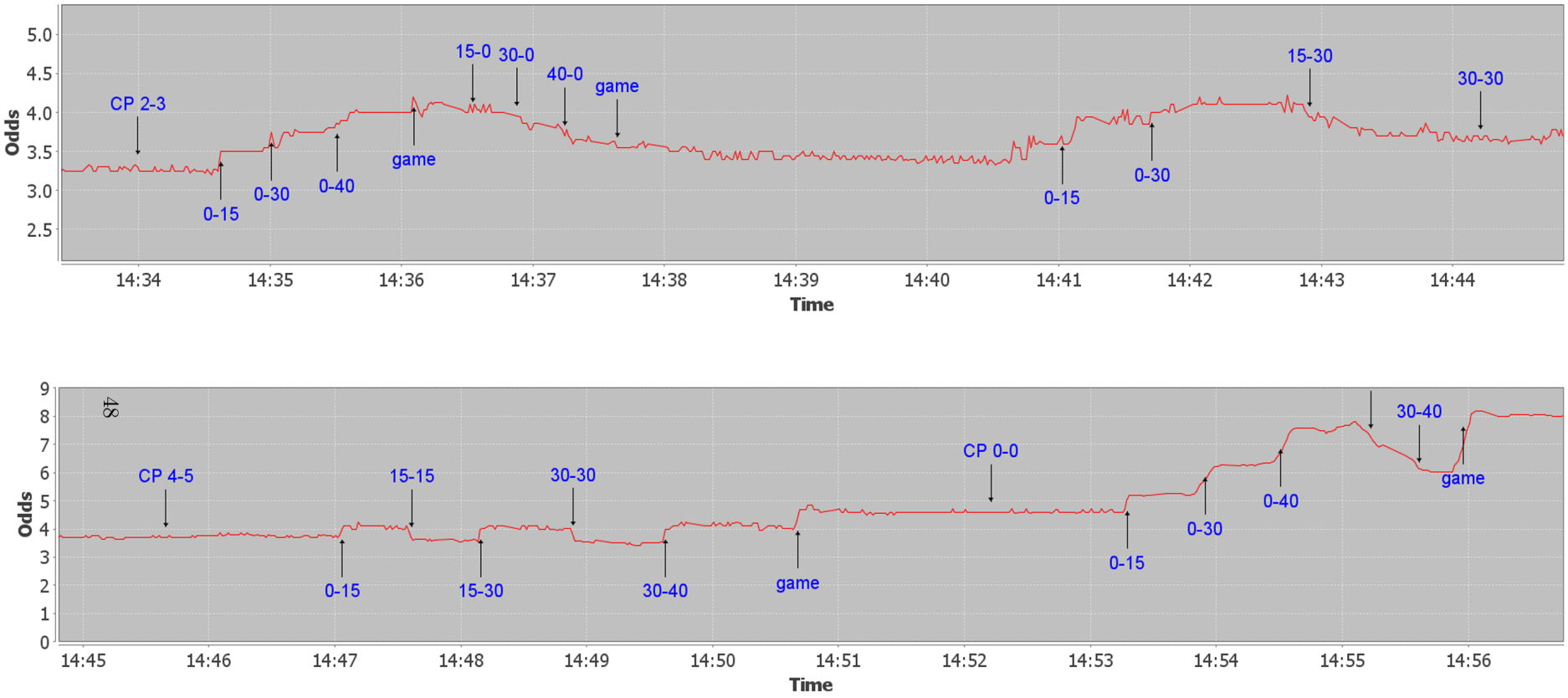


Figure 20: Schiavone V Li - first set score prediction with check-pointing. The odds plotted correspond to those used by the algorithm (i.e. moving averages when the odds are over 5 etc.) PART TWO

8.3.2 Bartoli V Schiavone

The Bartoli/Schiavone match also saw a great deal of success with the help of error correction. Although only three breaks in play were detected, two of these checkpoints made minor corrections to enable to algorithm to continue effectively. Similarly to the Schiavone/Li match, the final set scores were correct, and much of the match saw flawless prediction.

There were two main errors in the predictions during the match, each having a significant carry on effect. The first occurs just after the third game following a break in play. Figure 21 shows how the real odds (red) varied against what we would expect expect using the model odds (blue). In this case, we see there are seemingly two discrete 'step's taken by the odds (as indicated by the green circle), when there should really only have been one.

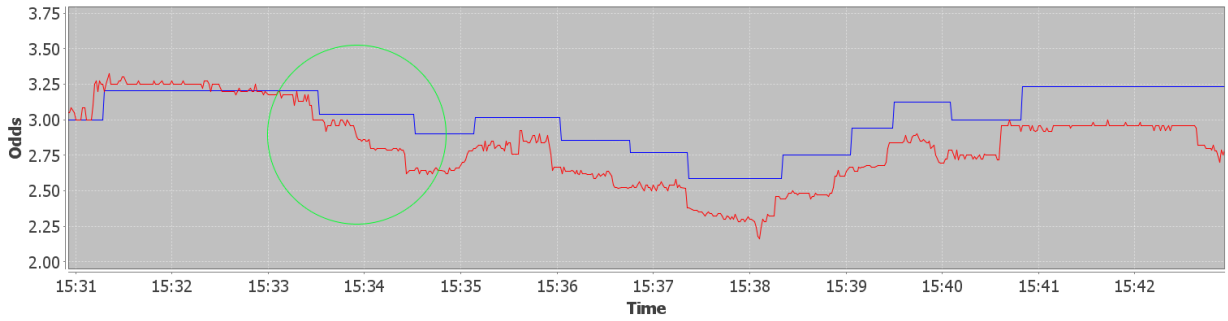


Figure 21: Bartoli V Schiavone - A graph of Bartoli's odds and a set of predicted odds, highlighting the source of an error after the third game

The algorithm therefore detects that two points have been won by Bartoli, and the rest of the score prediction up until the next checkpoint is invalid because of this error. It is also worth noting that the checkpoint just prior was not detected either. This is because the odds vary quite a lot from the beginning of the break period, up until the end. In fact, a point was awarded to Bartoli just prior to entering the green circle.

Interestingly, whilst this might be symptomatic of a sudden large bet, this is not the case at all. In fact, comparatively very little was traded during this period. Referring back to the comparison graph (Figure 6) we can suggest another reason for this activity in the odds. From the end of this game onwards, the model odds become noticeably higher than the real odds, even though they were largely similar for the first part of the match. This can also be seen by studying the beginning of figure 21.

This behaviour can be explained by accounting for Betfair users with modelling software. As already discussed in the comparisons section, Bartoli had a

much better start to the match. It appears as if this break in play was used for recalibration purposes. Whilst we might expect this to be a more gradual processes, there appears to have been a rather rapid transition in the PWOS values.

The next error is of a similar vein, shown in figure 22. It also occurs during a break, this time between the first and second sets. This time, as Bartoli has lost the set, the odds slowly increase as expected, perhaps due to model recalibration as before. There is also a large bet of over £250000 placed at roughly 16:02:10 on Schiavone to win, this may also account for the change in odds as the market adjusts.

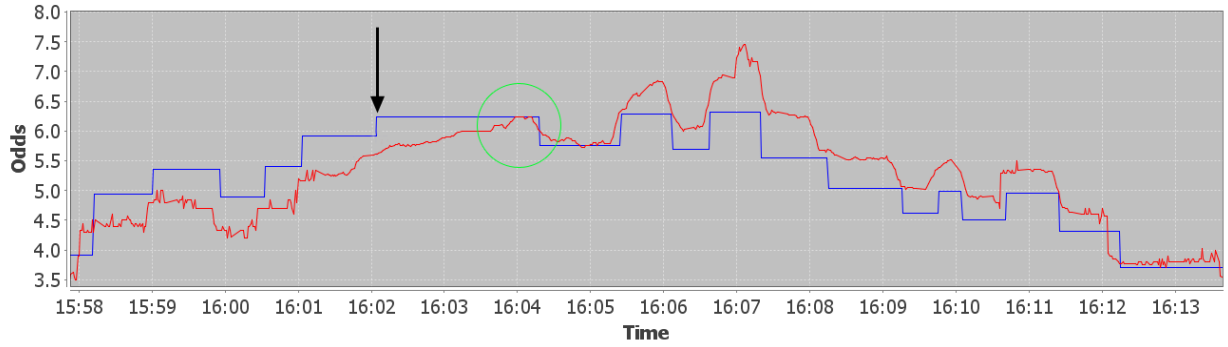


Figure 22: Bartoli V Schiavone - A graph of Bartoli's odds and a set of predicted odds, highlighting the source of an error inbetween sets. The arrow indicates the end of the first set

This rise at the end of the break is sufficient enough to award an erroneous point to Schiavone and again throws the algorithm off. Fortunately, the next possible break in play after this anomaly is detected, three games into the set. Once the error was successfully corrected to one game Schiavone, two games Li, the score is actually predicted with 100% accuracy for the remainder of the match.

Figure 23 shows the score prediction graph from the point of the error correcting checkpoint in the second set to the end of the match.

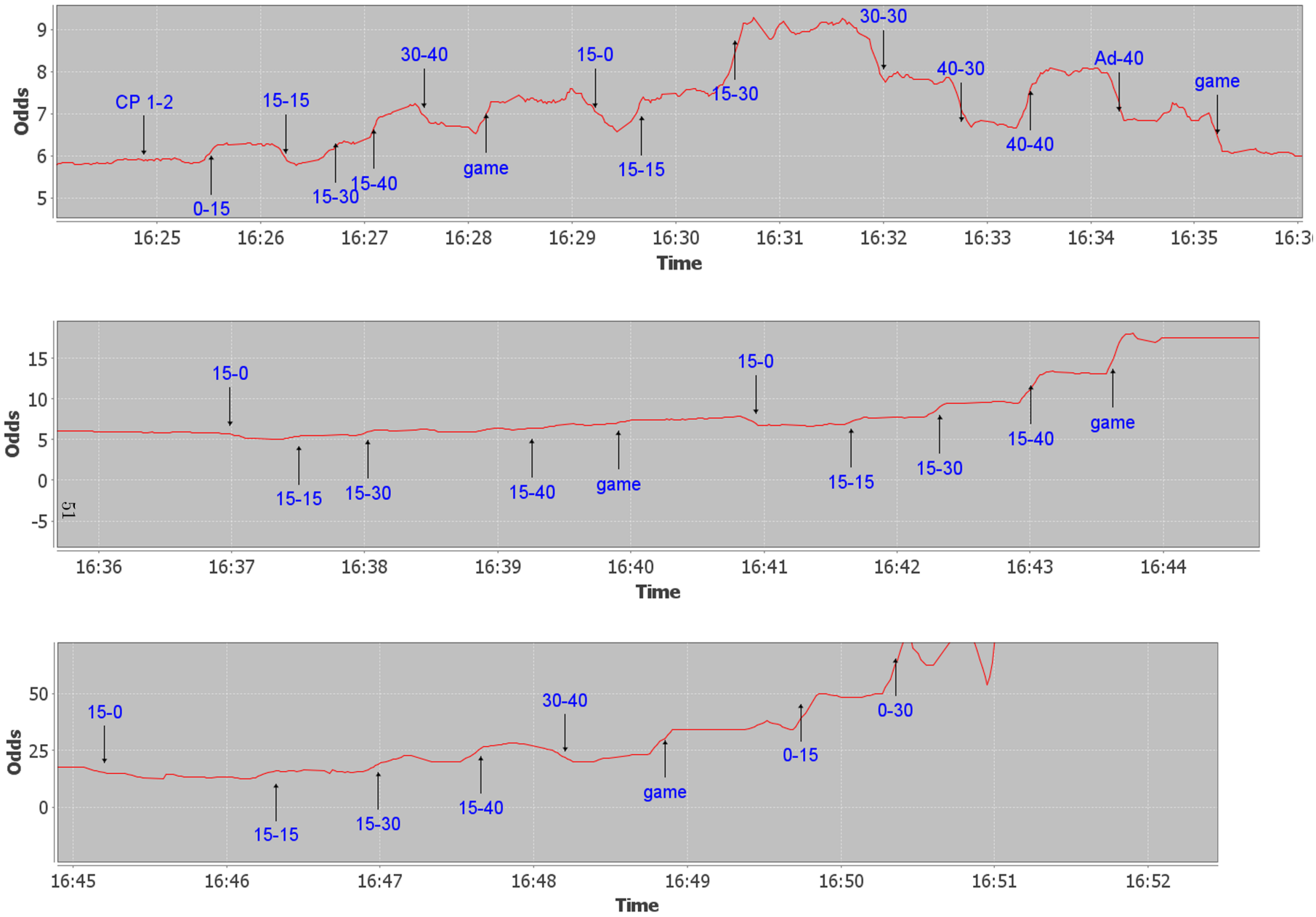


Figure 23: Bartoli V Schiavone - Graph of Bartoli's odds with score prediction during the latter part of the second set (the very last point is cut-off)

8.3.3 Murray V Nadal

The Murray/Nadal match predictions improved under the check-pointing method, but not as well as the other matches. There were periods where the score prediction was correct, but these were not of a great length. The number of errors made between check-points was generally increased due to the several reasons. Firstly, the match was the best of five sets, meaning the points were generally of a lower importance throughout, making it more common for points to be missed nearer the start of the match when Murray's odds were at there lowest.

Secondly, Murray's odds were also became very high, and as his chance of winning decreased, the quality of the odd data diminished. This is because the majority of money was being traded over Nadal, and we therefore had to use the highest available back prices for Murray. These did not give as good a picture as the last priced match values used for the previous two matches. There were often large jumps seen in the data, making errors more common.

8.4 Conclusions

Using breaks as a way of correcting the score has shown to be useful. On many occasions, the importance of the point can become too low or an anomaly in the odds can occur at any moment. Even a single mis-prediction can have a knock on effect for the entire match using the current algorithm and the overall result is poor.

Given a chance to correct an error, the original algorithm performed well, being able to correctly predict the score of a match for a lengthy periods at a time. In the Bartoli/Schiavone and Schiavone/Li matches, the score predictions were very accurate for the majority of the match.

However, the obvious downside of this method is that it does rely on being able to infer the periods of rest within a match. If the number of breaks in play detected is low, it is possible for large errors to occur. There is also a small chance that a break could be predicted when no such break has occurred. For example, if there is a very long point, an injury break or other such events. Although unlikely, any of the above will heavily derail the score prediction algorithm.

8.4.1 Set Markets

Another way of determining the moment when a set is finished can also be inferred from the 'set betting' market. Betfair often allows users to bet on the winner of a particular set, although in most cases, these set markets are poorly traded compared to the match odds. From this, we also assume that the data will not be very useful in determining the current score.

Whilst the set betting market would offer a good and conclusive idea as to the end of a particular set, it was not used in this project. The reason being that to obtain the Set market data required further API calls which would have eaten into the amount of requests allowed on the free API service, reducing the accuracy of the match odd data.

9 Evaluation

Based on the background research, a model was created to predict the chance of a player winning from any possible score-line reachable during a match. The model used several assumptions, as was common in relevant literature, such as that each point would be independently and identically distributed. Once complete, a simulator was created to ensure the values were correct. The end result was a fully functioning tennis prediction model.

The analysis of the Betfair odds and those generated by the model showed a good level of similarity. This implied there was scope to relate the changes seen in the Betfair odds to the derived model and this was a crucial factor in being able to determine the score of a match in progress. This allowed the project to continue, and inferring the score was attempted.

Generally speaking, the methods used to infer the score were successful. In most cases, a point could easily be detected. However, to obtain the correct score-line, error correction was necessary. From analysing the Betfair data, we found that there are clear cases whereby the odds would indicate that a certain event has occurred, when in reality it had not. When these occured, it often resulted in the score prediction becoming invalid.

Using the fact that players take scheduled breaks, a form of error detection and correction was implemented. This helped the score predictions get back on track once an error had occurred either due to an anomaly in the odds, or if a low importance point was missed. For the Schiavone/Li and Bartoli/Schiavone matches in particular, the error correction algorithm meant the score predictions were largely correct throughout. The correct set scores were determined in both cases, and there were long periods of flawless prediction.

A problem which occurred throughout the project was trying to gain an accurate representation of the odds for a player when the predicted outcome of the match was heavily one-sided. When a match reaches such a state, the vast majority of money is wagered on the favourite. This means that cross-matching can no longer be used to infer the odds as the odds become distorted as seen in the Murray/Nadal match. The best estimation therefore comes from the best available prices, as usually not enough money is traded over the non-favourite to give a reasonable last priced matched value.

For example during the Murray/Nadal match, when Murray's odds were in the 20 range, the best available back prices were £6400 offered at odds of 20, £4 offered at 18.5 and £27 at 18. The best lay prices were £43 at odds of 22, £2 at 23 and £148 at 25. Compared to the hundreds of thousands offered to back and lay Nadal at the same time, Murray's odds would appear to be a much weaker estimation of the 'actual' odds.

In fact, there is such little money being offered that there is are gaps between odd boundaries where no money is being traded. For example, no money is offered at the odds of 21. The best available prices we see are therefore likely to be largely different from the actual odds, which may lie anywhere in-between the highest available back price and lowest available lay price. If the actual odds were to change from 21 to 20.5, there would seemingly be no reason for the currently available odds to change.

This made it difficult to try and infer the PWOS from the market data, as it relied on an accurate representation of the 'actual' odds, and it would seem difficult to estimate these values from the data available. The score prediction still worked to a large extent when the odd values were higher, however, the number of errors generally increased due to periods of erratic behaviour in the highest available back price.

10 Conclusion

From the analysis performed on the Betfair data, the manner in which the odds data changed heavily suggests that the market is controlled by users using tennis modelling software. From looking at the amounts traded, it is hard not to assume these users are responsible for a large portion of the traded volume and thus they have a profound influence on the odds.

It became clear that that the Betfair market data collectively resembles a model which incorporates factors such as the momentum that a player can achieve by winning several points. This resulted in the need to account for dynamically changing PWOS values. We also saw how we can infer the changes in PWOS from the live Betfair odds and the actual score-line.

Given the current predicted state of the match, by considering a wide range of possible PWOS values that would give the current scoreline and Betfair odds combination, we were able to follow the progress of the match without having to estimate the actual PWOS values or incorporate non-iid effects such as momentum into our model.

Using the bounds checking method, we show that it is possible to infer when a point has been won or lost on the majority of occasions. This is done by looking for situations where the odds change significantly in relation to the boundary

values calculated using the tennis model. In general, the higher the importance of the point, the easier is it to detect. With some low importance points seemingly going undetectable.

From studying the Betfair data further, we see that anomalies often occur whereby the odds change unexpectedly. These are typically incorrectly regarded as points having occurred and the model proceeds to recalibrate itself incorrectly. As the scoring system of tennis is a hierarchical one, a single point can have a large effect on the score prediction. Therefore, we show that in order to give an accurate long term estimation of the scoreline, any prediction algorithm requires a form of error correction due the regularity of these anomalies.

Overall, we see that inferring the score from Betfair is in fact possible and can be achieved with a good accuracy providing the match is sufficiently traded and competitive. This proves that the market is highly adaptive to each point, and providing the importance of the point is substantial enough to warrant a change in the odds, almost all points can be detected. This indicates that the Betfair odds actually give a very accurate prediction as to the outcome of the match. Based on what is seen, there is also then scope for a gambling tool to be created, as outlined in the next section on future work.

11 Future Work

11.1 Gambling Tool

Being able to infer the score proves that generally the market is stable and the odds change in a rational manner. It also means that for each point, an accurate estimation of the odds after the next point has been resolved is possible. Given the current scoreline, we can say with a great degree of certainty where the odds can expect to lie if the next point is won or lost.

Therefore, given a live score-feed of a match, or watching the match on television offers an opportunity to utilise the model to place bets to give the potential to win money with almost zero risk. For example, if watching the match, a button could be clicked to indicate that a player has won or lost a point. Appropriate bets could then be placed automatically, taking into account the expected increase/decrease in odds.

For example, if the current odds are 4.5, and now that the point has finished we expect the odd to fall to 4.1, the software could automatically place a bet on the player with odds of 4.2. Although the re will be a five second delay until the bet can be matched, there is a chance that money may still be available at odds of 4.5 from users who have not yet reacted to the latest point.

If we are lucky, money may be matched at odds of 4.2 or above. This can then

be immediately layed off at the new lower odds, resulting in ensured profit. On the other hand, if the bet is not taken, we can cancel it immediately, regaining the money that was staked. In fact, the only way to lose money would be due to a large delay in the television stream, a software error, or a sudden unexpected change in odds. However, it would seem that the chance of this occurring could be outweighed by the positives.

To obtain even better prediction for how the odds should change, there is also scope for trying to determine the PWOS values from the live data. As shown in section 5.4, providing the odds don't reach high values, the estimations could be used to give more exact values of where the points should lie after a particular point. This may then allow you to place a less conservative bet. For example, if you can predict the odds will drop to at least 4.3, instead of 4.5, then you could place a bet with odds of 4.4 and still likely make a profit.

11.2 Further Error Correction

Errors occurring soon after the detection of a checkpoint can cause large sections of the score prediction to be wrong until the next checkpoint is found. In this section, a different form of error correction is described that aims to detect and correct errors soon after they occur, rather than during breaks in play. Unfortunately due to time restrictions, this could not be implemented, but the approach that would be undertaken is described here.

A single point missed, or erroneously detected can have a huge impact on the odds you would expect to see on the next point. For example, if you consider a new game where the score is 0-0 and the score prediction is incorrectly still 40-0 on the previous game. If player one was to win a point, the odds could jump rather dramatically, as player one would be 15-0 up against the serve of player two. However, if the score prediction algorithm detects this point it will assume it will be game point for player one.

If the current algorithm derails, it is possible that, at least until the next checkpoint, the change in odds seen for a point may be fairly different from the expected odds, especially if the following points are of high importance. Therefore you can aim to detect periods in the score prediction where the odds change far more than expected (i.e. using the max values in the boundary algorithm), or when there is a significant change in the odds that does not meet the minimum change requirements for the particular point. Either would be a heavy indicator that the current score prediction is wrong.

Once it has been detected that the score is wrong, the hard part becomes how to correct the error. In fact, any number of previous points predicted could be wrong. This process might involve going back a certain number of points before the error was detected or until the previous check-point. It would then be the case to find a possible scoreline that would give the change in odds actually seen

at a particular point.

This might be done by creating a binary search tree, iterating back over the recent data. Whenever a point was detected, then you could create two nodes representing the score-lines where the point was given, and the other if you were to ignore the point. If a point isn't detected for a long time (i.e. a point could have been missed) during a period when the match should be in-play, then a branch could also be made to explore the possibility that a point had in fact occurred. You could then look for the change in odds expected between score-lines and compare it to the change seen to find the nearest match.

References

[1] *Betfair* Betting Exchange website URL, accessed 2011
`http://www.betfair.com/`

[2] *International Tennis Federation*, Rules of Tennis, accessed 2011
`http://www.itftennis.com/abouttheitf/rulesregs/rules.asp`

[3] *Betfair Developer Program* API Reference Guide, accessed 2011
`http://bdp.betfair.com/`

[4] ATP World Tour Tennis rankings, accessed 2011
`http://www.atpworldtour.com/Rankings/Singles.aspx`

[5] Schiavone V Bartoli Match commentary and point by point scorecard, accessed 2011
`http://www.tennisearth.com/commentary/full_Roland-Garros_Women-Singles_SF_Bartoli_Schiavone_319414.htm`

[6] Newton, P. K. and J. B. Keller. (2005) Probability of Winning at Tennis I. Theory and Data. *Studies in Applied Mathematics*, 114:241-269.

[7] Klaasen, F. J. G. M and Magnus, J. R. (2001) Forecasting the winner of a tennis match.

[8] O'Malley, A. J. (2008) Probability Formulas and Statistical Analysis in Tennis. *Journal of Quantitative Analysis in Sports Volume 4, issue 2*

[9] Barnett T., Brown A., Clarke S. Developing a Model that reflects the outcomes of tennis matches.

[10] Klaasen, F. J. G. M and Magnus, J. R. On the independence and identical distribution of points in tennis

[11] Klaasen, F. J. G. M and Magnus, J. R. (1996) Testing some common tennis hypotheses: Four years at Wimbledon

[12] Klaasen, F. J. G. M and Magnus, J. R. (2001) Are Points in Tennis Independent and Identically Distributed? Evidence from a Dynamic Binary Panel Data Model *Journal of the American Statistical Association, Vol 96, No. 454, pp.500-509*