Inferring the score of a tennis match from in-play betting exchange markets

A. Gabriela Irina Dumitrescu*, B. Xinzhuo Huang**, C. William Knottenbelt***, D. Demetris Spanias****, E. James Wozniak*****

Department of Computing, Imperial College London, South Kensington Campus, London, SW7 2AZ

* *irinadumi@gmail.com*

** *xh208@ic.ac.uk*

*** *wjk@doc.ic.ac.uk*

**** *d.spanias10@imperial.ac.uk*

***** *jhwozniak@gmail.com*

Abstract. Over the past few years, betting exchanges have attracted a large variety of customers ranging from casual human participants to sophisticated automated trading agents. Professional tennis matches give rise to a number of betting exchange markets, which vary in liquidity. A problem faced by participants in tennis-related betting exchange markets is the lack of a reliable, low-latency and low-cost point-to-point score feed. Using existing quantitative tennis models, this paper demonstrates that it is possible to infer the score of a tennis match solely from live Match Odds betting exchange market data, assuming it has sufficient liquidity. By comparing the implied odds generated by our quantitative model during play with market data retrieved from the betting exchange, we devise an algorithm that detects when a point is scored by either player. This algorithm is further refined by identifying scenarios where false positives or misses occur and heuristically correcting for them. Testing this algorithm using live matches, we demonstrate that this idea is not only feasible but in fact is also capable of deducing the score of entire sets with few errors. While errors are still common and can lead to a derailment of the detection algorithm, with more work as well as improved data collection, the system has the potential of becoming a precise tool for score inference.

1. Introduction

Tennis is one of the more popular sports in the world, enjoyed by millions of spectators. The popularity of the sport makes it attractive to betting exchanges which offer a range of betting markets related for nearly all professional tennis matches. One of the largest betting exchanges is Betfair which offers its customers the opportunity to take up (back) or offer (lay) bets in a peer-to-peer manner. These bets can take place before the start of the match or during play. Being a large betting exchange, Betfair attracts a large volume of bets from customers which in turn results in very liquid markets – especially for the more popular professional tennis matches. High liquidity in an exchange market translates to exchange odds which change at a high speed and which react very efficiently to real life on-court events (such as a point being scored in a match).

This high liquidity in the exchange market is provided by bets placed by both humans and automated trading systems alike. Like in finance, the majority of liquidity is provided by automated systems which trade algorithmically. Any algorithmic trading software has some underlying quantitative model which drives the decisions made by the software. Speculating on how these models work is out of the scope of this paper, but certainly there must be an input for the current state of the match (like the score).

A good in-play automated trading system must monitor and adapt to real life on-court events. It should immediately change the target odds it trades at according to the current state of the game. As most systems do that in an efficient way, the market odds offered on the exchange appear to adapt very quickly to represent the state of the match.

A problem that new systems may face is the automated retrieval of data which can be used to identify the current state of the match. Manual entry of the current score of the match has been a common method used by current professional systems. The problem with this is that it is not cost effective and it is prone to data entry errors. A more reliable source of the current score of a tennis match is a score feed connected directly to the umpires' chair of every match but this is expensive to acquire. There are some other free sources available online, like the ATP World Tour official live scoreboard but most of these are susceptible to large delays and do not provide the data in an easily retrievable way. This paper aims to introduce a system which takes advantage of the variable odds of bets matched on a betting exchange to infer the current state of the match. The target result is to be able to provide a reliable score feed of any tennis match with sufficient liquidity in its betting exchange market.

2. Hierarchical Markov Model

The hierarchical scoring structure of tennis allows for the modelling of games, sets and matches through Markov Chains which are linked together recursively. The method, described in Liu (2001) is further adapted to account for the difference in the probability of a player winning a point when (s)he plays the role of the server or the receiver. This section details how Hierarchical Markov Chains can be employed to determine a player's odds in a best-of-three sets match – a best-of-five match can be modelled analogously.

As one of the key characteristics of Markov Chains is memorylessness – the next state that the system transitions to depends only on the current state; previous states have no impact on the transition. Consequently, this technique assumes that points in tennis are independent and identically distributed (i.i.d). Klaassen & Magnus (2001) show that while the i.i.d. hypothesis is not strictly a realistic assumption, the divergence from this in reality is small enough that making the i.i.d. assumption can produce good results.

The most straightforward way to illustrate this technique is by starting at the highest level – that of the match – and work our way down to the lowest level; a single point. The match level is the simplest one, as it is comprised of a finite set of states, or set scores. A best of 3 match, starts with the 0-0 score, and ends when a player is first to win 2 sets. Let p be the probability that player A wins a set, and *(1-p)* the probability of A losing a set. Using recursion, one can calculate the probability of A winning the match from any level in the chain, knowing that in the final two states, the probability of winning the match is either 0 (player A already lost) or 1 (player A already won). For example, given a 1-1 set score, to compute A's probability of winning the match, we calculate p multiplied by probability of winning the match from the Win final state (i.e. 1) plus *(1-p)* multiplied by the probability of winning the match from the Lose final state (i.e. 0).

$$P(A\ wins\ |1-1) = p \times 1 + (1-p) \times 0 = p$$

We then can use the result obtained above to compute the probability of A winning given a 1-0 score:

$$P(A\ wins\ |1-0) = p\ \times 1 + (1-p) \times P(A\ wins\ |1-1) = 2p - p^2$$

The Markov Chain for the set level is similarly constructed, with a few variations. There is a differentiation between the probability, p_A, of player A winning a game when A is the server and the probability, p_B, of Player B winning a game when B is the server.

Assuming the set starts with A serving, then p_A shall be used to compute the probability of A winning at every even game, and $1\text{-}p_B$ is the probability of Player A winning at every odd game. For a tie-break set, the Markov Chain has one special case, as the probability of moving from 6-6 to either final state is computed using p', the probability that A wins the tie-break.

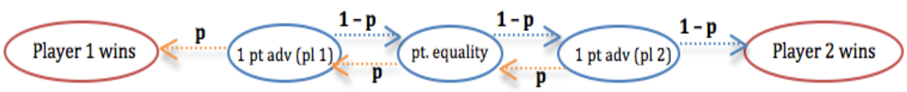

Figure 1 – Random walk with two absorbing states.

The advantage set model, the tie-break model and the game model all share one common characteristic – if there is a tie before winning, one of the two players needs to win two consecutive games/points to win the set/tiebreak/game. These two steps to victory are defined by a random walk with two absorbing states as shown in Figure 1.

From the point equality state in a game, the only way to reach the winning state is by winning two consecutive points *(*p^2*)*, and analogously for losing the game $(1-p)^2$. However, there is an indefinite number of transitions that can occur between the deuce point and either of the advantage points.

To return from deuce point back to itself, one of two things can occur: either player A loses a point and then wins the next one or player A wins the point and loses the next. The corresponding probabilities are:

$$p \times (1-p) + (1-p) \times p = 2p \times (1-p)$$

Since this can happen an infinite number of times, the probability of winning the point from deuce is:

$$\sum_{k=0}^{\infty} [2p \times (1-p)]^k = \frac{1}{1 - [2p \times (1-p)]}$$

More detail on recursively calculating a player's probability of winning the match from any score given the two players' probabilities of winning on serve (PWOS) can be found in Barnett & Clarke (2002). Figure 2 demonstrates the end-result of this hierarchical Markov chain showing how the probability of Player 1 winning the match alters as the PWOS parameters of the two players are varied. On the left graph of the figure we can see the probabilities at the beginning of the match and on the right graph we can see how the odds have changed in favour of Player2 when the score-line is 0-1 Sets and 0-5 Games.

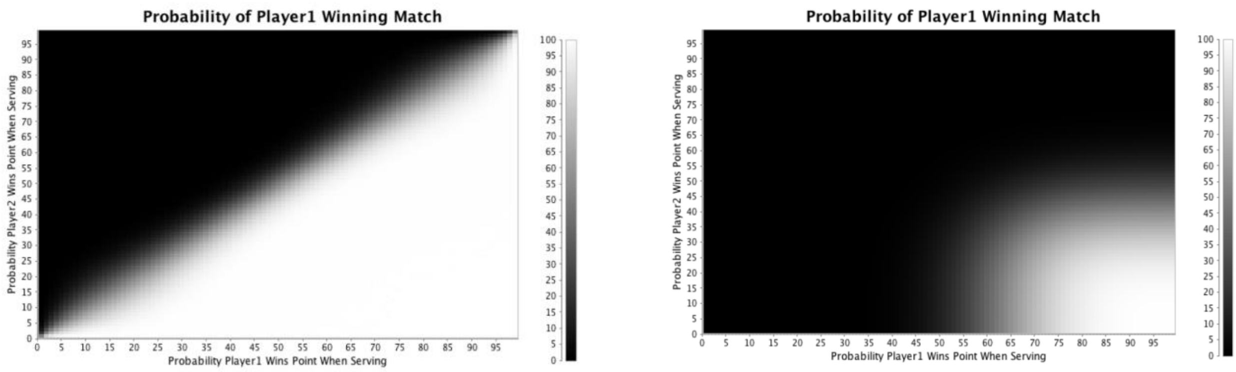

Figure 2 - A representation of the Probability of Player1 winning the match varying the probabilities of the players winning points while serving. (Left: from a score-line of 0-0 Sets, 0-0 Games, Right: from a score-line of 0-1 Sets, 0-5 Games and Player1 Serving)

Having discussed the higher levels of the hierarchical Markov model, all that is required to complete the model is the estimation of the parameter of the PWOS of the players involved in a match. These are the parameters which are used in the models to determine the probabilities of each player winning a game while serving and winning a tie-breaker.

There have been a few proposed methods for estimating the value of the PWOS parameters. A simplistic way would be to use the average number of points won while serving by each player over a period of matches and divide by the total points played while serving in the same period. Barnett & Clarke (2005) propose a technique of estimating PWOS by combining on a high level the statistics of the players compared and Spanias & Knottenbelt (2012) further refine this idea by modeling the point itself as a Markov chain to take into account individual strengths and weaknesses of players. Newton & Aslam (2009) also build upon Barnett & Clarke's idea by incorporating player instability in the estimation of the PWOS parameters. In Section 3.2 of this paper we will discuss a new concept for calculating adaptable PWOS parameters which take into consideration match odds from betting exchanges.

3. Tennis Score Inference

In order to do any sort of score inference from betting exchange odds, one first needs to estimate the implied-PWOS of both players from the live match odds, and then given the current known score, detect when a new point is scored. In this section we describe how one can calculate the implied probability of a player winning the match from the exchange matched odds. We then proceed to show a basic method of detecting when points are scored and then make suggestions on how to heuristically improve on this basic method.

3.1. Implied probability of winning the match from exchange odds

Live betting odds can be obtained from the world's largest betting exchange, Betfair. In our paper we used the Betfair API to retrieve information about the Match Odds betting market for a particular match at regular intervals (every second or so), for its entire duration.

The odds information supplied by Betfair is given in a decimal format, for example the odds of player A winning might be *2.36*. These can be easily transformed into the probability of player A winning the match by simply reciprocating the decimal odds.

P(player wins match) = 1 / decimal odds for player winning match = 1/2.36 = 42.4%

Hence, when the market displays decimal odds of value 2.36 for a player, this implies that the market roughly perceives the player to have 42.4% chance of winning the match. It is this implied probability which we will later use to infer the score of a match.

Another issue to consider is the potential of overrounds within the data collected from Betfair. Betfair's cross matching algorithm will sometimes round the odds offered. As a result, the sum of both players' implied probabilities, as taken from Betfair, sometimes exceeds 100%. The following example is taken from the Federer – Del Potro match (2012 Olympics), roughly two hours into the match. The odds are 1.46, and 3.0 for Federer and Del Potro, respectively. Converting these to probabilities results in a 68.49% probability in favour of Federer, and a 33.33% probability in favour of Del Potro. The resulting sum is 101.8%. To eliminate this overround, the values are normalized so that they sum to 100%.

3.2 Market Adaptive PWOS

In this paper we introduce a new method to calculate the PWOS of players involved which also takes into account the implied match winning probability provided by the exchange odds. The PWOS calculated in this way serves the purpose of inferring the score from the exchange odds in a much more efficient way as it accounts for offsets and adapts to the PWOS introduced by the market.

In their research, Klaassen & Magnus (2001) figured out the average sum of the PWOS for men is of 1.29, and 1.12 for women, with little variance. O'Malley (2008) shows that the probability of a player winning the match is determined by the relative difference between the two PWOS, rather than the absolute PWOS values. This is also visible in Figure 2 as it is the difference in the two PWOS values that cause the change in colour of the graph and not the absolute value. Therefore, given two equally strong male players, each of their PWOS values is approximated as 1.29/2 = 0.645. For players with different strengths, let 0.645 be the pivot value, and have the two players' PWOS values lay symmetrically on either side of the pivot:

PWOS(Player A) = 0.645 + difference/2,
PWOS(Player B) = 0.645 – difference/2.

The PWOS difference can then be calculated by solving for the difference whose match probability would yield the same normalised implied odds that can be found in the Betfair Match Odds market. The PWOS difference can therefore be estimated using the following binary search type algorithm.

```
calculatePwosDifference (impliedProb, score ){
        double pwosDifference = 0
        double increment = 0.5
        double calculatedProb = Calculator.calculateMatchWinningProb(score, pwosDifference)
        while (calculatedProb != impliedProb){
                if(calculatedProb < impliedProb){
                        pwosDifference += increment
                }
                else if (calculatedProb > impliedProb) {
                        pwosDifference -= increment
                }
                increment = increment/2
                calculatedProb = Calculator.calculateMatchWinningProb(score, pwosDifference)
        }
}
```

3.3. Score Inference

Before trying to infer the score, it is important to be able to extract an accurate probability for a player to win the match based on the activity on the betting market. We found that the best results came from using the ‘last price matched’ (LPM) values from Betfair, which gave the odds over which the last bet was made on the exchange.

To begin the score inference process, the current score-line (e.g. at the start of a match) must be known. The PWOS values for each player can then be calculated as outlined in Section 3.2, using the latest odds information from Betfair. By initializing our hierarchical Markov model with these PWOS values, we could therefore calculate the probability of either player winning the match from any given score-line, and hence get an idea of the odds we would expect to see from Betfair whilst the match is in play.

To track the score on a point by point basis, we chose one of the players and used our model to calculate the odds we would expect to see if the next point was won or lost. The actual live odds were then examined to see if there were any such changes. To allow for a slight variation from our model, we calculated a pair of threshold values that gave a fixed degree of leniency when testing the current Betfair odds against our predictions. Figure 3 gives a visual representation of the threshold values created for each point.

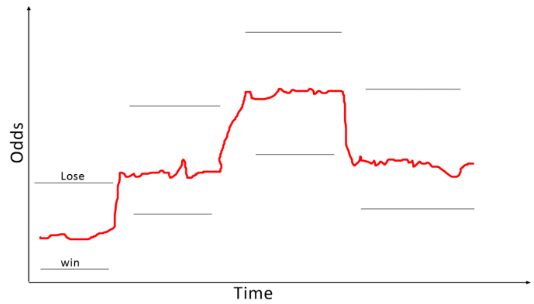

Figure 3 - Thresholds set while tracking live odds to detect the outcome of a point.

If the actual market odds breached either of the threshold values, we assumed a point had been resolved and the assumed score-line was appropriately updated. The new market odds were then used to recalibrate the PWOS estimations in attempt to capture player momentum, and the threshold process was repeated.

The success of this method was limited as the algorithm often either missed a point or detected a false positive. As this method is based on having a correct prediction of the score when the threshold values are generated, errors are carried over and rapidly derail the predicted score-line. From analysing the score inference errors alongside the correct score feed, there are occasions where the odds altered significantly without any change in the actual score-line, leading to a false positive. Figure 4 (Bartoli/Schiavone) shows a circular region where a point was predicted incorrectly.

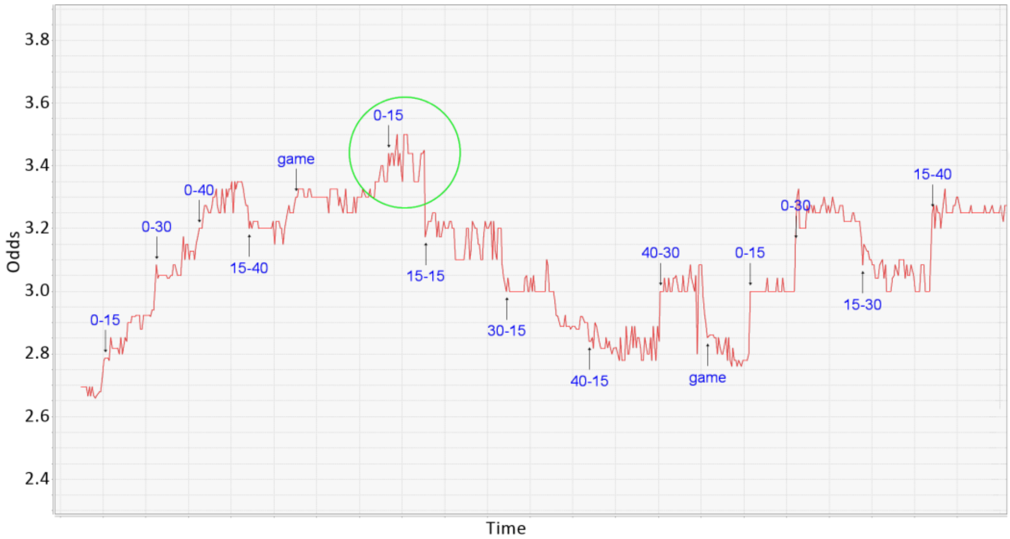

Figure 4 - Schiavone vs. Bartoli - Women’s singles Semi final. Roland Garros 2011. Graph of Bartoli’s odds during first/second game

In this case, looking at the match itself, the change in odds may be attributed to Schiavone missing an easy smash shot that in reality should have almost guaranteed her the point. For the financial incentive, bets may have been placed preemptively on Schiavone winning the point, causing a temporary shift in the market odds. Such activity highlights the need for any algorithm to be robust and able to handle errors should they occur.

3.4 Improvements

We can significantly improve inference accuracy by using additional heuristics. These techniques reduce false-positives and improve reliability by recognising and accounting for various market behaviour patterns.

3.4.1 Accounting for Volatility

False-positives often arise due to odds fluctuation between points. Therefore, we dynamically calculate odds volatility throughout a match and adjust point-scoring thresholds accordingly. Thresholds are widened when volatility is greater and vice versa. Knowledge of relative volatility is also useful in other heuristics, such as that described in section 3.4.2.

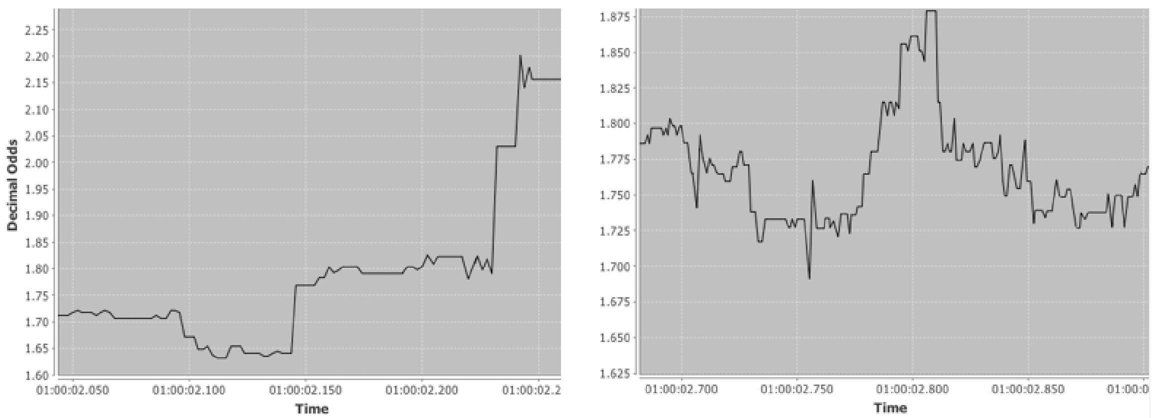

Figure 5 – LHS diagram shows match where volatility is low, score changes can easily be detected with narrow thresholds. RHS diagram shows match with high volatility. (Wozniacki vs. Pennetta - Qatar Ladies Open 2011 & Troicki vs. Djokovic - Sony Ericsson Open 2011)

We take volatility as the interquartile range of odds over the most recent 10 second period, updated continuously. Any statistical measure of spread may be used but we found interquartile range to be most suitable due to its robustness – it is less affected by small numbers of outliers than standard deviation, range, etc.

3.4.2 Recognising Large Odds Change as Scoring

We not only want to suppress false-positives but also ensure all scoring is captured. A change in odds is determined to be a scoring event if it is sufficiently large, sharp and sustained, even if the threshold is not crossed. *Sufficiently large* is defined relative to the difference between thresholds as well as adjusted for volatility, as described in section 3.4.1. When volatility is low, magnitude of odds change required for inference of scoring is lower than when volatility is high.

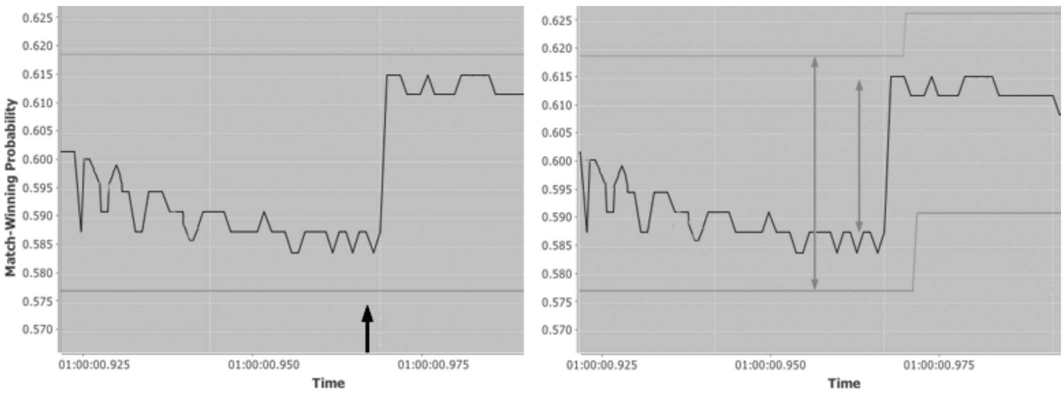

Figure 6 - Identical situation before and after implementing large odds change detection. Arrow on LHS indicates time of point scoring and subsequent odds change. Arrows in RHS denote threshold difference and odds change to which it is compared. In the RHS scenario, the program correctly detects scoring. (Wozniacki vs. Pennetta - Qatar Ladies Open 2011)

This is an important heuristic as we estimate that, out of all inferred points, two thirds are attributed to threshold crossing and the rest to odds-change detection.

3.4.3 Periodic Point Synchronisation

An incorrect inference of a single point may offset the score for the remaining match. Therefore, we periodically synchronise the score throughout the match. During changes of serve and end of sets, the match odds remain relatively constant for long periods. On detection of such patterns, we may recognise that our current score inference is incorrect and adjust it to the nearest game or set accordingly.

4. Case Study - Cilic vs Baghdatis, Roger’s Cup 2012

Time-stamped point-by-point data is not freely available in bulk in tennis. Consequently, the results we present in this paper are based on manually collected points and synchronized in time with live market information from Betfair.

Figure 7 shows Cilic’s match-winning probability as the match progresses, in his 2012 Roger’s cup match against Baghdatis played in Toronto, and compares the expected match-winning probability (calculated using a pair of static PWOS values and the actual scoreline) and the probability implied by the Betfair odds. It can be seen that, while the trends of the two curves are very similar, the two values can diverge, emphasising the importance of dynamically updating the PWOS values to agree with the market odds.

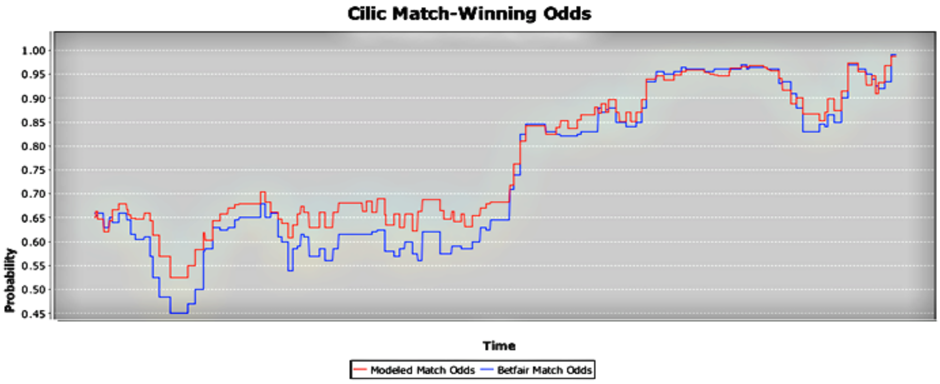

Figure 7- A comparison of modelled match odds with Betfair implied odds of Cilic winning the match against Baghdatis in Roger's Cup 2012.

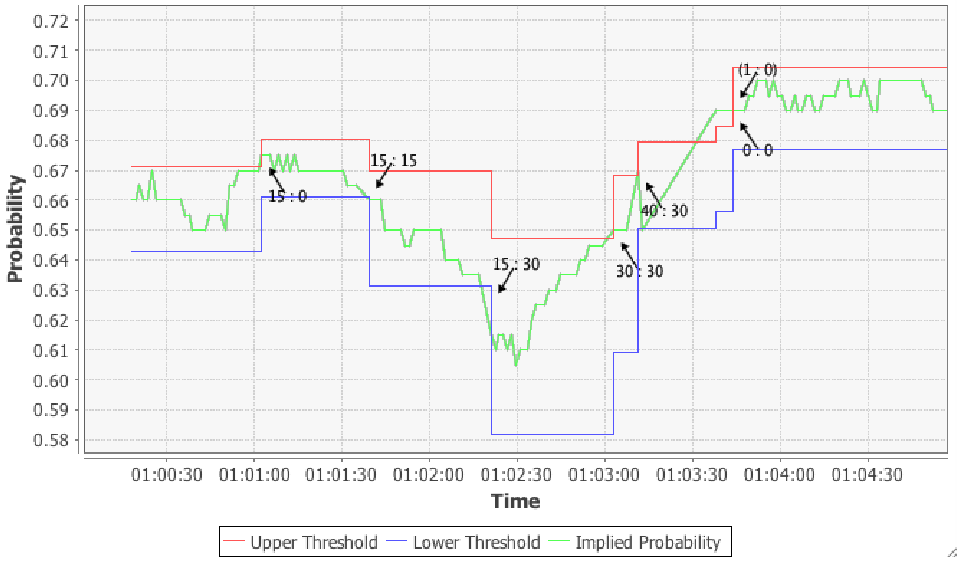

Figure 8 – First game of the Cilic vs. Baghdatis match in Roger's Cup 2012

Figure 8 shows the first game of the Cilic vs. Baghdatis match, and the corresponding thresholds used to infer scores. All of the inferred scores are correct. We note that the incorrect inference of a point being scored shortly after 40-30 is avoided by the use of a timeout mechanism that enforces a delay of at least 8 seconds between points.

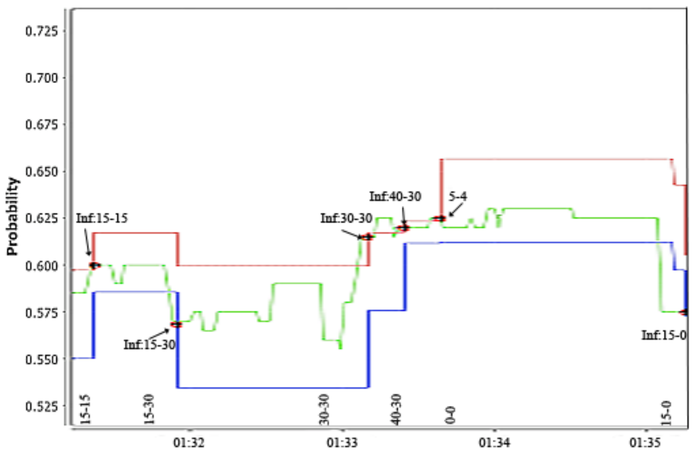

Figure 9 - Live capture of the system's point thresholds and the Betfair implied odds in the match Cilic vs. Baghdatis Roger's Cup 2012. The dots represent the different points detected.

Points continue to be detected correctly up to 5-4 in the first set, the points around which are shown in Figure 9. We note the delayed detection of the 30-30 point, which we speculate might have been caused by market uncertainty about the point outcome (e.g. because of a challenge). In the next game, the inference of a key point is missed because of an insufficient movement in market odds. This rapidly leads to a decoupling of the model and market's implied match odds (even when the former uses dynamically updated PWOS values, since these are now based on different score lines), resulting in cascading point errors. Indeed, while the final score is 7-5, 6-3 in favour of Cilic, the score-detection algorithm infers the final score to be 6-4, 5-3.

5. Conclusion

Using a hierarchical Markov scoring model, the parameters of which automatically adapt to live betting exchange odds, we have managed to build a system which can infer when points are scored in a match from live betting exchange market data. This is done by detecting when the live odds cross some pre-estimated upper or lower bound and depending on which bound was crossed the system interprets it as a point scored by a particular player. The system is not perfect and any errors made in the form of false positives or missed points carry over to future deductions can quickly destabilize the system. Nevertheless heuristic improvements have been added to the system and are effective at improving detection of points and reducing the effect of errors on future deductions.

References

Barnett, T. J. & Clarke, S. R., (2002) Using Microsoft Excel to model a tennis match. *6th Australian Conference on Mathematics and Computers in Sport (G. Cohen ed.),* pp. 63-68.

Barnett, T. J. & Clarke, S. R., (2005) Combining player statistics to predict outcomes of tennis matches. *IMA Journal of Management Mathematics,* pp. 113-120.

Klaassen, F. J. G. M. & 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,* 96(454), pp. 500-509.

Knottenbelt, W. J., Spanias, D. & Madurska, A. M., (2012) A common-opponent stochastic model for predicting the outcome of professional tennis matches. *Computers & Mathematics with Applications,* 64(12), pp. 3820 - 3827.

Liu, Y., (2001) Random walks in tennis.. *Missouri Journal of Mathematical Sciences,* 13(3).

Newton, P. K. & Aslam, K., (2009) Monte Carlo Tennis: A Stochastic Markov Chain Model. *Journal of Quantitative Analysis in Sports,* 4(3), pp. 1-42.

O'Malley, J. A., (2008) Probability Formulas and Statistical Analysis in Tennis. *Journal of Quantitative Analysis in Sports,* 4(2), p. 15.

Spanias, D. & Knottenbelt, W., (2012) Predicting the outcomes of tennis matches using a low-level point model.. *IMA Journal of Management Mathematics.*