Barto, A Bayesian Binomial Rating Method Applied to Professional Tennis Matches

by
Bart von Meijenfeldt (ANR 445626)

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Econometrics and Mathematical Economics

Tilburg School of Economics and Management
Tilburg University

Supervised by:
Prof Dr. J.H.J. Einmahl
July 9, 2018

Abstract

A wide variety of research has been conducted into predicting tennis matches. The most successful one being the Elo model.The Elo model gives each player a skill level, called a rating and increases this rating appropriately after a win and decreases it after a loss. A limitation of this model is that is only uses the win/loss result of a match to update the ratings. Therefore this paper creates a rating-based model which is able to use a more detailed result (for example the result on set level) to update the ratings. The idea being that the strength of a player can be better measured from this result. This model outperforms the Elo model when predicting the top 30% of matches played in 2013-2014, for a total of 1288 matches. This model is applicable to all sports which have a set or game structure such as volleyball, darts and snooker.

Contents

1 Introduction

Sports can be enjoyed in many ways. When one is young, one often participates oneself. But as one gets older, more often one tends to watch professionals play and find enjoyment in the post-game analysis or the ranking of players or teams. In many sports one needs to put effort in a good ranking oneself.

But in chess there is a quite reliable Bayesian rating method available at the Fide website [1]. This is due to Arpad Elo who created this Bayesian method to rank player's levels. A simplified version of his original statistical method is implemented by the Fide (International Chess Federation) and available for all to see. This method can be applied to all sports where two players or teams compete.

But Glickman (1999) decided to extend the Elo method such that it would also take in account the reliability of a player's rating by introducing the standard deviation of the rating to the method. This rating deviation increases each period in which no game has been played and reduces after a game is played. As a reference to the Elo method, he named his method Glicko.

These two methods are generally applicable to most sports in which only two players or teams participate. But there are also tennis-specific models created from which the most interesting is the common opponent model created by Knottenbelt, Spanias, and Madurska (2012). This model uses the relatively sparse player field in top tennis matches, resulting in players often having played the same opponents in the last year. The match results of these matches are then used to compare the players' strengths and to predict their result often using Markov chains.

Recently Kovalchik (2016) published a literature overview in which she found that the most promising model is the Elo model as adapted by FiveThirtyEight [2]. FiveThirtyEight uses an updating rule in which the number of matches a player has played changes the updating rule. Players with few matches have a more volatile rating than players who are more experienced. This model is the current state-of-the-art.

But this model only uses the binary result of the match (win/loss) to update a player's rating. Room of improvement might be found by using a rating system which uses a more precise match result, for example on set level. Therefore the goal of this research is to create a Bayesian rating system which uses a more precise match outcome to improve on the current

[1] www.fide.com, the official website of the International Chess Federation

[2] www.fivethirtyeight.com is a website which is occupied with predicting

state-of-the-art model, Elo.

The rest of this paper is structured as follows. Section 2 discusses the previous work on which this paper builds. Section 3 outlines the model we will build to attempt to improve on the Elo model. Section 4 discusses how we will apply this model on tennis matches. Section 5 describes the data set. Section 6 demonstrates how we estimate the hyperparameters of both the Elo model and our new model. Section 7 reports the obtained results. Section 8 concludes this paper.

2 Bayesian Models

2.1 Bayes Theorem

One might not be surprised to hear that the Bayesian models which will be discussed in this section are based on the Bayes theorem. We have written it in a form which is relevant to tennis prediction below:

$$P(\mu_a, \mu_b | M) = \frac{P(M|\mu_a, \mu_b)P(\mu_a, \mu_b)}{P(M)}, \tag{1}$$

in which $P(\mu_a, \mu_b)$ is defined as the prior, the initial degree of belief in μ_a and μ_b, the variables denoting the ratings of the players competing in a match. $P(\mu_a, \mu_b|M)$ the posterior probability in μ, the degree of belief in μ after accounting for M, the match statistics. $P(M|\mu_a, \mu_b)$ is called the likelihood, the probability that the match results occurred given the ratings.

2.2 Elo

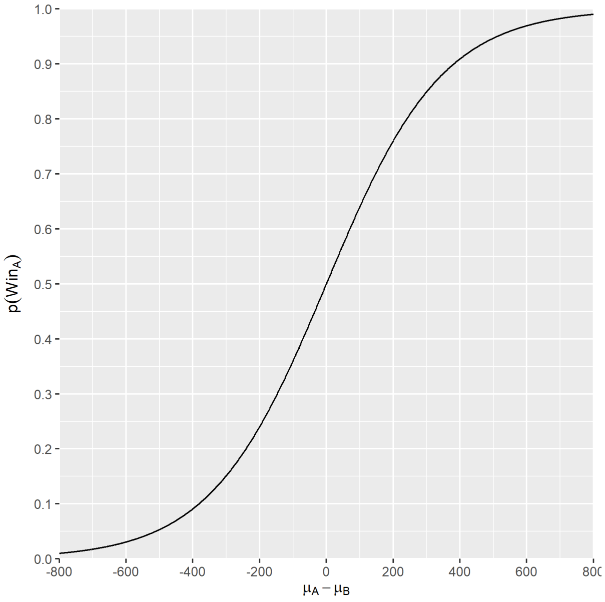
The original Bayesian system for match prediction widely used is invented by Elo (1978). He suggested a method which initializes the rating, μ, of a new player to a number, in this paper we will use 1500. Whenever two players play a match, the method uses the formula defined below to estimate player A's chance of winning versus player B:

$$P(W_A = 1|\mu_A, \mu_B) = \frac{1}{1 + 10^{-(\mu_A - \mu_B)/400}}. \tag{2}$$

Where W_A is a dummy variable signifying a win for player A. To simplify notation we will write $P(W_A)$ when we mean $P(W_A = 1|\mu_B, \mu_B)$ in the rest of this paper.

In Figure 1 rating differences starting from -800 up to 800 versus the probability of winning for player A are plotted. It is a continuous function with S-shaped values, also known as a sigmoid function. Meaning that a rating change for similarly rated players results in a greater change in probability than unevenly matched players. Note that the function values are always between 0 and 1.

Figure 1: Rating differences versus chance of winning



The prediction is combined with the match result to calculate the posterior rating of the players in the following manner for player A (and in similar manner for player B):

$$\mu'_A = \mu_A + K \cdot (W_A - P(W_A)). \tag{3}$$

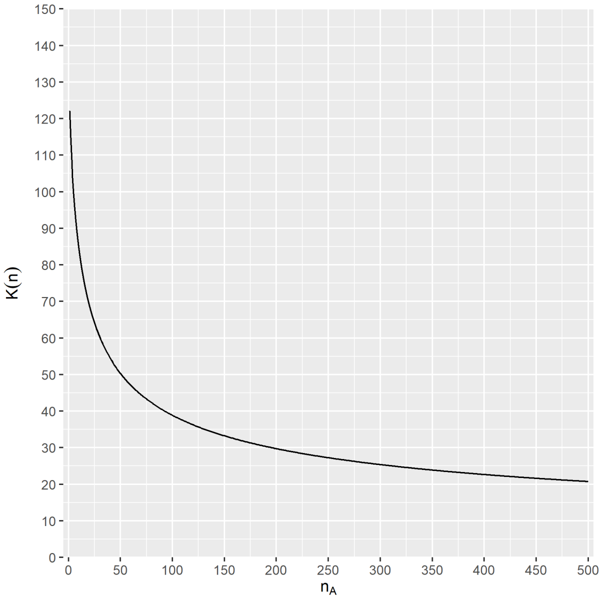
This makes $W_A - P(W_A)$ the surplus result. A number higher than 0.5 signifies that player A was the underdog yet won, a number smaller than 0.5 but higher than 0 means that player a was the favorite and won, a number smaller than 0 but higher than -0.5 means that player A was the underdog and lost, and lastly, a number smaller than -0.5 means that player A was the favorite yet lost. Note that $-1 < (W_A - P(W_A)) < 1$. K defines the learning rate of the rating. So this judges how to weight the old rating versus the surplus result. μ_A defines the prior rating and μ'_A the posterior rating.
In Elo's original model K is a constant, but in FiveThirtyEight's adaption K is a function depending on the number of matches played:

$$\mathrm{K}_{538}(n_A) = \frac{250}{(n_A + 5)^{0.4}}. \tag{4}$$

Equation (4) is plotted in Figure 2. One can see that K's value drops sharply for the first few matches but decreases less the more experience a player has. This translates to high uncertainty about a player's rating in the first few matches, since the result of one new game can change a player's rating quite drastically. The more matches a player played, the more certainty about a player's strength and the changes become smaller and smaller.

In Section 6.1 we find optimal values for both the constant K that Elo proposed and for the formula of FiveThirtyEight and we choose our favorite model.

Figure 2: K-formula from FiveThirtyEight



2.3 Glicko

As was written in the introduction, Glickman invented a method named Glicko that does not only have a rating parameter, μ, but also has a rating deviance parameter, RD. It uses both the rating and the rating deviance to predict a match outcome. Just as in Elo's method we will initialize ratings to 1500. One also needs to set the initial rating deviance, which will also be the maximum rating deviance a player can have. We will call this value RD_{MAX}. The last variable one needs to set is the rating deviance increase variable, ν. This variable will be used to update a player's rating deviance after each period in the following manner:

$$RD'_A = \min(\sqrt{RD_A^2 + \nu^2}, RD_{MAX}). \tag{5}$$

So the rating deviances increases every period up to a specified maximal rating deviance. But after each played match the uncertainty of a player's strength reduces and therefore the rating deviance decreases using a Bayesian updating rule. The updating rule Glickman uses are of a complex nature therefore we do not show them in this paper. Details can be found in Glickman (1999), we will summarize the most relevant properties of his updating rules in the rest of this section.

After each game the updating rules follows from assuming Gaussian uncertainty in the parameters μ'_A, μ'_B with respective means μ_A, μ_B and respective standard deviation RD_A, RD_B and using Equation (2) to transform rating differences into probabilities. The fundamental difference between Elo's model and Glickman's model is that Elo's model implicitly assumes μ to be a player's exact rating while Glicko assumes μ to be an estimate of the mean of a player's true rating, allowing uncertainty.

The probability of a win is based on the same sigmoid function as is used by Elo. This means that the ratings are on the same scale, 100 rating gain in Elo's method means the same shift in strength as 100 rating gain in Glickman's method. Glickman also shows in this paper that when one takes RD $= 0$ that his updating rule, is the same as Equation (3), Elo's updating rule.

2.4 Scores

Both of these methods only utilize the binary result, the match outcome, to update a player's rating, without looking at a match score. This makes sense when one is looking at a chess match, for which a score system within a match is difficult to formalize. In many sports not utilizing the score results wastes information.

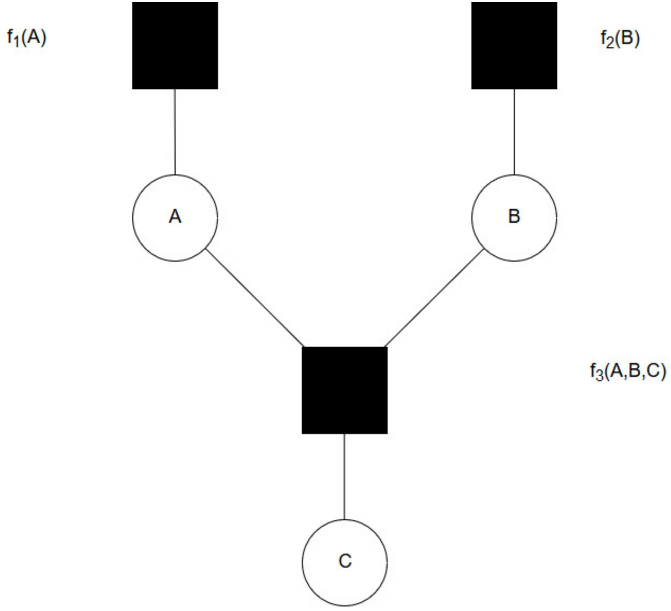
Let us take an example based on tennis. Imagine two players A and B both have a rating of 1500. They both play versus an opponent which has a rating of 1400. Player A wins 3 sets without dropping a single set. Player B plays a 5-setter, winning 3 sets and losing 2. In both the Elo and Glicko method player A gets the same rating increase as player B. Yet intuition indicates that player A should receive a higher increase in rating than player B, because he probably performed better.

Therefore we will create a method in Section 3, which is able to utilize this information to hopefully model a player's skill more accurately. This model will be applicable to all games which have a binomial structure, such as sports which have a game/set structure or point-based games.

2.5 Factor Graphs

A factor graph allows us to break up a joint probability distribution in smaller parts, which often allows one to better understand the connection between variables. This will be used in Section 3.1 to visualize the model. The graph uses two types of nodes, factors (functions) which are represented as boxes and variables which are represented as circles. The edges, the lines between the nodes, show the dependencies between the nodes.

Figure 3: Example factor graph



A factor graph of the simple function $f_1(A)f_2(B)f_3(A,B,C)$ is shown in Figure 3. Each dependency between a variable and a factor has one edge between them.

2.6 Logistic Regression

Since the results of our matches are either a win (1) or a loss (0) we need a model which is suited for dealing with binary results. The model we chose is the logistic regression developed by Cox (1958) . Its general latent form is given below:

$$\begin{aligned} y_i^* &= \beta_0 + \beta_1 x_i + \epsilon_i \\ y_i &= 1[y_i^* > 0], \end{aligned} \tag{6}$$

where y_i^* is an unobserved variable and y_i is the observed variable (representing in our case a win or loss). x_i are the explanatory variables. β_0 is the constant, and β_1 signifies the effect of x_i on y_i^*. The error term, ϵ_i, is assumed to be distributed by the standard logistic distribution. Resulting in the probability of winning being defined as shown below:

$$P(y_i = 1|x_i) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_i}}. \tag{7}$$

2.7 Log Loss

To benchmark the model we will develop in Section 3, with the Elo model we need to have a metric to compare the models. We will choose one which is typically chosen in logistic regressions, the log loss. It is defined as the negative value of the averaged log likelihood:

$$LL = -\frac{1}{n} \sum_{i=1}^{n} y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}). \tag{8}$$

For example if two matches were played in the first match the probability of player A winning is estimated to be 0.8 and for the second match the probability is estimated to be 0.6. The first match player A wins, but the second one player B wins. This results in a log loss of:

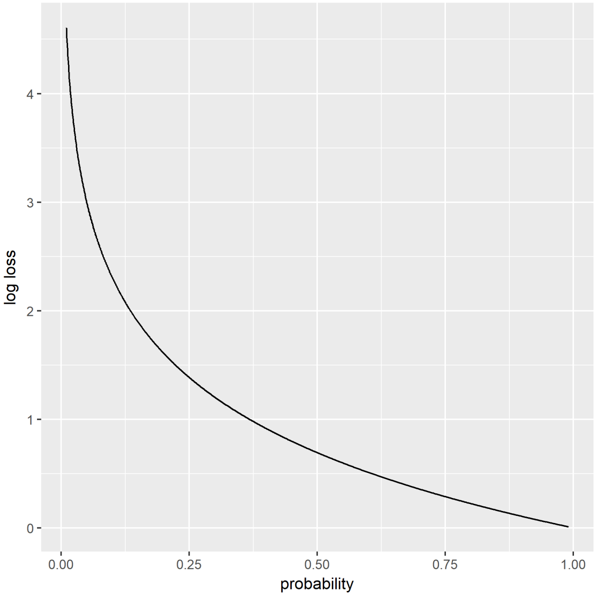
$$LL = -\frac{1}{2}(log(0.8) + log(0.4)) = 0.57. \tag{9}$$

As can be seen in Figure 4 the log loss punishes overconfident predictions strongly. If one predicts the chance of an outcome to be 0.99 and the other outcome realizes, causes a loss of $-log(0.01) = 4.61$ while predicting safely that the outcomes are both equally likely and having the other outcome realized only leads to a loss of $-log(0.5) = 0.69$.

2.8 Bootstrapping

To obtain some statistics in Section 7 we will use a technique created by Efron (1979) called bootstrapping. Bootstrapping is any test or metric that

Figure 4: log loss per probability



relies on random sampling with replacement to sample estimates. The size of the samples is the same as the size of the data. For example if there are five observations, x_1, x_2, ..., x_5 the following are all bootstrap samples, $[x_2, x_2, x_4, x_3, x_5]$, $[x_2, x_2, x_2, x_2, x_5]$ and $[x_1, x_2, x_3, x_4, x_5]$. The key idea is that by creating many of these samples to represent the population of which this data comes from and using this to calculate relevant statistics. This technique is useful when the distribution of statistics one wants to calculate are complicated to calculate or impossible.

3 Barto

3.1 Model

We assume that a player A and a player B play a match consisting out of n smaller contests. In each of these contests there will be one winner, player A or player B. The probability of player A winning a contest, p_C, will be constant throughout the match. The total number of contests won by player A will be referred to as w.

We model the probability of winning a contest, p_C, in a similar manner as Elo (Equation (2), but with the base 10 replaced by the base e, resulting in the following sigmoid function:

$$p_C = P(C_A = 1) = \frac{1}{1 + e^{(P_B - P_A)\beta_p}}, \tag{10}$$

where C_A stands for player A winning a contest, P_i for the level of the performance of a player during a match and β_p is a parameter which scales the difference in performance levels to a contest probability. The difference in performance levels does not translate to a probability of winning a match in the same manner as the Elo and the Glicko method. The main reason is that the difference in rating is implies the probability of winning a contest instead of winning a match. We also used a slightly different sigmoid function to prevent confusion on this matter.

We assume the performance levels to be distributed in a Gaussian manner with the mean the true strength of a player S_i and the standard deviance the day form, β_d. The reason for using P_i instead of S_i in Equation (10) is to account for match forms which do not reflect the true strength of a player. We assume S_i to not be perfectly known, but instead assume Gaussian uncertainty. To recap, we assume the following distributions:

$$\begin{aligned} &S_A \sim \mathcal{N}(\mu_A,\, \sigma_A^2), S_B \sim \mathcal{N}(\mu_B,\, \sigma_B^2) \\ &P_A|S_A \sim \mathcal{N}(S_A,\, \beta_d^2), P_B|S_B \sim \mathcal{N}(S_B,\, \beta_d^2) \\ &w|P_A, P_B, n \sim B(n, \frac{1}{1 + e^{(P_B - P_A)\beta_p}}). \end{aligned} \tag{11}$$

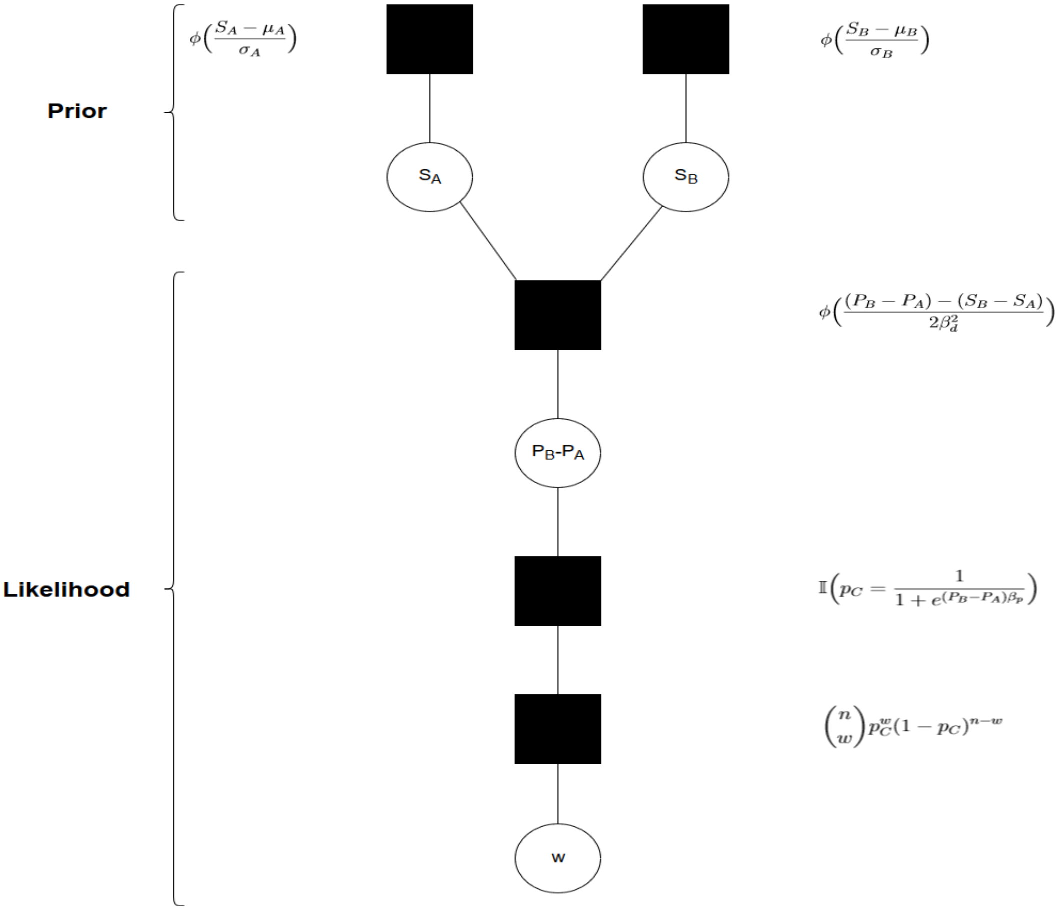
We also assume both the performance levels to be independently distributed, which means that subtracting them leads to:

$$P_B - P_A|S_B - S_A \sim \mathcal{N}(S_B - S_A,\, 2\beta_d^2). \tag{12}$$

Using these distributions, a factor graph representation of the model is made and shown in Figure 5 to better visualize the joint density probability of a match. Note that in this model we allow different standard deviations in the priors, this can be useful for example to update ratings of inexperienced players faster than experienced players as is done in the Glicko model. We do assume the performances of the players to have equal variance, which can be seen by the variable shared by both players day performance, β_d.

In practice we will have for each player a μ which estimates a player's strength, thus is the rating variable of the model and needs to be initialized. Each player will also have a σ, which is the standard deviation signifying how much this strength can deviate from μ. β_d indicates how much a match form influences a player's strength. Factors which can influence a day form are for example motivation and fatigue. The slope of the sigmoid function

Figure 5: A factor graph of the Barto


$\phi\left(\frac{S_A - \mu_A}{\sigma_A}\right)$
$\phi\left(\frac{S_B - \mu_B}{\sigma_B}\right)$
Prior
S_A
S_B
$\phi\left(\frac{(P_B - P_A) - (S_B - S_A)}{2\beta_d^2}\right)$
P_B-P_A
Likelihood
$\mathbb{I}\left(p_C = \frac{1}{1 + e^{(P_B - P_A)\beta_p}}\right)$
$\binom{n}{w} p_C^w (1 - p_C)^{n-w}$
w

scaling the difference in performance levels to the probability a contest is won is given by β_p. A higher value for this parameter mean that bigger difference in performances are required for changing the probability of winning with the same amount.

3.2 Posterior Probability

Using Bayes' theorem stated in Section 2.1 we can calculate the posterior probability of a given rating given the match result. We will look at the two relevant match results, n the number of contests played, and w the number of contests won by player A. Using Bayes' theorem we know that:

$$P(S_A = s_a, S_B = s_b | w, n) = \frac{P(w|n, S_A = s_A, S_B = s_b) P(S_A = s_a, S_B = s_b | \mu_A, \mu_B)}{P(w|n)}. \tag{13}$$

For this result we assume that n is deterministic. In the rest of this paper we will only write down S_A or S_B as shorthand notation for respectively $S_A = s_a|\mu_A$ and $S_B = s_B|\mu_B$.

3.3 Maximizing Posterior

We use this posterior probability to be able to update both players' rating after each match. This is done by estimating the most highly probable posterior strengths for both players given the match result. This is defined in mathematical terms below. After each match we set both player's ratings, μ_A and μ_B, equal to the estimates of the players' skill:

$$(\mu'_A, \mu'_B) = \underset{S_A, S_B}{\arg\max}\, P(S_A, S_B | n; w) = \underset{S_A, S_B}{\arg\max}\, \frac{P(n, w|S_A, S_B) P(S_A, S_B)}{P(w|n)}, \tag{14}$$

where μ' stands for a player's new rating, after a match is played. Since the denominator does not depend on the S_A and S_B, Equation (14) reduces to:

$$\underset{S_A, S_B}{\arg\max}\, P(S_A, S_B | n, w) = \underset{S_A, S_B}{\arg\max}\, P(n, w|S_A, S_B) P(S_A, S_B). \tag{15}$$

To continue we need to define the prior and the likelihood in Equation (15). For the priors we use the Gaussian probability density functions. For the likelihood we have to account for all possible values of the differences in the Gaussian distributed day performances, P_A and P_B, and the binomially distributed variable w. This results in the formula defined below:

$$\begin{aligned} &P(n,w|S_A,S_B) \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi 2\beta_D^2}} e^{-\frac{1}{2}\left(\frac{(p_d)-(S_B-S_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w} \left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w} dp_d. \end{aligned} \tag{16}$$

So to update the ratings after a match, we want to solve the following equation after each match:

$$\begin{aligned} &\arg\max_{S_A,S_B} P(S_A,S_B|n;w) \\ &= \arg\max_{S_A,S_B} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi 2\beta_D^2}} e^{-\frac{1}{2}\left(\frac{(p_d)-(S_B-S_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w} \left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w} d_{p_d} \\ &\qquad \frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{S_A-\mu_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{S_B-\mu_B}{\sigma_B}\right)^2}. \end{aligned} \tag{17}$$

We can redefine the solution in Equation (17) for S_A and S_B in terms of the change relative to the old rating:

$$\begin{aligned} S_A &= \delta_A + \mu_A \\ S_B &= \delta_B + \mu_B \end{aligned} \tag{18}$$

This results in the form:

$$\begin{aligned} &\arg\max_{\delta_A,\delta_B} P(\delta_A,\delta_B|n;w) \\ &= \arg\max_{\delta_A,\delta_B} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi 2\beta_D^2}} e^{-\frac{1}{2}\left(\frac{(p_d)-(\delta_B-\delta_A+\mu_B-\mu_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w} \left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w} d_{p_d} \\ &\qquad \frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2}. \end{aligned} \tag{19}$$

Now suppose some solution is found, in the likelihood (the first line of Equation (19)) this solution only depends on $\delta_B - \delta_A$ which equals some number, let's say q. Equiped with this information we know that the priors (the second line of Equation (19)) are optimized given that $q = \delta_B - \delta_A$.

$$\begin{gathered}\operatorname*{arg\,max}_{\delta_A,\delta_B} \frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2} \\ \text{subject to } q = \delta_B - \delta_A\end{gathered} \tag{20}$$

Since the priors are exponential functions, which are convex, we can solve above maximization problem by using the Lagrange function:

$$L(\delta_A, \delta_B, \lambda) = \frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2} - \lambda(\delta_B - \delta_A - q).$$

Taking the derivative leads to the following set of equations:

$$\begin{cases} \frac{dL(\delta_A,\delta_B,\lambda)}{d\delta_A} = -\frac{\delta_A}{\sigma_A^2}\frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2} + \lambda = 0 \\ \frac{dL(\delta_A,\delta_B,\lambda)}{d\delta_B} = -\frac{\delta_B}{\sigma_B^2}\frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2} - \lambda = 0 \\ \frac{dL(\delta_A,\delta_B,\lambda)}{d\lambda} = \delta_B - \delta_A = q \end{cases}$$

, replacing the λ in the first line with the value λ takes in the second line (after moving the λ to the right-hand side of the equation) leads to the following equation:

$$\left(\frac{\delta_A}{\sigma_A^2} + \frac{\delta_B}{\sigma_B^2}\right)\frac{1}{2\sigma_A\sigma_B\pi} e^{-\frac{1}{2}\left(\frac{\delta_A}{\sigma_A}\right)^2} e^{-\frac{1}{2}\left(\frac{\delta_B}{\sigma_B}\right)^2} = 0 \Longleftrightarrow \delta_A = -\frac{\sigma_A^2}{\sigma_B^2}\delta_B. \tag{21}$$

This result demonstrates that if one player's rating increases, it will be at the expense of his opponent. It also shows that the rating of a player with a higher uncertainty will update faster than his opponent. But the most relevant consequence of this result is that we can write one of the variables in terms to optimize over in terms of the other variable, making it a one-dimensional optimization problem. This reduces the time needed to calculate the posterior, which is useful when we start working with the tennis data in Section 6, since we will be working with tens of thousands of matches. Below we created a one-dimensional problem by substituting δ_A:

$$\begin{aligned}
&\arg\max_{S_A,S_B} P(S_A, S_B|n;w) = \\
&\arg\max_{\delta_B} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}2\beta_D^2} e^{-\frac{1}{2}\left(\frac{(p_d)-((1+\frac{\sigma_B^2}{\sigma_A^2})\delta_B+\mu_B-\mu_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w} \left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w} d_{p_d} \\
&\qquad \frac{1}{2\sigma^2\pi} e^{-\frac{1}{2}\left(\frac{\sigma_B^2\delta_B}{\sigma_A^3}\right)^2+\left(\frac{\delta_B}{\sigma}\right)^2}.
\end{aligned} \tag{22}$$

In the rest of this paper we assume the variances in the priors to be equal for all players, yet we have derived the general form which might be useful for further work.

3.4 Estimation Barto

We update a player's rating using numerical approximation. This means that we do not find a closed-form solution to Equation (22). Instead we approximate the integral using the method of Romberg (1955) based on the trapezoidal rule. We use the method of Brent (2013) to solve the one-dimensional optimization problem. In Figure 6 and Figure 7 we plotted respectively the likelihood versus the performance difference and the posterior versus the rating difference.[3] Both functions appear to be well-behaved continuous bell-shaped functions, therefore we expect no problems using these approximations instead of closed-form solutions.

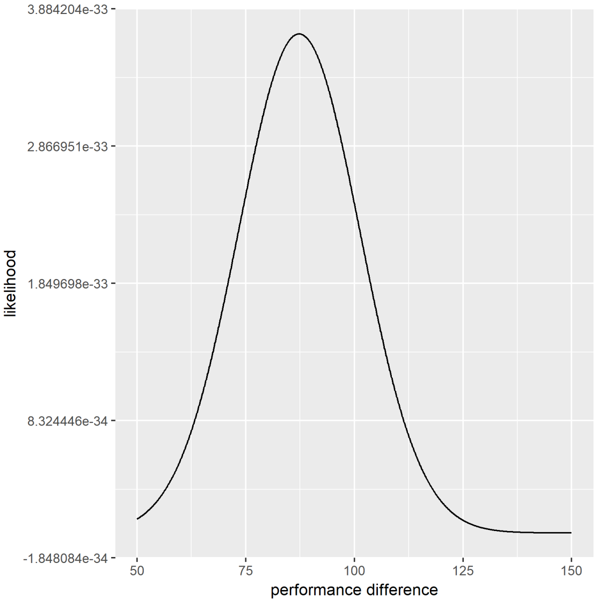
4 Tennis

4.1 Scoring

Tennis is a sport consisting out of sets, games and points. In each game one player is designated to be the server while the other play is the returner. The server is the person who hits the ball first, and the returner has to return the ball back to the server who then tries to hit the ball back to the returner. This goes on until a mistake is made by either player, e.g. a player hits the ball into the net. This gives his opponent a point. If one of the players has at least four of these points, while maintaining a 2 point lead this player receives a game and the point score is reset to 0-0. In the next game the

[3] For the values $\mu_1 - \mu_2 = 100, n = 100, w = 70, \beta_p = 1/400, \beta_d = 20, \sigma = 10$

Figure 6: Likelihood $\frac{1}{\sqrt{2\pi}2\beta_D^2 e^{-\frac{1}{2}\left(\frac{(p_d)-(2\delta_B+\mu_B-\mu_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w}\left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w}}$



player who served now has to return the ball and vice versa. In a similar manner a player is awarded a set when winning at least 6 games with a two game lead. If the score in games is 6-6 a tiebreaker occurs to decide the winner of the set. Each match is either a best of 5, where the winner is the person who first wins 3 sets or best of 3 which is won by the player reaching two sets first [4].

4.2 Surfaces

Professional tennis is played on three different types of surfaces, hard, grass and clay [5]. All of these surfaces create slightly different dynamics. For example the ball bounces fastest on grass. A player who relies on his fast serve is therefore likely to excel on grass surface. Because of the differences that can occur over surfaces, we will only focus our analysis on the surface on which most matches are played, the hard surface.

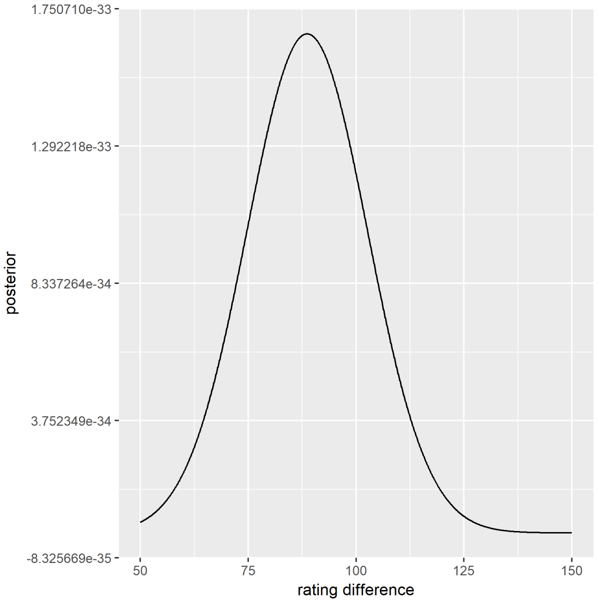
[4] This is a simplified explanation, the most up-to-date full set of rules can be found at www.atpworldtour.com/en/corporate/rulebook.

[5] Up to 2009 professional matches were also played on carpet.

Figure 7: Posterior

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}2\beta_D^2} e^{-\frac{1}{2}\left(\frac{(p_d)-(2\delta_B+\mu_B-\mu_A)}{\sqrt{2\beta_d^2}}\right)^2} \binom{n}{w} \left(\frac{1}{1+e^{(p_d)\beta_p}}\right)^n \left(e^{(p_d)\beta_p}\right)^{n-w} d_{p_d}$$

$$\frac{1}{2\sigma^2\pi} e^{-\frac{1}{2}\left(\frac{\sigma_B^2\delta_B}{\sigma_A^3}\right)^2+\left(\frac{\delta_B}{\sigma}\right)^2}$$



4.3 Barto

To be able to model tennis matches using the Barto there needs to exist a multitude of smaller contests which has one winner per contest and for which the probability is the same for each contest.

Since both for winning on serve the probabilities are approximately independent and identically distributed as well as winning on return (Klaassen and Magnus (2001)), sets are approximately independent and identically distributed as well. Therefore one possibility is to take the sets as a contest. This leads to a relatively low level of contests in a match but the contests are directly linked to the match outcome, i.e. a player who wins most sets wins the match.

If we were to proceed in this manner, a player has one rating denoting a player's strength as is the case in the Elo method. We would use a more accurate representation of the score compared to Elo's method, possible leading to a more accurate measure of a player's strength and therefore to improve prediction. This has the drawback of having only a maximum of 5 contests per match to gauge the players' strength. We are also able to take as the contests the smallest measure possible, a point. This has the advantage that on average we measure 160 small contests. But this would have as drawbacks that the contests are not directly related to the result of the match and that the probability in each contest is not constant anymore since it depends on who is serving.

Whether the contests not directly being related to the results is truly a problem is debatable, since points are approximately i.i.d. it can be argued that to find out a player's strength one needs to only look at the results on points level, the final score does not matter.

To solve the second problem we have to separate each match into two matches, one match where player A serves versus player B, and one match where player B serves versus player A. This means that a player will have two different type of ratings, one for his serve and one for his return we add these two ratings together for a player for prediction later on.

We will create the Barto rating variables both for the high level contests, sets, and for the low level contests, points. In the case of the high level contest we will initialize the ratings of the players at 1500. Since in our data we can see that approximately 69 percent of points are won on service. Therefore to adjust for this for the points data we will initialize the serve rating at 1650 and the return rating at 1350. As later will be seen we fix $\beta_p = \frac{1}{400}$ and this 300 rating difference for starting players results in a expected chance of winning a point of 68 percent, close enough to the real value.

4.4 Benchmarking

To compare which model performs best on the hard surface, we will also create the Elo rating variables. In fact we create for both Barto and Elo the general ratings for the players (using all matches played on all surfaces) as well as a hard-specific court rating. For each method, both of these variables will then be regressed on the (binary) match outcome. Using the parameters resulting from the logistic regression we will then predict the match outcomes on unseen data. Then we will measure the best model by computing the log loss of these predictions for each method.

5 Data

5.1 Description

We take Jeff Sackman's data [6], which contains ATP matches starting from January 1968 up to 11 september 2017. This data contains all sort of match information from which we outline the most important in Table 1

Table 1: Features of Jeff Sackman's data

Features	Explanation
Match Information	
Tournament	Name of the tournament
Date	Date of the tournament
Surface	Surface on which the match is played
Best of	Best of 3 or 5 sets
Player Information	
Names	Names of both players
Rank	ATP ranking of both players
Match Results	
Score	Denoted in games e.g. 6-3 4-6 6-4
Minutes	Duration of the match in minutes
Points played	Number of points a player played on serve
1st won	Number of points a player won on first serve
2nd won	Number of points a player won on second serve
Comment	Comment denoting a special event, e.g. retired or walkover

We take the matches from January 2000 up to December 2014. In total there are 47221 ATP matches in this period. We split this data in four

[6] www.github.com/JeffSackmann/tennis_atp

parts. We use the data in the years 2000 up to 2004 to initialize the ratings so that the rating variables have seen enough matches to be valid for use in regressions starting from 2005. The data from 2005 up to 2010 is used to train logistic regressions for multiple searches for the best hyperparameter settings and we estimate these hyperparameters by looking for which settings the predictions on data from 2011 up to 2012 is best. Then we combine the data from 2005 up to 2012 to train the models using the selected hyperparameters and compare the log loss of the models on data from 2013 up to 2014 to find the most suitable model.

5.2 Filtering

As our rating variables' accuracy is dependent on the number of match results obtained from a player, we can only make meaningful predictions if the players have already played some matches in our database. To filter the matches on reliability we need a measure indicating the reliability of the variables used before a match is played. We will do this in a similar manner as Sipko and Knottenbelt (2015). We multiply the number of matches both players played $N_A \cdot N_B$ and filter the 30 percent of the matches for which this value is the highest. In the rest of this paper when we train a logistic regression or predict the probability of a match outcome we have already filtered the match to only include the 30 percent of the most certain matches. It is important to realize that we do use all of the matches to calculate the rating variables.

5.3 Missing Point Data

Almost 11 percent of our matches miss data on point level. We still want to use these matches for the Barto method which takes the contest at point level as just ignoring these 11 percent of matches would be a waste of information. To do this we use the results of the sets (e.g. 6-4, 4-6, 6-4) to estimate the score. If there is at least one exact match in the results of the sets [7], we take the rounded average of the points played and points won of these matches. For the matches that contain no match on set level, we look for matches which have the same results in number of games won and lost and again take the rounded averages of the point scores. After this step less than 0.1 percent of the matches has no point data.

[7] We do not differentiate between the order the sets are played, so the set result 6-3, 6-4 matches with 6-4, 6-3.

5.4 Variables

All of the explanatory variables we will utilize will be expressed as a difference of the rating of player A minus the rating of player B. For example, the variable denoting the difference in Elo rating on the hard surface will be named and calculated as follows:

$$\text{EloHardDiff} := \text{Elo hard Player A} - \text{Elo hard Player B},$$

while the variable denoting the difference between the Barto ratings using as contests the sets is named and calculated as follows:

$$\text{BartoSetsDiff} := \text{Barto Sets Player A} - \text{Barto Sets Player B}.$$

In a similar manner we will name the other variables which you will encounter in the next sections. Since the values of the variables depends on which participating player is player A and which participating player is player B, we pick who was player A and player B at random for each match. This also decides whether a game resulted in a win or a loss.

Lastly, we will set the constant, β_0, of Equation (6) equal to 0 in all the logistic regression models. Otherwise the model will predict a different probability of winning if one is player A or player B and with no rating difference predict an outcome unequal to 0.5.

6 Hyperparameter Tuning

We have several hyperparameters, these are the parameters whose values are set before the training process begins. In our case these hyperparameters are relevant for updating the ratings of both Elo and Barto after each match. In the case of the Elo the hyperparameters are the parameters which affect the shape of K from Equation (3). In the Barto the hyperparameters denote the variances of the prior and the performance, and the scaling parameter β_p which are displayed in Figure 5.

We will estimate these hyperparameters by creating a logistic regression model which is relevant to the hyperparameters we want to optimize. We follow this up by running an exhaustive amount of initializations for the parameters to create the necessary rating variable repeatedly. For each initialization we train the logistic regression on the data from 2005 up to 2010 and then predict for 2011 and 2012 the match outcomes then we compare these values with the real values and calculate the log loss. The parameter initialization that minimizes this loss function is the optimized settings of the hyperparameters and will be used later to test the model. This manner of optimizing over an exhaustive number of hyperparameters is commonly called a grid search.

6.1 Elo Hyperparameters

The first hyperparameters we want to optimize are from the Elo method while using for K a function dependent on the number of matches and created by FiveThirtyEight. We will optimize over the power, p, of the denominator and the constant value, c, of the numerator. Below the function of K for which we derive the parameters is shown:

$$K_{538}(n_A) = \frac{c}{(5 + n_A)^p}. \tag{23}$$

The form of the logistic regression which we train for each set of hyperparameters is shown below:

$$y_i^* = \beta_1 \text{EloDiff}_i + \beta_2 \text{EloHardDiff}_i + \epsilon_i. \tag{24}$$

Searching in a grid search where the constant value varies from 20 up to 30 and the power from 0.02 up to 0.11 results in selecting the optimal value shown below.

Table 2: optimal hyperparameters Elo FiveThirtyEight

constant	**power**	**log loss**
21	0.03	0.5478

Since this results in a relatively flat function compared to the original formula of FiveThirtyEight, this begs the question how well an Elo system functions using a constant value for K. Using the same logistic regression, but now only optimizing over a new constant value, c this gives the following result (grid search from 17 to 22 for the constant).

Table 3: optimal hyperparameters Elo constant

constant	**log loss**
19.7	0.5478

Now one can see that the extra complexity introduced by FiveThirtyEight function for K compared to a constant value does not improve the predictive power of the model (the log losses are equal). Therefore in the next sections we will use the simpler model, using the constant value of 19.7 for K in Equation (3) to update the Elo ratings.

6.2 Barto Hyperparameters

We have proposed two levels at which to take the contest, sets and points. In both cases we fix β_p to equal $\frac{1}{400}$ as is also done in the Elo and Glicko method. This leaves two variables to optimize over, σ, denoting the standard deviance of the prior and β_d denoting the standard deviance of the day form. We will start with optimizing it with taking sets as the contest levels, with the regression below corresponding:

$$y_i^* = \beta_1 \text{BartoSetsDiff}_i + \beta_2 \text{BartoSetsHardDiff}_i + \epsilon_i. \tag{25}$$

We constructed this grid search with the σ ranging from 70 to 90 and we set β_d to range from 2 up to 30. The optimal hyperparameters selected are shown below.

Table 4: optimal hyperparameters Barto sets FiveThirtyEight

σ	β_d	**log loss**
80	2	0.5470

Even though the estimated β_d value is 2 and therefore on the edge of the grid we searched over. We still consider this optimization method converged since the values of the log loss for β_d are not changing in any significant matter from 12 through 2.

We can see here a slight hint that Barto might be able to outperform Elo since the optimized log loss is lower in the Barto method than in the Elo method.

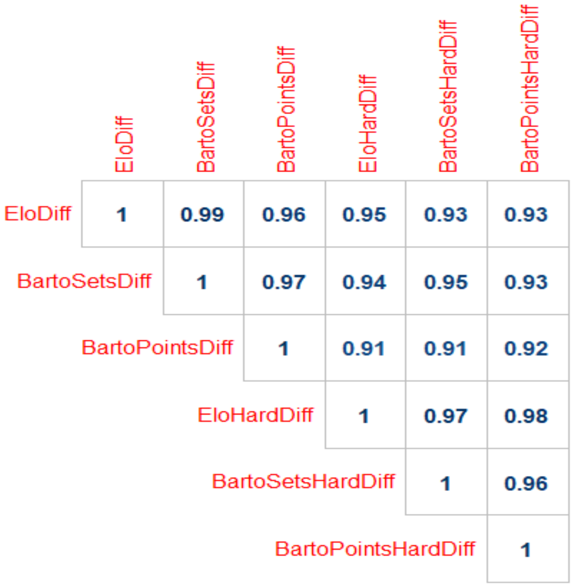
If we take the same steps for the Barto but when taking the points as the size of the contest, and apply the grid search on the interval of σ from 20 up to 24 and β_d from 40 up to 60 we obtain the following results.

Table 5: optimal hyperparameters Barto points FiveThirtyEight

σ	β_d	**log loss**
23.25	58	0.5458

Again the log loss is reduced, again hinting that the Barto system might be able to outperform the Elo method. Now that we have estimated the hyperparameters we put this theory to the test in the next section.

Figure 8: Correlation matrix



	EloDiff	BartoSetsDiff	BartoPointsDiff	EloHardDiff	BartoSetsHardDiff	BartoPointsHardDiff
EloDiff	1	0.99	0.96	0.95	0.93	0.93
BartoSetsDiff		1	0.97	0.94	0.95	0.93
BartoPointsDiff			1	0.91	0.91	0.92
EloHardDiff				1	0.97	0.98
BartoSetsHardDiff					1	0.96
BartoPointsHardDiff						1

7 Results

In Figure 8 the correlation of the variables is shown. Both the variables measuring the general skill and the variables measuring the hard skill show very high correlations with themselves, meaning that the models generate similar estimates of a player's strength.

Before the logistic regressions are performed the standard deviation of the variables has been normalized, i.e. set equal to 1. This is done so that the coefficients of the logistic regressions are comparable, higher weights mean more importance is given to one variable over another.

In Table 6 the statistics of the logistic regressions on data from 2005 up to 2012 are presented. The standard errors of the coefficients are relatively low, resulting in the effect of the variables being significantly significantly different from zero at a 1 percent level for all three models. Also in all three models the coefficient relating to the rating taken over all three matches is higher than the rating taken only on hard surface matches, meaning that a general tennis strength of a player explains a bigger part of the wins and losses than his strength on only the hard surface.

The log likelihood for the models from highest to lowest are Barto using points followed by Barto using sets and after that the Elo model. This indicates that the Barto models are better able to explain the relation between winning a match and the used rating variable than the Elo model is able to.

Table 6: Logistic regressions on data from 2005 up to 2012

	Dependent variable:		
	y		
	(Elo)	(BartoSets)	(BartoPoints)
EloDiff	0.641***		
	(0.115)		
EloHardDiff	0.553***		
	(0.115)		
BartoSetsDiff		0.797***	
		(0.114)	
BartoSetsHardDiff		0.408***	
		(0.113)	
BartoPointsDiff			0.719***
			(0.095)
BartoPointsHardDiff			0.511***
			(0.095)
Observations	3,949	3,949	3,949
Log Likelihood	-2,275.452	-2,270.594	-2,257.206
Akaike Inf. Crit.	4,554.904	4,545.189	4,518.411
Note:		*p<0.1; **p<0.05; ***p<0.01	

For the three models presented in Table 6 we predict the outcomes for the matches played in 2013 and 2014 which totals 1288 matches. Then we compute the log losses using these predictions and the actual results. The results are shown in Table 7. Remember that lower log losses indicate better predictions. We see that again the Barto models outperform the Elo model, meaning that the predictive power of the Barto models is stronger than the Elo model. The Barto models have comparative predictive power.

We find that a model which mixes the probabilities of the Barto model as shown below

$$\text{Barto mixed} = 0.5 \text{ Barto sets} + 0.5 \text{ Barto points} \tag{26}$$

improves the predictive power of the model. This implies that even though the models Barto sets and Barto points have the same predictive power they capture the strength of the player in a sufficiently distinct manner that mixing both models better represents the strength of a player.

Table 7: Log losses of predictions on data from 2013 up to 2014

model	Elo	Barto sets	Barto points	Barto mixed
log loss	0.5495	0.5461	0.5460	0.5443

To get a better sense of the significance of the differences we compare the frequency that the log loss of a model's predictions is lower than the other models. We computed 10,000 bootstrap samples per comparison, the results are shown in Table 8. We can see that only the Barto mixed model predicts significantly better than the Elo model at a 5% significance level, with the other Barto models predicting significantly better than the Elo model at a 10% significance level. The Barto sets model and the Barto points model seem to out-predict each other at approximately the same rate. The Barto mixed model can not be said to predict significantly better than any of the other Barto models.

8 Conclusion

We have extended the literature of rating based methods such as the Elo (1978) method and the method of Glickman (1999). While Elo and Glicko only look at the binary result of a match, we have created a novel method coined Barto, which is able to use scores granted that the score represents contests which has a winner in each contest such as sets or games. Examples of sports which have such a structures are tennis, volleyball, darts and

Table 8: probability lowest log loss (10,000 bootstrap samples)

	Elo	**Barto sets**	**Barto points**	**Barto mixed**
Elo		0.060	0.057	0.010
Barto sets	0.940		0.493	0.139
Barto points	0.943	0.507		0.153
Barto mixed	0.990	0.861	0.847	

The table is read such that 0.060 signifies that Elo has a lower log loss than Barto sets with a probability of 0.060.

snooker which all have game and/or set structures. This method is not applicable to sports which have a time limit in which points can be scored such as football and hockey.

We have applied this method on tennis matches and compared it with the state-of-the-art model, the Elo model. We found that it gives similar estimates of the skill of a player as the Elo model, yet the estimates are sufficiently different to outperform the Elo method in both prediction and in explanation. We found significant improvements in prediction, making this model the new state-of-the-art model in tennis prediction.

In practice this method can be used for both prediction and ranking. Both of which have no shortage in demand in the sports world. The drawback to the Barto compared to the Elo is that it is a more complicated method to implement while its advantage is that it more accurately estimates a player's strength. Furthermore in the case of tennis the Barto is able to split a player's strength in both a serve strength and a return strength.

Future work can be found in both applying this model to other sports and extending the current model. Other sports in which this might lead to state-of-the-art ranking and/or prediction are volleyball, darts or snooker. A possible further extension of the model can be found in working with a volatile standard deviation of the rating, a rating deviation, such as Glickman has extended the Elo model. He used Bayesian foundations to reduce a player's rating deviation after each match (since a more accurate rating can be estimated when one has more information) and after each period a player's rating deviation increased (to increase uncertainty after inactivity of a player). We have already formulated the manner in which to update ratings which have unequal rating deviations in Section 3.3. All that remains is finding a method to update the rating deviation after a match and over time.

References

Brent, R. P. (2013). *Algorithms for minimization without derivatives.* Courier Corporation.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 215–242.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics*, *7*, 1–26.

Elo, A. E. (1978). *The rating of chessplayers, past and present.* Arco Pub.

Glickman, M. E. (1999). Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *48*(3), 377–394.

Klaassen, F. J., & 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), 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), 3820–3827.

Kovalchik, S. A. (2016). Searching for the goat of tennis win prediction. *Journal of Quantitative Analysis in Sports*, *12*(3), 127–138.

Romberg, W. (1955). Vereinfachte numerische integration. *Norske Vid. Selsk. Forh.*, *28*, 30–36.

Sipko, M., & Knottenbelt, W. (2015). *Machine learning for the prediction of professional tennis matches* (Unpublished doctoral dissertation). Imperial College London.