A bayesian rating method to predict (professional) tennis matches

Introduction

Common Opponent Models

This method’s goal is to use the most recent matches (e.g. 12 months) and calculate the performance metrics such as “proportion of points won on serve” to predict the probability that player i wins versus player j. This could cause problems, if both players played versus opponents of different strengths. Therefore a subset of the matches is used, only opponents where both players played versus each other. Because the top players often play versus each other this method is applicable in tennis. (Knottenbelt et al 2012).

Bayesian Rating Systems:

Elo

Arpad Elo created a rating system for the USCF (United States Chess Federation) based on the normal distribution. The assumption is that each game is distributed in the following manner: pi ~ N(pi; si, β2). This results in an expectation for player 1 of:

E(w1) = P(p1 > p2 | s1, s2) =

To update the ratings of the players the following rule is used:

newRatingPlayer1 = oldRatingPlayer1 + k \* (result - E(w1))  
newRatingPlayer2 = oldRatingPlayer2 + k \* (result – E(w2))

where the k-factor decides the relevance of one game played.

Glicko  
  
Mark Glickman makes use of an unique standard deviation for each player, making the result dependent on both the players skill levels and the players standard deviations. A player’s standard deviation increases when he doesn’t play any games and decreases for each game played.

##Story about why logloss

.

The Tennis Dataset

The data is taken from [www.github.com/JeffSackmann/tennis\_atp](http://www.github.com/JeffSackmann/tennis_atp), which contains ATP matches starting from January 1968 up to 11 september 2017. Only the matches from January 2000 up to December 2016 are taken. In total there are 53508 ATP matches in this period. This data contains all sorts of match statistics. But does not contain the day the match is played, nor betting odds. Therefore I join the data at [www.tennis-data.co.uk](http://www.tennis-data.co.uk) to add these variables. After this step only 46780 matches are left, but these matches are double checked for accuracy. Since matches on carpet have not been played since 2009, these matches are also removed resulting in a database with 45104 matches.

|  |  |  |  |
| --- | --- | --- | --- |
| Surface | Clay | Grass | Hard |
| matches | 15347 | 5189 | 24568 |

Bibliography:  
Knottenbelt, W. J., D. Spanias, and A. M. Madurska. 2012. “A Common-

Opponent Stochastic Model for Predicting the Outcome of

Professional Tennis Matches.” *Computers & Mathematics with*

*Applications* 64(12):3820–3827.

Approximating Scores Missing Games

47740 have point data and the other 5768 miss this data. This data is necessary for the Barto rating system. Therefore for the missing data, the scores are approximated in the following manner:  
 If there are other matches with the exact same result in set scores (while not caring about the order in which the sets are played, so 6-4, 4-6, 6-4 is the same as 4-6, 6-4, 6-4). Then the average number of points played in those matches will be taken for the feature w\_svpt and l\_svpt. Then these average number of points are multiplied by the average percent of points won on respectively serve and return and rounded to the nearest integer to fill the features w\_svpt\_won and l\_svpt\_won. If there is no other match with the same result in set scores, the same steps are taken, using matches with the exact same number of games won during the whole match. If there are none of these matches the point data remains blank for the game.

To prevent data/information? leakage. The point data filled up to the validation period, only uses results up to the train model dataset and the test data set uses results up to the validation period.

Missing values before imputation: 5765 out of 53062  
 Missing values after imputation: 42 out of 53062

Discussion: better ways to impute missing values, or search for the real values, but probably too little to care about.

Setup hyperparameters:

We optimize the uncertainty parameters by looking at the original model of choosing ratings by Five-Thirty Eight, running a regression on rating and surface rating and then optimize over q, minimizing the logloss.

Parameters five thirty eight according to comparison model:

Because we want to find a robust solution, we will take the uncertainty values q = 21 : 40 to optimize the hyperparameters in a greedy manner, one by one. By a gridsearch.

Parameters to optimize:

1. Elo, five thirty eight and constant
2. Barto
3. Fatigue using the covariate barto, so that fatigue will not measure skill too much