Tennis Match Prediction using Machine Leanring

Ajay Krishna Amudan
ajkrishn@stanford.edu

1. Abstract

Tennis is a truly international sport with players coming from more than 100 countries. The sport is played across more than 50 countries at the elite level - ATP and more than 100 countries at professional level - Challengers, Futures and qualifying. There are a number of playing styles including single vs double handed backhand, left vs right handed, continental grip vs wester grip, high topspin vs flat shots, grinding-baseline vs rush-to-the-net and many more.

Tennis betting is a lucrative market because it is quite hard to predictive the outcome of matches as there are so many factors that affect it. In this project we attempt to predict the outcome of tennis matches before the start of the tennis match using statistics that we have available at that point.

2. Introduction

Tennis matches can't be predicted perfectly. In fact sports betting can never aim to achieve 100% accuracy or even close to it- the whole purpose of playing sports is to see who wins this unpredictable battle of mind, body and soul. But the good part of this is that even a slight improvement over the existing algorithms and techniques can mean a huge profit.

We attempt to beat benchmark of getting more predictions right than the simple technique of always predicting the higher ranked player as the winner by using machine learning and engineering the right features and using the right models.

3. Related Work

Two CS229 projects have been in the area of tennis match prediction in 2017 – one in the area of in-match prediction and one in the area of pre-match prediction which is where we will concentrate in this report. There are also a few other research papers published in the area of tennis betting which I detailed in the reference section.

Having said that tennis match prediction overall seems to have very little publicly available research. To the best of my knowledge, even the two CS229 projects do not have any publicly available implementation of their approach and many details in the report which might be obvious if the code is read is not very clear purely from the report.

A major reason to continue work in this area is because the dataset available is growing continuously. Jeff Sackman who maintains a Github repository that seems to be the only publicly available reliable dataset for this problem has been continuously updating it with new statistics every year and fixing a number of issues that existing previously.

4. Dataset and Features

The Jeff Sackman Dataset contains multiple sets of files. We use the Sackman_tennis_atp-master project which contains match level statistics from the year 1968. Since more extensive match level statistics are only available from the year 1997, we concentrate on the ATP match level files from 1997 to 2019. For challengers, from the year 2010, we have extensive statistics. We choose to train the years from 1997 to 2018 including challenges from 2010 to 2019 and predict on the ATP matches that happened in 2019.

The dataset was initially cleaned up by fixing a number of values and removing a number of rows that were possibly wrong data. All rows of data which had any missing column when a value was expected was also ignored. A number of sanity tests were performed to judge the quality of data as well.

We chose to represent most features in the form feature_player_1 – feature_player_2 – that is as a difference of two features for the two players. We did this to reduce the number of features and also because the accuracy did not go down while experimentation. The following are the set of features chosen. Each feature was chosen carefully after identifying that there was a meaningful increase in accuracy after adding it. A number of features were rejected as they did not improve the accuracy or decreased it.

Tournament Form – Historical win/loss for the player in the specific tournament

Match based Form – Number of matches won/lost in the last 5/10/15/25 matches

Time based Form – Number of matches won/lost in the last 1 month

Surface Level Overall Form – Number of matches won/lost overall in the particular surface

Surface Level Match based Form – Number of matches won/lost on the surface in the last 5/10/15/25 matches on the surface

Overall performance – Number of matches won/lost overall

Overall head to head – Number of matches won/lost against each other

Component Opponents Win/Loss – Number of matches won/lost against common opponents both players have played against

Surface level head to head – Number of matches won/lost against each other in the same surface

Player attributes – Height, Left vs Right Hand, Rank, Ranking Points, Player Seed

Overall Historic Average and cumulative Match Level Statistics – First serve percentage, Aces, Double Faults, First Serve won, Second Serve Won, Break points faced, break points saved, Service Games Played

Each match in both training and test dataset is represented by two points and in the training set each pair of such points consist of one with label 0 and one with label 1. The label 0 indicates the player who is represented as player_1 in feature_player_1 lost the match and similarly label 1 is defined as the player_1 winning the match. We do this so that the algorithm can understand what features will lead to a win and which ones will lead to a loss.

In the test set, we predict the label of both pairs as described above and if the labels come out to be the opposite then they both agree on a single winner. If they have the same value, we choose that player for whom the probability of the sum of the two points predicting that player is maximum.

Secondly it is very important to notice that the raw dataset is also a time series in some manner – which means that the ordering of raw data matters. To remove this time dependence – we always make sure that the raw dataset is sorted when computing the training and test data set. All features we compute are only averaged up to the time that the match occurs Since this is the case we also ensure that we include challenger level matches to ensure that we have as much data as possible about newer players and about players which might have only met during challengers before the time we test them. This helps us remove some data skew that is inherently present because this is a time series dataset and younger players will have lesser training data.

5. Methods

After cleaning the datasets, I focused most of this project on feature selection. I selected feature which seem reasonable and add them to the current model iteratively and see how the performance changes. If it satisfactorily increases the accuracy in both training and test dataset, I chose to keep these features.

We run logistic regression whose probability of outcome being y is given by:

$$\mathcal{P}_{\boldsymbol{w}}(y = \pm 1|\boldsymbol{x}) \equiv \frac{1}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}},$$

Logistic Regression is the model that I used most heavily. I used the sklearn library and used liblinear model with L-2 regularization. This was done because the size of the dataset was not prohibitively high to not use liblinear mode – the training dataset was of size 120,000 rows approximately and 51 columns and the test dataset was of size 1000-3000 depending on exactly which months and year I tested for.

I also tried running random forests using the sklearn.ensemble library. But because it took longer to run, I did not run it on all the different features sets I was testing. The random forest implementation averages the result from each decision tree in the random forest. Each and every decision tree is internally generated to maximie the entropy given by

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

C ={yes,no}

Finally, I also tried using SVM – from the sklearn library as well. But the amount of time it took to train was extremely long and hence I only ran it for a few choices of features and training dataset.

6. Experiments, Results and Discussions

Code - https://github.com/ajkrish95/cs229-tennis-prediction

Dataset - https://github.com/JeffSackmann/tennis_atp

Logistic Regression:

Logistic regression gave an accuracy of 71% on test data and 72% on training data.

The benchmark accuracy to beat was 65.5% which is the accuracy one gets if the higher ranked player is always predicted to win. This 65% also seems to be quite constant year over year in terms of the total accuracy when the higher ranked player is predicted to win.

The hardest matches to predict correctly seem to fall under 3 categories – players who have not played many matches before, two players who meet who have never faced each other before and lower rank players beating the higher ranked players.

The second problem was solved reasonably well by adding common opponent head to head as a feature – which is basically the win/loss record for the two players against common opponents

The first problem was solved reasonably well by adding challenger series and qualifying matches data as well which helped in getting more data on each player since all ATP players necessarily play some challenges before turning pro.

The third problem was the hardest problem to solve. I attempted to solve by analyzing the matches that were predicting wrongly and adding features accordingly. For example, there are a number of clay court specialists who play very well on clay but have quite a bad record outside it. This was a major reason why I added surface level statistics. There are also injuries which happen which limit the number of matches played. The feature to see how many matches were played in the last month was introduced with that mind. There are often players who historically play well at a certain tournament or at a certain time of the year. This is why tournament level statistics were introduced. Finally, both the average value of match level statistics as well the total sum was added since often times experience needs to be given its due.

Features like height, rank and ranking points were included separately without taking the difference because it was seen that taking the difference hides behind details which are very important. For example, the difference between the rank 1 and 6 is not that same as the difference between the rank 200 and 206 – 200 and 206 ranked players are very close in skill level.

One of the other clear observations was that increasing the amount of clean data clearly improved the algorithm. This tells us that as we get more and more relevant contemporary clean data – we should keep attempting the problem because the most likely outcome is an increase in accuracy.

Finally the training and test accuracy remained almost completely constant when I changed the year from 2019 to 2018 and 2017 – I did not want to go any earlier because then we start reintroducing the skew for younger players because of omission of challengers series level statistics

SVM and Random Forests:

SVM and random forests gave me an accuracy of 65% and 68% respectively. Since I was heavily focused on optimizing my feature set, I did not try to derive many insights on these numbers.

7. Conclusion and Future Work

Logistic regression seems to work well to predict the outcome of tennis matches. For this specific problem, the key seems to lie in selecting and carefully curating the right features. Since that is the case other models – including SVM and Random Forests seem to improve whenever logistic regression improves on selecting a different set of features by adding or removing features or modifying existing ones.

I think future work should mostly lie in analyzing why each match that was predicted wrongly was predicted that way and try to understand which features are missing. I think for this problem the question is what set of features are most relevant and cover cases where lower ranked player upsets a higher ranked player or a bad head to head match up is reversed. This problem cannot be solved in totality – sometimes completely irrational results do happen. But other times it is simply a matter of understanding some hidden connection that was missed – and this is where I believe most future work should lie.

I am very excited to continue this project over the next few months – I strongly believe that I can start betting on tennis very soon once I develop a bit more confidence algorithm in this algorithm – which I believe is very possible with some more model testing and feature addition and removals.

8. Contributions

I was the only one who worked on this project. Since I could not find any previous implementations for any project that worked on the Jeff Sackman data, a significant portion of the time I took was in understanding and cleaning the data. I plan to open source my code once I clean it up more and make it more usable and friendly to people who want to start with some existing code.

9. References

1.http://cs229.stanford.edu/proj2017/final-reports/5242116.pdf

2.http://cs229.stanford.edu/proj2017/final-reports/5243744.pdf

3.https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf

4. T. Barnett and S. R. Clarke. Combining player statistics to predict outcomes of tennis matches. IMA Journal of Management Mathematics, 16:113120, 2005.

5.https://github.com/JeffSackmann/tennis%20atp%20,%20Jeff%20Sackmann

6.http://www.tennis-data.co.uk/alldata.php

7. http://cs229.stanford.edu/notes/cs229-notes3.pdf , Stanford University, CS229 Lecture Notes, Andrew Ng

8. A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap. Tennis Winner Prediction based on Time-Series History with Neural Modeling.IMECS 2009: International Multi-Conference of Engineers and Computer Scientists, Vols I and II, I:127132, 2009.

9. Agnieszka M. Madurska. A set-by-set analysis method for predicting the outcome of professional singles tennis matches. MEng computing- final year project, Imperial College London, amm208@doc.ic.ac.uk, June 2012.

10. Jeff Sackmann - https://github.com/JeffSackmann

11. et al. Jeff Sackmann. The tennis abstract match charting project, 2017