Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2427441

Modelling Momentum and its Impact on Tennis Matches

Summary

Many tennis players could feel “momentum” while playing. It is like a force that affects the performance of the player and the “flow” of the game, sometimes causing a “swing” in the flow. We sought to model momentum, flow, and swing through the analysis of Wimbledon data. We found that **momentum only had minimal effect on the flow of the game and cannot be used to accurately predict flow.**

Firstly, we filled in the missing data using a standard BP neural network.

Secondly, two models are applied to model the momentum: the **Elo Rating model** and **LSTM Recurrent Neural Network**. Momentum is modeled as the probability of winning the next point, calculated with consideration of different factors such as the strike speed and elapsed time. However, momentum did poorly in predicting the outcome of the next point.

Thirdly, the flow of the game is characterized with a **linear weighting model** and **AHP weight evaluation**. Since the calculated flow had a **fractural characteristic**, the mean of each game and set is calculated and swings are identified when a player’s flow crosses the mean.

Fourthly, we performed **Chi-square Goodness of Fit Test** and **mean value comparison** on streak lengths and confirmed that momentum do have a slight impact on flow. But after modeling each factor’s importance in determining momentum using **dropout analysis** on the RNN, we were unable to identify any indicators for momentum other than the player serving.

Finally, a **Hidden Markov Model** was built to predict the flow and swings based on the momentum from Elo or RNN models. The model was able to roughly predict the flow and swings of the game based on probabilistic calculations of whether the player is serving, but the momentum data did not increase the accuracy of prediction.

Considering the poor prediction performance of momentum, the lack of useful indicators and the minimal difference between streak length and random data, we concluded that momentum may not have as much effect on the game as the players felt, and the effect may be mostly psychological or biased.

Keywords: sports prediction; deep learning; markov chain; statistic analysis

Table of Contents

1 Introduction

1.1 Background

In tennis, momentum characterizes the psychological edge a player obtains over their rival by consecutively securing several **points, games** or **sets** in a **match**. The successive wins can affect the other player's confidence, resulting in errors and lost chances. Achieving momentum is vital as it bestows the player with a feeling of dominance, compelling the opponent to attempt to reclaim the upper hand. Possessing momentum allows a player to compete more boldly and without restraint, encouraging a more assertive and self-assured manner of playing (Teo, 2023). In the following parts of this paper, we will develop several models to quantify momentum and examine to what extent it influences the tennis matches.

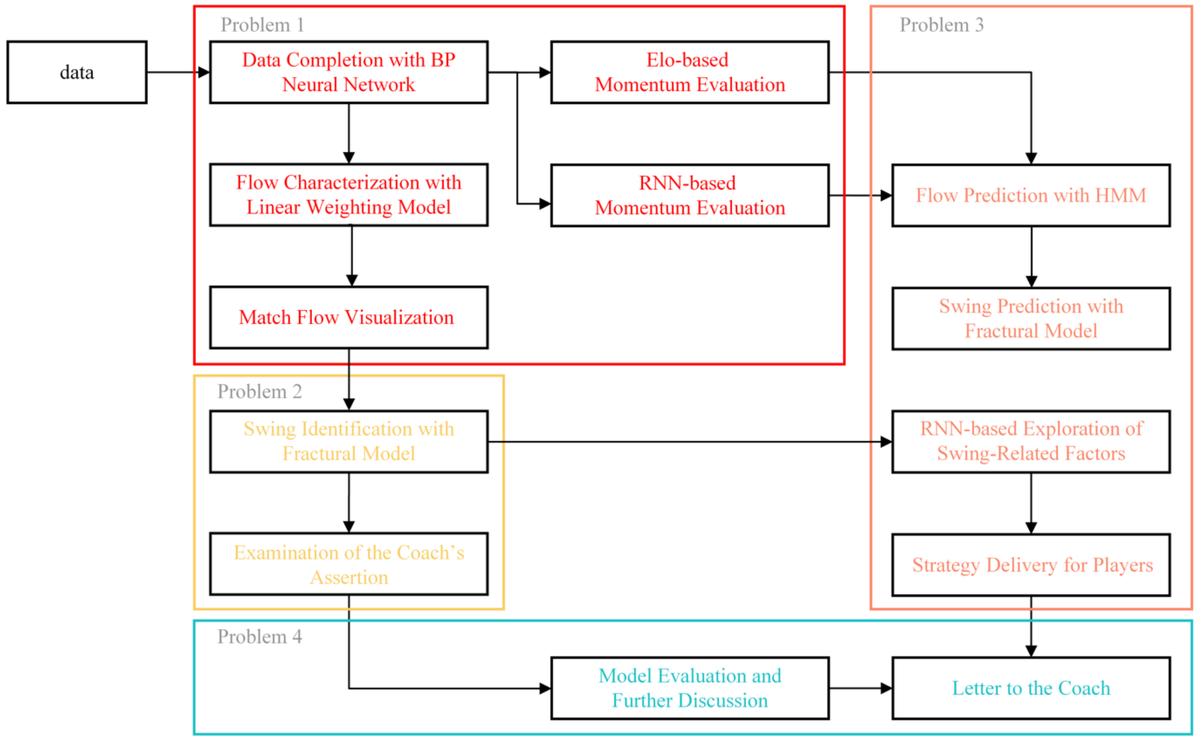
As for rules of the tennis match, USTA stated that "the aim of tennis is to win enough **points** to win a **game**, enough **games** to win a **set**, and enough **sets** to win a **match**" (Tennis scoring). Words that specifically describe these hierarchies will be **bolded** in our thesis to avoid ambiguity. Unbolded words may mean any and all of these hierarchies. The primary notations used in this paper are listed separately in different sections.

1.2 Restatement of the Problem

Given the background information, we need to solve the following tasks:

1. How can we quantitatively evaluate the flow of play in tennis matches and the "momentum" felt by players? How can we visually depict the flow with the mathematical models? How can the models incorporate the server's higher probability of winning **points/games** in tennis?
2. How do we identify the "swings" on the flow of the match? How can we assess the tennis coach's assertion that "momentum" plays no role in the match and that swings are merely random? Whether or not this assertion holds?
3. Are there indicators that signals an impending swing from one player to the other? If so, what factors seem most correlated? Considering previous instances of "momentum" swings, how do we advise players entering new matches against different opponents?
4. How accurately do our models forecast the swings in a match? Should the model occasionally underperform, can we pinpoint any elements that may need to be integrated into future iterations? To what extent can our model be generalized to other matches, tournaments, court surfaces and even to other sports such as table tennis?

1.3 Our Work



2 Assumptions and Justifications

1. Flow only describes the outcome of the game. In other words, it it only determined by the points won in each **set, game,** and **point**.
2. Momentum can be used to predict the flow of the game. The value of momentum can be translated to winning probability of a certain **match, set, game,** or **point**.
3. Swings may happen in different levels of the game. For example, a swing can happen across **points** within a **game**, even if the player is leading in the current **set**.

3 Flow Characterization and Momentum Evaluation (Problem 1)

3.1 Data Completion with BP Neural Network

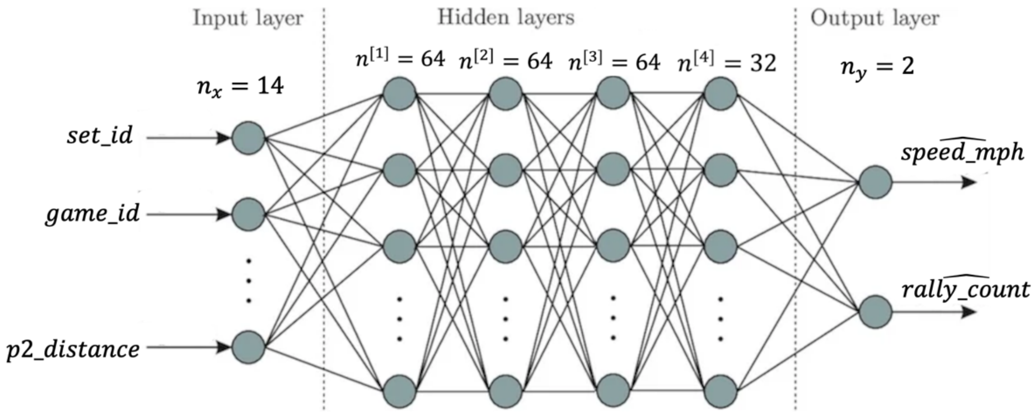
Notations

Symbol	Description
n_x	Number of inputs
n_y	Number of outputs
n^i	Number of nodes in layer i

On initial assessment of the data, we noticed that *rally_count* and *speed_mph* has missing entries. There are also several incorrect time data, which were adjusted by hand. We constructed a simple BP neural network to fill in the missing data using existing data. The following columns were extracted as input to the network:

set_no	game_no	point_no	p1_points_won
p2_points_won	p1_ace	p2_ace	p1_winner
p2_winner	winner_shot_type	p1_net_pt	p2_net_pt
p1_distance_run	p2_distance_run		

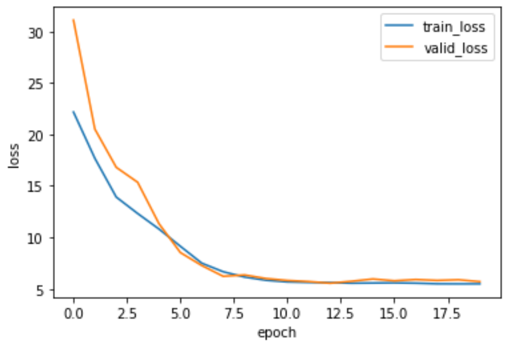
The structure of the network is shown below.



The hyperparameters of the network are as follows.

Hyperparameters	Value
Learning rate	0.001
# of epochs	20
Batch size	4
Learning rate decay	None
Weight decay	None

We trained the network for 20 epochs, the loss converged to about 5 at epoch 10. Although

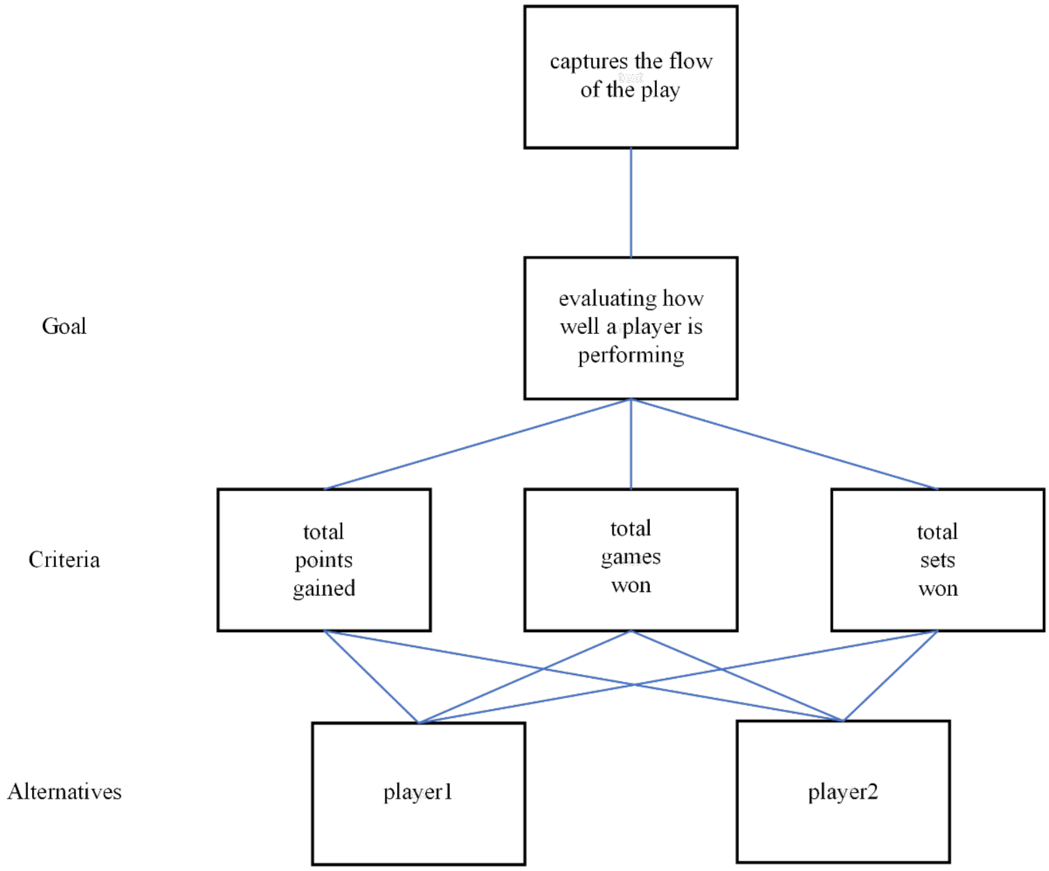


there was no weight decay, the trend of *valid_loss* showed little sign of overfitting. We hypothesize that the small size of the network was sufficient to avoid overfitting. However, there are still much room for the optimization of hyperparameters.

3.2 Flow Characterization with Linear Weighting Model

To capture the flow of the match, we constructed a simple linear weighting model that calculates a score for each player using the outcome of each **point, game** and **set.**

To obtain the weights of the model, we developed a flowchart of the analytical hierarchy process.



$$\begin{array}{cc} & \begin{array}{ccc} pts & gms & sts \end{array} \\ \begin{array}{c} pts \\ gms \\ sts \end{array} & \begin{pmatrix} 1 & 1/5 & 1/10 \\ 5 & 1 & 1/5 \\ 10 & 5 & 1 \end{pmatrix} \end{array}$$

By following evaluation matrices of the criteria, we can calculate out that the row weights.

$$W_1 = 0.05477615$$
$$W_2 = 0.20179707$$
$$W_3 = 0.74342677$$

The weights are then normalized to set W_1 to 1, they are as follows.

$$W_1 = 1$$
$$W_2 = 3.6840315$$
$$W_3 = 13.57208808$$

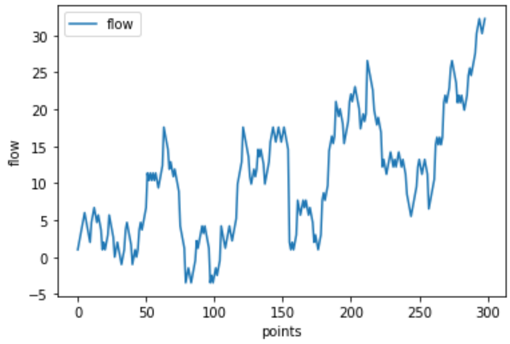
The score of each player, D_{ij} and D_{ji}, are calculated with the following equation.

$$D_{ij} = W_1 \cdot points\_won_i + W_2 \cdot games\_won_i + W_3 \cdot sets\_won_i$$
$$D_{ji} = W_1 \cdot points\_won_j + W_2 \cdot games\_won_j + W_3 \cdot sets\_won_i$$

The flow of the game is the difference between the players' scores. A positive value indicates that the situation favors player 1, vice versa.

$$F = D_{ij} - D_{ji}$$

The flow diagram of the first match is as follows.



3.3 Momentum Evaluation

3.3.1 Elo Prediction Model

Notations

Symbol	Description
$m_{ij}(t)$	The momentum of player i in a match against j *at time* t
$\widehat{P}_{ij}(t)$	*The probability that the next point will be won by* player i in a match against j *at time* t
K_{ij}	A coefficient that reflects time and distance run by a player
s_{ij}	A coefficient that reflects whether player i *is the server*
α_1	A coefficient that reflects time elapsed
α_2	A coefficient that reflects total distance run by a player

To evaluate a player's momentum at a given moment, we modified a model built by Elo (Elo, 1978). The Elo model was originally designed to evaluate the performance of chess player across matches. The original formula of the Elo model is as follows.

$$\hat{P}_{ij}(t) = \left(1 + 10^{\frac{m_{ji}(t) - m_{ij}(t)}{20}}\right)^{-1}$$

$$m_{ij}(t+1) = \begin{cases} m_{ij}(t) + K_{ij} * s_{ij} * \left(W_i(t) - \hat{P}_{ij}(t)\right) & between\ points \\ \dfrac{1}{2} m_{ij}(t) & between\ games \end{cases}$$

$$K_{ij} = \left(K_{ij0}(1+\alpha_1)(1+\alpha_2)\right) \frac{1 + \alpha_2 e^{-\frac{distance_1}{dist}}}{1 + \alpha_1 e^{-\frac{\Delta Time}{\Delta Time}}}$$

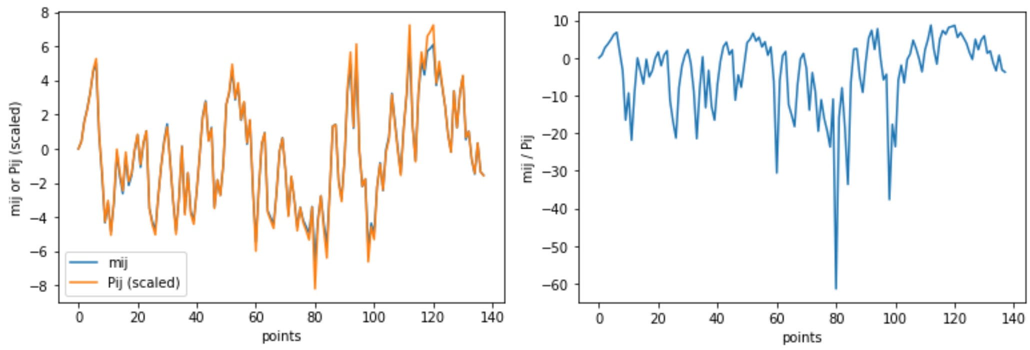
In our formula, t is the index of points. The initial value of the variables are as follows.

$$s_{ij} = \begin{cases} 0.8 & if\ W_i(t) = serve_i \\ 1.2 & if\ W_i(t) \neq serve_i \end{cases}$$
$$m_{ij}(0) = 0$$
$$K_{ij0} = 1$$
$$\alpha_1 = 10$$

The value $m_{ij}(t)$ is used to evaluate a player's momentum. We modified the K_{ij} factor it to incorporate various factors in a match. Intuitively, it represents the extent a player's momentum is affected by the outcome of the point.

First, we assumed a player's momentum changes quicker as $time\_elaped$ and $distance\_run$ increases, denoted by α_1. Second, a player's momentum may change drastically after a grueling point. Third, according to various research, whether a player is serving plays a big role in their chance of scoring, denoted by s_{ij}. When a player fails a serve or succeed in a return, s_{ij} is high, and vice versa. To simulate the change of momentum between each game, the value of $m_{ij}(t)$ is halved after the end of one game.

Since the Elo model was originally designed to evaluate chances of winning between different opponents, the value of $m_{ij}(t)$ and $\hat{P}_{ij}(t)$ turned out to be overlapping (but not identical) when continuously evaluating the same pair of players. The graph of $m_{ij}(t)$ and $\hat{P}_{ij}(t)$ (scaled) is as follows.



The accuracy of next-**point** prediction using the modified Elo model is as follows. We were unable to achieve high accuracy since there's limited time to tune the parameters. Further optimization may increase the accuracy.

# of games considered	prediction accuracy
6	0.4643
12	0.5455
300	0.5250
All	0.5181

3.3.2 Recurrent Neural Network (RNN)

Notations

Symbol	Description
x	Input vector
n_x	Number of inputs
n_h	Number of nodes in the hidden layers after input

n_y	Number of outputs
L	Number of LSTM layer

An alternative method to predict the momentum is with Deep Learning. Recurrent Neural Networks (RNNs) are capable of predictions of winning or losing states while taking into account the past **points**. We used the following data as the input to the RNN and the model was trained to predict the outcome of the next **point**.

Input:

set_no	*game_no*	*point_no*	*server*	*serve_no*	
point_victor	*p1_ace*	*p2_ace*	*p1_winner*	*p2_winner*	
winner_shot_type	*p1_unf_err*	*p2_unf_err*	*p1_net_pt*	*p2_net_pt*	*(t)*
p1_distance_run	*p2_distance_run*	*rally_count*	*speed_mph*	*serve_width*	
serve_depth	*return_depth*				

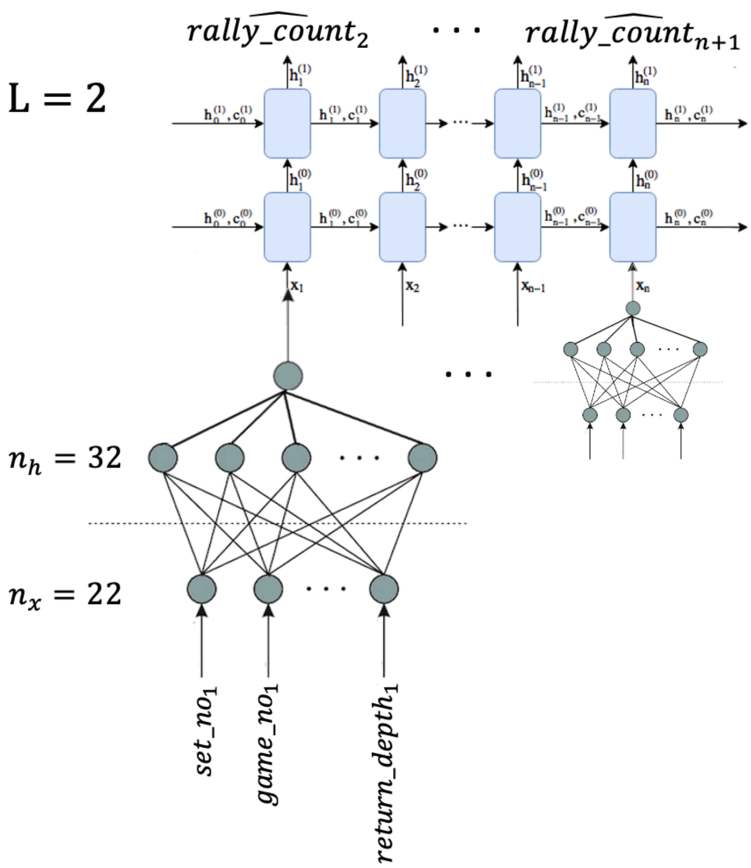
Output:

$point\_victor(t + 1)$

The inputs are first normalized with the following formula.

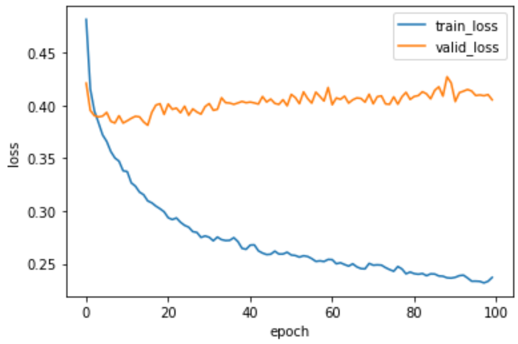
$$x_{norm} = \frac{x - \bar{x}}{\sigma(x)}$$

Then the input is run through the following RNN with Long Short Term Memory (LSTM) layers. The network treats the input with a 32-node hidden layer, then fed to the LSTM module of two layers.



Hyperparameters	Value
Learning rate	0.001
# of epochs	100
Batch size	16
Learning rate decay	0.99 per 100 steps
Weight decay	5e-4
Train/valid/test set size	5500/1000/752

The training loss converged over time, but the validation loss refused to converge after a few epochs. We suspect that overfitting is not to blame, since decreasing the model size or increasing weight decay showed little effect. It is possible that the current data was not sufficient to predict the outcome of the game beyond a certain accuracy since real life sports event is extremely complicated.



The accuracy of the prediction on the test set is as follows.

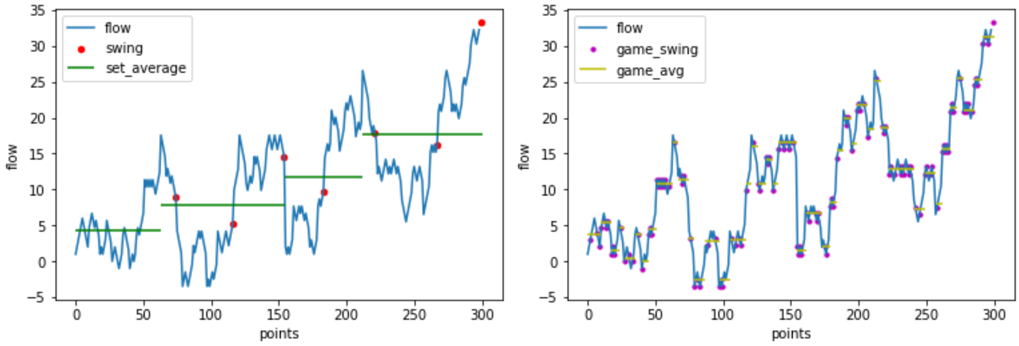
$$accuracy = 0.6015$$

4 Identification of the Swing in Play and Examination of the Coach's Assertion (Problem 2)

4.1 Identifying Swings with Fractural Model

Upon examining the characteristics of our flow model, we discovered that the flow had a fractural characteristic. This was appropriate because winning each **set, game,** and **point** was attributed a different weight with distinct magnitude. It is therefore appropriate to assume that each **match, set** and **game** has its corresponding swings.

To identify a **set-level** swing, the average flow of each **set** is first calculated. Then a swing is considered to have occurred when then flow crosses the average value. **Match-** or **game-**level swings are calculated likewise. The swings in **match** 1301 is as follows.

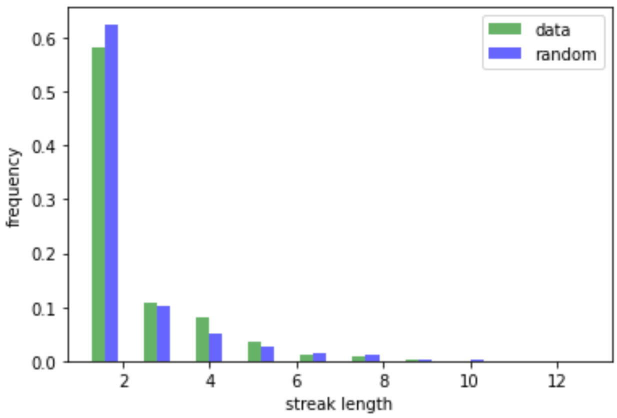


4.2 Testing for the Randomness of Swings with Statistic Analyses

4.2.1 Measuring the Distribution of Streak Length

We assume that, since swings happen when the flow of a player is interrupted, the occurrence of swings should be related to the length of winning or losing streaks of a player. We extrapolated the lengths of winning/losing streaks and performed statistic analyses.

The distribution of streak lengths compared to random data are as follows.



4.2.2 Chi-square Goodness of Fit Test

We first used the Chi-square Goodness of Fit Test to determine the relationship between real world data and random data. The null hypothesis is that the distribution of the real-world data follows a perfect random distribution. To validate this, a Chi-square goodness of fit test was performed. The formula of the chi-square value is as follows. k is the total number of bins, O_i is the observed frequency of data, and T_i is the expected frequency.

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - T_i)^2}{T_i} = 138.1735$$

Looking at the table for $k = 14$ and $\alpha = 0.001$, the critical value is 34.528. Since $138.1735 > 34.528$, the null hypothesis is proven false, we are 99.9% sure that the real-world data is not random.

4.2.3 Mean Value Analysis

The average streak length is as follows. The average streak length of real-world data is 6.5% longer than random data. This may imply that streaks are less likely to be broken due to momentum.

$$\overline{l_{data}} = 2.1411$$
$$\overline{l_{random}} = 2.0113$$

4.3 Evaluating the Importance of Different Factors

4.3.1 RNN Dropout Analysis

After training the RNN, we assessed the importance of each factor by removing factors from the RNN input and assessing the model accuracy. Each input set was trained to 100 epochs with the same hyperparameters. The findings are as follows.

Parameters Dropped	**Accuracy Loss** l_i	**Relative Importance** r_i
none	0.6015	NA
set_no,game_no,point_no	0.0275	0.5158
server	0.0627	1.0000
server_no	-0.0006	0.1293
point_victor	0.0285	0.5296
p1_ace, p2_ace, p1_winner, p2_winner	0.0178	0.3824
p1_unf_err, p2_unf_err	0.0213	0.4305
p1_net_pt, p2_net_pt	-0.01	0.0000
p1_distance_run, p2_distance_run	0.0031	0.1802
rally_count	0.0053	0.2105
speed_mph	0.0034	0.1843
serve_width, serve_depth, return_depth	0.0094	0.2669

From the finding, we can see that whether the player is serving plays a great role in defining momentum. Consecutive serves within a game may greatly boost momentum.

The current progress and outcome of the **game**, indicated by set_no, $game\_no$, $point\_no$, and $point\_victor$ also plays a great role. This does coincide with the traditional perception of momentum: a continuation of outcome.

Other factors are not as important in defining momentum.

5 Swing Prediction Model and Strategy Delivery for Players (Problem 3)

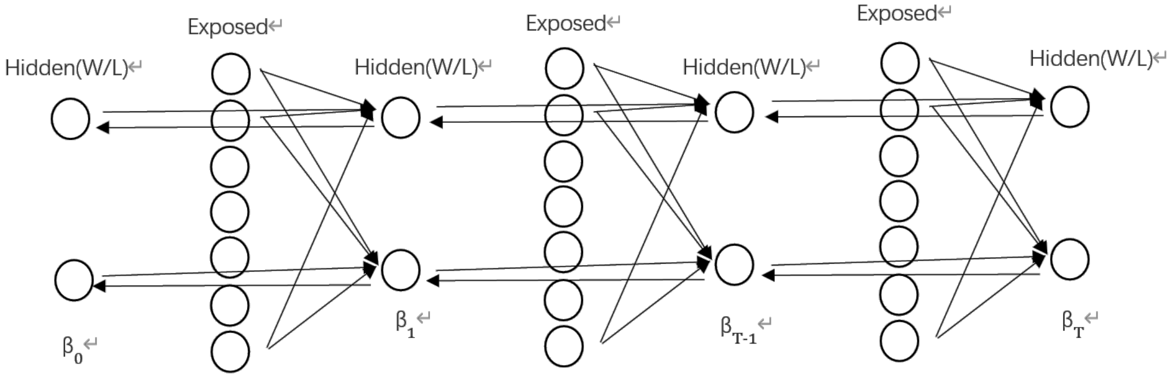
Notations

Symbol	**Description**
A=$[a_{ij}]$	Exposed State: Probability from state i to state j
B=$[b_{ij}]$	From Exposed to Hidden: Probability from state I (Ex) to state j (Hi)

π	Initial probability distribution
λ	Current probability background
β_t	Probability of each item in the t-th round
X	*Refers to various events in general*

5.1 Building the Hidden Markov Chain Model (HMM)

We used the Hidden Markov Chain Model (HMM) to extend single **point** predictions to multiple **points**, thereby predicting the flow and swings of the game. The diagram of HMM is as follows.



First, the probabilities of latter nodes must not interfere with the former nodes, as denoted by the following equation.

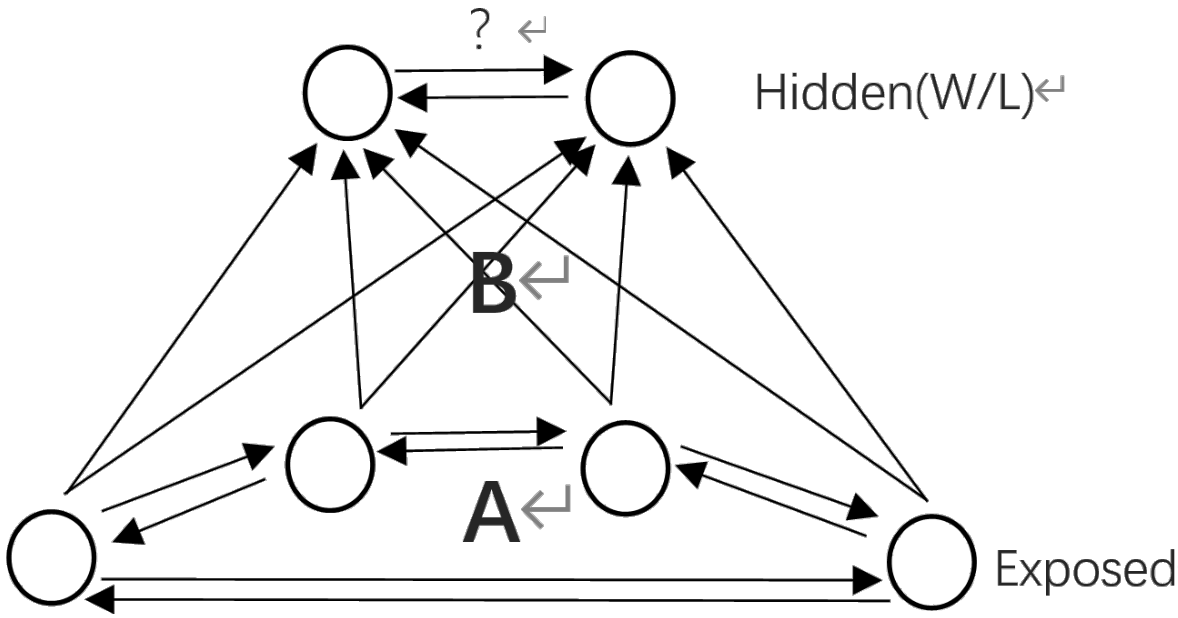
$$P\left(X_{n+1=j} \middle| X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0\right) = P(X_{n+1} = j | X_n = i) = P_{ij}$$

λ represents the various data of the entire process, that is, "in the current situation". π represents the probability of various states appearing at the beginning, and we believe that these probabilities are all the same.

$$\lambda = (\pi, A, B), X = (x_1, x_2, \dots, x_T) \Rightarrow P(X|\lambda)$$

$$P(X|\lambda) = \sum_{Z} P(X, Z|\lambda)$$

HMM has two layers, the hidden layer and the exposed layer. The winning probability is stored in the hidden layer and the player serving, and player fatigue is stored in the exposed layer, since player fatigue is apparent but winning probability is hidden.



We can get weight matrix A and B. A is the probability of transitioning from one exposed state to another. B is the probability of transitioning from one exposed state to one hidden state. Here, B is the predicted winning rate, obtained from Elo or RNN model.

$$A=\begin{bmatrix} a_{11} & \cdots \\ \vdots & a_{ij} \end{bmatrix},\ B=\begin{bmatrix} b_{11} & \cdots \\ \vdots & b_{ij} \end{bmatrix}$$

If we want to let player i win, we need to know the possibility of following the previous state and the relationship between victory and defeat from the beginning to the present.

For $\beta_t(i)$, if $\mathrm{P}(\beta_{start}(i)|\beta_t(i)=1,\lambda)>\mathrm{P}(\beta_{start}(i)|\beta_t(i)=0,\lambda)$ then player i is forecasted to win.

The derivation of the final probability formula is as follows. Z represents the various states of the players in the previous rounds.

$$\beta_t(i)=\mathrm{P}(x_T,x_{T-1},\dots,x_{t+1}|z_t=q_i,\lambda)$$

$$=\sum_{\mathrm{j=1}}^{\mathrm{P}} P\left(x_T,x_{T-1},\dots,x_{t+1},z_{t+1}=q_j \middle| z_t=q_i,\lambda\right)$$

$$=\sum_{j=1}^{N} \mathrm{P}\left(x_T,x_{T-1},\dots,x_{t+1} \middle| z_{t+1}=q_j,z_t=q_i,\lambda\right) P\left(z_{t+1}=q_j \middle| z_t=q_i,\lambda\right)$$

$$=\sum_{\mathrm{j=1}}^{\mathrm{N}} a_{ij}\mathrm{P}\left(x_T,x_{T-1},\dots,x_{t+1} \middle| z_{t+1}=q_j,z_t=q_i,\lambda\right)$$

$$=\sum_{\mathrm{j=1}}^{\mathrm{N}} a_{ij}\mathrm{P}\left(x_T,x_{T-1},\dots,x_{t+1} \middle| z_{t+1}=q_j,\lambda\right)$$

$$=\sum_{\mathrm{j=1}}^{\mathrm{N}} a_{ij}\mathrm{P}\left(\mathrm{x_{t+1}} \middle| \mathrm{x_T},\dots,\mathrm{x_{t+2}},\mathrm{z_{t+1}}=\mathrm{q_j},\lambda\right)\mathrm{P}\left(\mathrm{x_T},\dots,\mathrm{x_{t+2}} \middle| \mathrm{z_{t+1}}=\mathrm{q_j},\lambda\right)$$

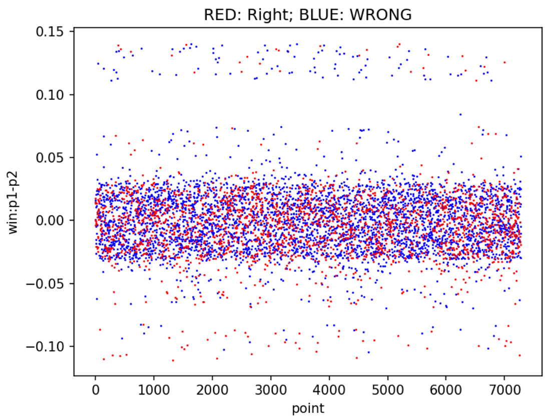
$$= \sum_{j=1}^{N} a_{ij} \mathrm{P}(x_{t+1}|z_{t+1} = q_j)\mathrm{P}(x_T, \dots, x_{t+2}|z_{t+1} = q_j, \lambda)$$

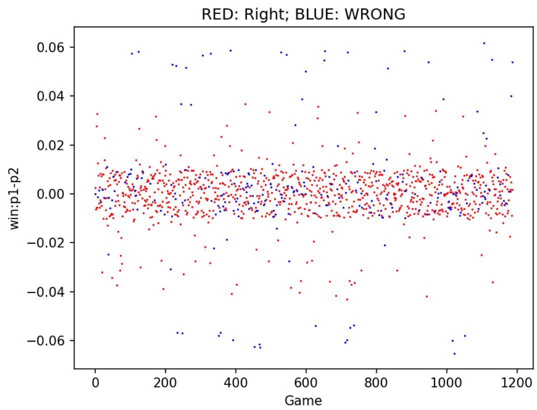
$$= \sum_{j=1}^{N} a_{ij} b_j(x_{t+1})\beta_{t+1}(j)$$

After we got this equation, we can predict the outcome of this **point**.

5.2 Predicting with the Hidden Markov Chain Model (HMM)

The results of single point prediction are as follows.





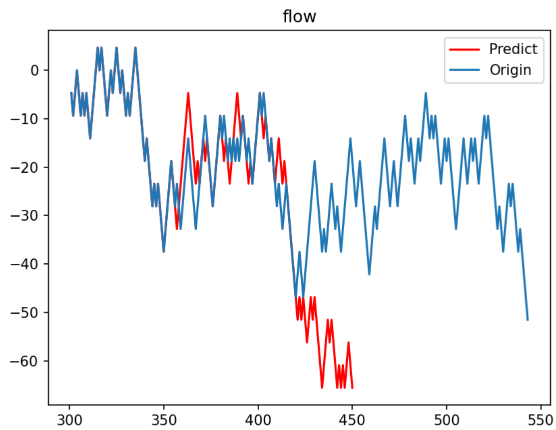
The red dots represent correct predictions, while the blue dots represent incorrect predictions.

Point: There are 6091 dots in total, and 3983 dots of them are red (65.4%).

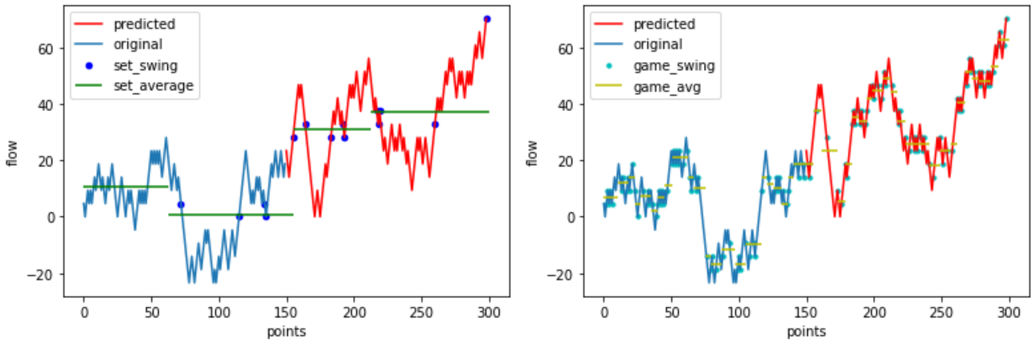
Game: There are 1187 dots in total, and 985 dots of them are red (79.6%).

The predicted results were highly reliant on *server*, without considering who's serving, the predicted success rate of this method drops to 54%.

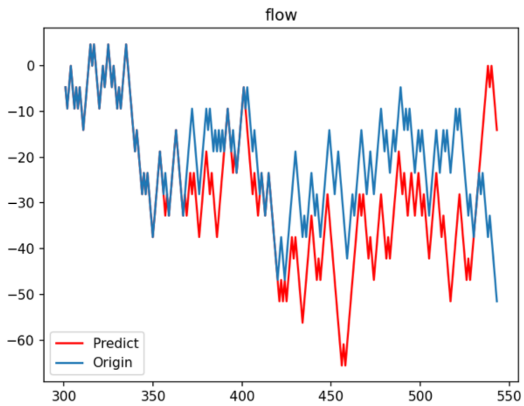
With more iterations, HMM can be used to predict flow, which can in turn be used to predict swing. Flow prediction of the second match using HMM yielded the following results.



The swing prediction of the first match (after point 150) is as follows.



The prediction matched the general outline of the flow, but it veered off at set 3. We suspect the close fit to real data was mainly attributed to *server*. Since HMM can predict who's serving based on probabilities derived from existing data, the frequency of predicted swing is roughly accurate. This can be proved with the following graph, which only considered the predicted *server,* rather than momentum.



Though Markov chain was able to predict the general flow of the game based on probability, considering other factors that may potentially affect momentum actually decreased its reliability.

6 Model Evaluation

6.1 Strengths

1. We used two separate methods to model momentum, both conventional and Deep Learning based, balancing the interpretability of conventional models and the accuracy of Deep Learning models.
2. The fractural nature of our flow model represents real-world sport events accurately.
3. We were able to assess the relative importance of factors using dropout method. Since neural network is exceptionally good at capturing relations, this was strong evidence that momentums were.
4. Multiple statistic methods were used to validate the non-randomness of swings, making the result more credible.

6.2 Weaknesses

1. The model was unable to give accurate predictions due to limitations of data and the complexity the sports event.
2. It takes much time and computing resources to train and tune the Recurrent Neural Network, leaving us little opportunities to optimize the model.
3. The parameters of the Elo model were not optimized, leading to poor results.

7 Conclusion and Further Discussions

7.1 Conclusions

1. The flow of the game and the occurrence of swings are not random. As proven by statistic analysis on streak length data.

2. Momentum does play a role in shaping flow, as average streak lengths are slightly longer than random.

3. The effect of momentum on flow is very small, as the streak length distribution mostly matches random data, and both RNN and Elo models were unable to give results of more than 65% accuracy.

4. It is possible to predict the flow and swing of the game, but it is purely based on probability rather than momentum, and no indicator other than the player serving can be identified.

7.2 Discussion

- **How well do you predict the swings in the match?**

  We were able to roughly predict the flow of the game using probability with good accuracy. However, the correlation between our calculated momentum and flow/swing proved to be very low.

- **If the model performs poorly at times, can you identify any factors that might need to be included in future models?**

  The poor corralation may be due to a lack of data, or inherently limited by the complexity of the game. To somehow capture this level of complexity, more aspects of data is needed. Court surface was proven by research to be vital (Gollub, 2017). In addition, temperature, wind, crowd and other random events could all potentially affect the flow and cause a swing.

- **How generalizable is your model to other matches?**

  The three main models, Elo, RNN and HMM are all generalizable to most one-to-one events, such as table tennis or badminton. However, when generalizing to other tournaments, court surfaces, or sports, the parameters need to be re-adjusted and AI models retrained. One notable exception is that, since tennis has a complicated hierarchical scoring system, the fractural characteristics of flow may not appear in other sports.

8 Letter to the Coach

Dear tennis coaches,

I hope this letter finds you well. In tennis, it is widely acknowledged that “momentum” is significant to the flow of the play. But based on our models, we advise players to pay less attention to “momentum.” In other words, we suggest them to focus on the sport itself, rather than their self-perception. This piece of advice can be supported by the following reason.

As the Oxford Dictionary of Sports Science suggests, “psychological momentum is the positive or negative change in cognition, affect, physiology, and behavior caused by an event or series of events that affects either the perceptions of the competitors or, perhaps, the quality of performance and the outcome of the competition.” Recently, we used a large AI algorithm to model momentum and its impact on the flow of play in tennis matches, attempting to find their relations.

Although, as the result of our data analysis suggest, momentum does have an impact on the flow of the match, the impact is minimal, as the real-world data and the randomly generated ones are highly similar in terms of distribution of streak length. This is further proved by the fact that our models’ poor performance in predicting the match, with their accuracy highly depending on the serving role. This means it is nearly impossible to predict the flow of the match in long term, not to mention its swing.

Best wishes,

Team #2427441

References

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

Gollub, J. (2017). Producing Win Probabilities for Professional Tennis Matches from any Score. https://dash.harvard.edu/bitstream/handle/1/41024787/my_thesis_11_6_final.pdf?sequence=5

Tennis scoring: Points, sets & games: Tennis rules. USTA. https://www.usta.com/en/home/improve/tips-and-instruction/national/tennis-scoring-rules.html

Appendices

https://github.com/TheJavaNoob/MCM2024