A Tennis Paradox

Justine Lai and Corrine Chen

May 21, 2026

Abstract

This paper explores the counterintuitive result that someone loses a tennis match despite winning more points. Using tools in probability, such as binomial random variables and Markov transition matrices, we modeled the probabilities of winning individual games in all scoring scenarios. Such results were then summed to find the probability in the scope of an entire tennis set.

Contents

1 Introduction **2**

2 Concepts **2**
- 2.1 Random Variables 2
- 2.2 Binomial Random Variables 2
- 2.3 Markov Transition Matrix 2

3 A Tennis Paradox **3**

4 Tennis Scoring **3**

5 Solution **4**
- 5.1 Different Cases of Game Scores 4
- 5.2 Considering All Match Combinations 7
- 5.3 Applying Probability Expressions 8

6 Results **11**

7 Conclusion **11**

8 Acknowledgment **12**

1 Introduction

Probability is a branch of mathematics that measures the likelihood of an event occurring. It helps quantify uncertainty, which then allows us to analyze and reason with unpredictable events. Probability is widely used and can be found in branches ranging from weather prediction to artificial intelligence. This paper will introduce two types of random variables and apply them to analyze a tennis paradox.

2 Concepts

2.1 Random Variables

In probability, **random variables** are used to model events with uncertainty. For example, when rolling a standard die, there are multiple possible outcomes. So, we can define a variable X as the outcome of rolling a die. In this case, X is a random variable that takes the values 1,2,3,4,5,6. This set is the **state space**, as it includes all possible values the random variable can take on.

2.2 Binomial Random Variables

Binomial random variables model the number of successes in a fixed number of trials. It has parameters (n, p), where n is the total number of trials and p is the probability of a successful outcome in a single trial. For example, we can look at the number of heads that come up after 5 coin flips. A head on one trial would be considered a successful trial, and the total number of heads throughout the 5 coin tosses can be modeled as a binomial random variable X with parameters (5, 0.5).

With these parameters, the following expression gives the probability that a binomial random variable takes on some number k:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Here, $P(X = k)$ denotes the probability that X is k, or that the number of observed successes is k.

2.3 Markov Transition Matrix

To understand what transition matrices are, we must first learn about Markov random variables.

A **Markov random variable** is a random variable that models states which depend solely on previous states. For example, looking at weather, we might say that our current state is "sunny." At a later time, the weather may be different. This variability between states is modeled by Markov random variables.

A **transition matrix** then organizes the various probabilities of moving between states. In other words, it mathematically maps out the probability of a system transitioning to a specific future state, given its current state. For example, consider the following matrix for different result states of weather:

	X_{t+1} = sunny	X_{t+1} = cloudy	X_{t+1} = rainy
X_t = sunny	0.6	0.3	0.1
X_t = cloudy	0.3	0.3	0.4
X_t = rainy	0.2	0.4	0.4

In the matrix, the rows represent the possible current states at a time t. These states consist of sunny, cloudy, and rainy. The columns represent the possible next states, at time $t + 1$, which are also sunny, cloudy, and rainy. The entries in the matrix organize the probability of moving from one state to its consecutive state. For instance, entry $X_{1,2}$, indicates that there is a 0.3 probability of going from sunny weather to cloudy weather. Note here that the probability of moving into a state depends on the previous state. The first column, for example, shows different probabilities depending on the previous state at time t.

3 A Tennis Paradox

In 2019, World Class tennis players Novak Djokovic and Roger Federer competed in the Wimbledon Final, one of the most prestigious championships in sports. Among the 5 sets of the entire match, here were the point break downs:

Set	**Winner**	**Federer games won**	**Djokovic games won**
1	Djokovic	6	7
2	Federer	6	1
3	Djokovic	6	7
4	Federer	6	4
5	Djokovic	12	13

With the Wimbledon title and £2.35 million on the line, we see that Federer won 36 games to Djokovic's 32! Yet, Federer lost the match as Djokovic ultimately won more sets. This is because Federer won his matches by a larger margin than Djokovic, who won his matches by one point.

Because of the structure of tennis, this occurrence is more common than you might think. Individual points are not as important as winning whole sets, and so it is possible to lose a tennis match while winning more points. But shouldn't the winner be the person who wins most overall? Using the principles above, we will look at the probability of this seemingly unfair outcome.

4 Tennis Scoring

In this paper, we will be looking at the probability that a person wins the entire match, yet won fewer points than their opponent.

Tennis consists of games, sets, and a match. In each **game**, a person wins by scoring 4 points with a 2 points lead. For instance, you can win a game 4-2 (game-30) but not 4-3 (game-40). If during a game the score reaches 3-3 (more formally counted as 40-40), the game enters a **deuce**. In this case, players will play until someone scores two consecutive points and wins the game.

Each **set** is made up of 6 games. To win a set, players must win 6 games with a lead of 2. Like individual games, there is also a deuce scenario where players reach a tie between the number of games won, in which case a tiebreaker is played. For the simplicity of the problem, we will only consider the deuce case within individual games and not the overall set. In other words, the only cases we will consider are scores of 6-0, 6-1, 6-2, 6-3, and 6-4.

In each **match**, players play up to either 2 or 3 sets. The first person to win 2 sets wins the entire match, however, if both players each win one set, a 10-point tiebreaker is initiated. Again, to preserve the simplicity of our question, we will only be considering the results of one set, and so we will not consider this tiebreaker.

5 Solution

5.1 Different Cases of Game Scores

To consider the probability of losing a match while winning more points, we can first simplify the problem by looking at the different point distributions within a single game.

Let $p = P(\text{winning a point})$, $q = P(\text{losing a point})$. We set these probabilities to be able to create expressions. Then we can use binomial random variables to determine the probabilities of winning in different cases. Let $P(X : Y)$ be the probability of winning X games and losing Y games. Since each of the 4 games have a probability p of winning, we know that the probability of winning a game by $4 : 0$ is:

$$P(4 : 0) = p^4$$

Now, we calculate the probability of winning a game by $4 : 1$. We know that the last point cannot be lost. This would result in a game that looks like WWWWL, in which case the game would already be won by the 4th point. Since we need the last point to be a win, the combination function thus becomes $\binom{4}{1}$ instead of $\binom{5}{1}$. We then set the last game as a win by multiplying this expression by p:

$$P(4 : 1) = \binom{4}{1} p^3 qp = 4p^4 q$$

For calculating the probability of winning a game by $4 : 2$, we use the same approach to get the following expression:

$$P(4:2) = \binom{5}{3} p^3 q^2 p = 10p^4q^2$$

We do not need to consider winning by 4 : 3 since you need to lead by two points in order to win. So, the remaining scenario is the deuce, where the players tie 3 : 3 and one player needs to win two consecutive points to win the game.

The probability of getting into a deuce is $\binom{6}{3}p^3q^3$, because we are distributing 3 wins and 3 losses across 6 point opportunities (This will be used later).

From the deuce, we can use a transition matrix to represent the different states within a deuce:

- D = deuce (tied game)
- W = leading with one win
- L = losing with one loss
- WW = leading with two wins, i.e. you have won the game
- LL = losing with two losses, i.e. you have lost the game

	D	W	L	WW	LL
D	0	p	q	0	0
W	q	0	0	p	0
L	p	0	0	0	q
WW	0	0	0	0	0
LL	0	0	0	0	0

The rows of this matrix are the possible current states while the columns are the possible next states (the possibilities after the next point). The entries give probabilities of moving from a current state to a next state. For instance, entry $X_{1,2}$ tells us that the probability of starting at a deuce, then winning by one after the next point is p.

We want to use this matrix to determine the probability of winning from a deuce. However, to get from deuce to winning requires at least 2 steps, because you need to win two points in a row to win a deuce. So, we can approach the probability of winning from a deuce by considering the different states you can enter from a deuce, and then considering the various probabilities of winning from those states. We will map out this approach below.

From the first row of the transition matrix, there is a probability of p entering state W and a probability q of entering state L. No other states are possible. The probability of winning from a deuce can then be broken down into two components:

1. (*probability of entering W from D*) $\cdot$ (*probability of winning from W*)
2. (*probability of entering L from D*) $\cdot$ (*probability of winning from L*)

With the simplification that P(X) denotes the probability of winning from state X, we can then create following equation:

$$P(D) = p \cdot P(W) + q \cdot P(L)$$

Using the same approach, we can expand P(W) and P(L) using rows 2 and 3 of the transition matrix:

$$P(D) = p \cdot [q \cdot P(D) + p \cdot P(WW)] + q \cdot [p \cdot P(D) + q \cdot P(LL)]$$

If a player is in a WW state, they have won two points in a row and have thus won the deuce. This means that the probability of winning from a WW state, or P(WW), is 1. Similarly, if a player is in a LL state, they have automatically lost the deuce. So, P(LL)=0. Substituting these values gives:

$$P(D) = p \cdot [q \cdot P(D) + p \cdot 1] + q \cdot [p \cdot P(D) + q \cdot 0]$$

Conveniently, we can solve for the probability of winning from a deuce:

$$P(D) = \frac{p^2}{1 - 2pq}$$

Lastly, to get the probability of a 0-0 game ending in a win from a deuce, we multiply the probability of entering a deuce in the first place by P(D):

$$P(\text{Winning Deuce}) = [\binom{6}{3} p^3 q^3] \cdot [\frac{p^2}{1 - 2pq}] = \frac{20p^5 q^3}{1 - 2pq}$$

So far, we have determined the probabilities for winning in all scenarios. To get the probabilities of losing (in the same scenarios) we can simply swap p and q. In summary, we end up with:

Game Outcome	Probability Equation
4-0	p^4
4-1	$4p^4 q$
4-2	$10p^4 q^2$
Deuce Win	$\frac{20p^5 q^3}{1-2pq}$
0-4	q^4
1-4	$4q^4 p$
2-4	$10q^4 p^2$
Deuce Loss	$\frac{20q^5 p^3}{1-2pq}$

5.2 Considering All Match Combinations

Now that we know how to calculate for the probability of winning under different outcomes for a single game, we want to consider the probability of winning on a larger scale—winning sets.

To do this, we broke down the problem using casework, with cases of 6-0, 6-1, 6-2, 6-3, and 6-4. In the case of 6-0, we realize that, in each game, the winning player has to have won more points than their opponent in order to win that game. In other words, for all six games, the winning player won more points compared to their opponent and, thus, cannot have won the set without winning fewer points.

In the case of 6-1, we can consider the extreme sub-case in order to determine its possibility as a case for an opponent winning in general but winning fewer points. We consider the extreme by setting every match that the winning player won as a win by 4-2 or deuce, and every match that the opponent won as a win by 4-0. When counting points earned in deuce, the difference in points earned is more important than recording the actual number of points earned by each player. Thus, we can also consider all deuces as 4-2. Now, summing the number of points earned by each player, we confirm that, in the extreme sub-case of our player winning by 6-1, that player will have earned $6 \cdot 4 + 1 \cdot 0 = 24$ points in the set while their opponent will have earned $6 \cdot 2 + 1 \cdot 4 = 16$ points. Since the total number of points earned by the losing opponent isn't larger than the winning player, this case is also not possible and is not included in our evaluation.

In the case of 6-2, we again consider the extreme sub-case to determine its possibility as a case that requires evaluation. Taking each player's points, we know that the winning player will have earned $6 \cdot 4 + 2 \cdot 0 = 24$ points in the set, while their opponent will have earned $6 \cdot 2 + 2 \cdot 4 = 20$ points. Since $20 < 24$, we know that 6-2 is not a case that we need to consider, as it is impossible for the winning player to have accumulated fewer points than the other.

In the case of 6-3, we consider the extreme sub-case again. The extreme scenario for winning by 6-3 results in the winning player earning $6 \cdot 4 + 3 \cdot 0 = 24$ points, while the losing player earns $6 \cdot 2 + 3 \cdot 4 = 24$ points. Since we are solely concerned about the probability of a winning player earning fewer points, we will not analyze this case where both players earn the same amount of points even as one wins the game and one loses it.

In the case of 6-4, however, our extreme sub-case does provide proof for the relevancy of this case. This case sets all the winning player's matches as win-by 4-2 or deuce, and all the losing player's matches as win by 4-0. Since all deuces can be counted as 4-2 as well, the winning player would end with $6 \cdot 4 + 4 \cdot 0 = 24$ points and the losing player with $6 \cdot 2 + 4 \cdot 4 = 28$ points. Since the losing player wins more points in total, even though they lost the set, the case of 6-4 is valid in general and needs to be analyzed for our question.

5.3 Applying Probability Expressions

In our method of brute forcing all cases of a player winning a set by 6-4 yet winning fewer points overall, we have created a Python program to compute the probability of all these cases combined.

Initially, our program ran a simulation that listed out all permutations of possible outcomes for the set. This was done in the form of for-loops and the iterations package in Python. However, when operating the program, we realize that the run-time for this program was only possible in theory and impractical when actually ran. Thus, we turned to using recursion of self-defined functions:

```
2   from functools import lru_cache
3
4   p = float(input("Probability of winning a point: "))
5   q = 1 - p
6   prob = {"-d":(20*(q**5)*(p**3))/(1-2*p*q),
7           -2:10*(q**4)*(p**2), -3:4*(q**4)*p,
8           -4:q**4, "d":(20*(p**5)*(q**3))/(1-2*p*q),
9             2:10*(p**4)*(q**2), 3:4*(p**4)*q, 4:p**4}
10
11  def value(outcome):
12      if outcome == "d":
13          return 2
14      if outcome == "-d":
15          return -2
16      return outcome
17
18  @lru_cache(None)
19  def dp(w, l, score, last_win):
20      if w == 6 and l == 4:
21          if last_win and score < 0:
22              return 1
23          return 0
24
25      res = 0
26
27      if w < 6:
28          for outcome in [2, 3, 4, "d"]:
29              res += prob[outcome] * dp(w + 1, l, score + value(outcome), True)
30
31      if l < 4:
32          for outcome in [-2, -3, -4, "-d"]:
33              res += prob[outcome] * dp(w, l + 1, score + value(outcome), False)
34
35      return res
36
37
38
39  res = dp(0, 0, 0, False)
40  print("Probability of winning a set but winning less points overall: ", res)
```

Figure 1: The full code.

This forty-line code can be broken down into four sections: lines 2-9, lines 10-16, lines 17-36, and lines 37-40. The first section consists of:

```python
from functools import lru_cache

p = float(input("Probability of winning a point: "))
q = 1 - p
prob = {"-d":(20*(q**5)*(p**3))/(1-2*p*q),
        -2:10*(q**4)*(p**2), -3:4*(q**4)*p,
        -4:q**4, "d":(20*(p**5)*(q**3))/(1-2*p*q),
          2:10*(p**4)*(q**2), 3:4*(p**4)*q, 4:p**4}

```

Figure 2: Initiation and Naming of Variables

In this section, we have imported a decorator for functions called "lru_cache" from the module called "functool" in Python's library. When "lru_cache" is decorated onto a self-defined function, Python creates an internal dictionary that stores results in order for the recalling of previous function values to be optimized. In addition, we have also defined variables p and q to be rational numbers that vary depending on what is inputted as p. Lastly, we defined prob to be a dictionary that assigns all potential outcomes for a game, i.e. $\{-d, -2, -3, -4, 2, 3, 4, d\}$, to their respective probability equations that we have determined in section 5.1.

The second section consists of:

```python
def value(outcome):
    if outcome == "d":
        return 2
    if outcome == "-d":
        return -2
    return outcome
```

Figure 3: Definition of the value() Function

In this section, we define value() to be a function with a single required input—the outcome. The format of our outcome is the same as the one that is used in the key values of our dictionary prob, and the function outputs an integer value that represents the difference in number of points earned in a game. In other words, this function is for changing the format of potential deuce outcomes from a string to a number for later use.

The third section consists of:

```python
@lru_cache(None)
def dp(w, l, score, last_win):
    if w == 6 and l == 4:
        if last_win and score < 0:
            return 1
        return 0

    res = 0

    if w < 6:
        for outcome in [2, 3, 4, "d"]:
            res += prob[outcome] * dp(w + 1, l, score + value(outcome), True)

    if l < 4:
        for outcome in [-2, -3, -4, "-d"]:
            res += prob[outcome] * dp(w, l + 1, score + value(outcome), False)

    return res

```

Figure 4: Definition and Implementation of the dp() Function

In this section, we defined a function called dp() and used it with the lru_cache decorator that we called earlier. The function requires four different inputs:

1. "w" is a variable that counts the number of game wins.
2. "l" accounts for the number of games lost.
3. "score" accounts for the difference in amount of points won between the player and their opponent.
4. "last_win" is a boolean variable that stands True when a game win was added in the last call of dp(), and stands False when a game loss was added in the last call of dp().

The function starts with a check of whether the recursion has reached its goal. Thus, if 6-4 games have been simulated, the last game was a win for our player, and the overall score difference was less than 0, meaning that our player won fewer games, the function would return a numerical value 1 or 0 to indicate whether a correct outcome was accounted for. Then, a result variable called res is defined and used to account for the total probability of the player winning but winning fewer points in general. This step is done by recalling dp() to simulate adding a game win or a game loss to a potential sequence of game outcomes.

The last section consists of:

```python
res = dp(0, 0, 0, False)
print("Probability of winning a set but winning less points overall: ", res)
```

Figure 5: Initiating the dp() Function and Printing Results

In this section, we define a variable called "res" to equal the final probability value that dp() returns. Then, we print out this value.

6 Results

As we run the program a few times, each time inputting a different p-value, there are some trends and properties that we noticed. Our p-values and outcomes include:

Inputted p	Total Probability
1	0.0
0.9	$2.682 \cdot 10^{-17}$
0.8	$3.505 \cdot 10^{-10}$
0.7	$1.563 \cdot 10^{-6}$
0.6	0.0001363
0.5	0.00065943
0.4	0.000226
0.3	$4.58 \cdot 10^{-6}$
0.2	$2.1142 \cdot 10^{-9}$
0.1	$4.8966 \cdot 10^{-16}$
0	0.0

Although the probability of this outcome occurring is generally unlikely, it is still a phenomenon that happens in real life. Referencing the table above, it can be observed that the output probability increases with more equal chances of winning a point. Similar to our simplified analysis of game outcomes, we notice that in the Federer vs. Djokovic match, Djokovic won his games with very close scores and lost games with bigger point differences. This resulted in his ability to win despite winning fewer points overall. In addition, we can be assured of the values for total probability when considering that we only reviewed the scenario of winning with a 6-4 score.

7 Conclusion

The results achieved in this paper can be extended to an entire tennis match, considering all tiebreaker cases, to be applied to how this occurrence might happen in reality. The probability of winning more points and losing an entire tennis match will likely be smaller than within a single set, as you would need to satisfy the same conditions on a larger scale. However, the consideration of tie breakers would likely increase the probability of this occurrence. This is because tiebreaker cases have closer point scores, allowing a person to lose by small margins which then increases the likelihood of being able to win back more points.

When extending our results, it is important to note the limitations of our approach. One of the most important simplifications to our approach is the assumption that a person has a constant probability of winning a point. In reality, this probability would be influenced by factors like player morale or tiredness.

Our tennis problem falls within a larger paradox: Simpson's Paradox. This is a statistical phenomenon where an association between two variables is different between subpopulations and a whole population. In the case of tennis, points within individual games act as "subpopulations" to the total points won in a match. The player with more points might be victorious within a game, but not in the overall match. This phenomenon arises well beyond our problem, in a variety of data analyses. In a 1975 analysis on UC Berkeley's admissions, data demonstrated a preference for males in graduate programs. When looking at individual departments, however, no such preference was shown. These instances remind us that probability must also take into account context. In addition to calculations, it is important to consider real world implications in order to achieve reliable results.

8 Acknowledgment

Corrine and Justine would like to thank their mentor Jenny Jin, PRIMES Circle head mentors Mary Stelow and Paige Bright, other PRIMES Circle staff, and MIT for facilitating their participation in the PRIMES Circle program.

Corrine would like to thank her past and present mentors, and the support of her family and friends throughout the program.

Justine would like to thank her family and all PRIMES Circle participants for their contributions to the program.

References

Gentlemen's singles results - wimbledon 2019 [Accessed: 2026-04-30]. (2019). https://imgvideoarchive.com/client/the_wimbledon_archive/results/gentlemens_singles?season=2019

Ross, S. M. (2014). *A first course in probability* (10th ed.). Pearson.

Sprenger, J., & Weinberger, N. (2021). Simpson's Paradox. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Summer 2021). Metaphysics Research Lab, Stanford University.