Choosing Optimal Players Combination in Volleyball Games

Gong Cheng: gc2865
Michael Grandel: mg3856
Ziyang Zhang: zz2732

Project Background

Our goal was to simulate a volleyball game, and figure out which team will win the game. Victory is winning best of 3 sets, and winning a set is winning 25 points. The complexity of a volleyball game is large, and so we choose to start with many simple assumptions and build up from there.

Problem Statement

Given a roster of 12 real life players, determine which combination of 6 players will maximize your chances of winning the volleyball game. Let's assume this is not an adversarial simulation; the opponent has a known and fixed set of 6 players.

Value Proposition

From college sports, to the Olympics, volleyball is everywhere and is a very competitive sport. What makes sports so difficult in general, is that they are adversarial and difficult to predict. But, prediction may not necessarily be impossible and is a growing field of study (artificial intelligence, for example). As a coach, you may have a roster of 12 players, but only 6 players get to play at any given time. Some players are better than others at certain skills, some players get exhausted, and so much more. When a coach needs to choose which 6 players to place, it would be helpful to use statistics and simulations to figure out the optimal combination of players.

Likewise, this is important for sponsors. Sponsors can support those teams that win more often and play smarter. A team using statistics to their advantage will definitely be playing smarter, and so sponsors are more likely to support the right team and players. They may themselves predict which team they wish to support, based on the simulations they run. In addition, building simulation models in sports such as Volleyball will enlighten research and reveal hidden statistical truths about certain elements.

Rules of the Game

Before diving into the details of the skills and simulation, it is first important to lay out the basic rules of the volleyball game, and also the rules we will be using for this simulation. This simulation can expand greatly by invoking more and more rules, so not every single rule is involved. Our jupyter notebook features many more details about the types of players and assumptions we make with our models.

Volleyball is played between two teams each occupying one side of the court with a net in the middle and the ball is hit over the net to the other side of the court by each team. The main idea is, if the ball is on your side of the court - you must NOT let it touch the ground. If it does, then you lose the round. However, if the other team gets the ball over the net and into your side of the court, and the ball lands outside of the court boundaries, then you win the round.

If the ball was indeed going outside the court, but you still chose to touch the ball (on accident or not), then the ball is now considered yours and you must get it over the net.

Your team has 3 touches to get the ball over the net. If on the 3rd touch the ball does not make it over the net and into the other team's court, you lose the point. Note, you can get it over on the first, second, or third. Generally though, the game flows by using all three touches.

If you touch the ball, you may not touch the ball again until someone else on your team touches it. So if you're the first touch, you cannot be the second. But, you can be the third. Another example is if you're the second touch, you cannot touch the ball again (because that means you either touched the 3rd time, which is a consecutive touch, or you touched the 4th time, which is over the touch limit). If you do happen to touch it a second time, your team loses the round.

The net cannot be touched by any player. If you touch the net, you lose the point. However, the ball is allowed to touch the net. Say your team uses the first touch to pass the ball and accidentally passes it into the net. You still are allowed to use a second touch to pass the ball, but you must be careful to not touch the net. If you used the third touch and got the ball into the net, you ran out of touches so you cannot save the ball with a fourth touch, and so it'll fall down onto your side and you lose the round.

Model Construction

The model simulates a 25-point game between the two teams chosen for multiple times with the winning team and game duration as return. Players from each team have five possible actions: serve, pass, set, attack and block, with each action having different possible outcomes listed below. The actions have three possible outcomes: the ball kept in play, point won and point lost. 'Point Won' is missing in some actions as they are non-offensive in nature.

Action	Outcome		
	Still in Play	Point Won	Point Lost
Serve	Success	Ace	Error
Attack	Success	Kill	Error
Pass	Success	/	Error
Block	Success	Kill	Error
Set	Success	/	Error

From the start, a team is randomly selected to serve then game proceeds following rules described above. That is, the receiving side formulates an offense through pass-set-attack action sequence. In response, the serving side performs blocking. If the ball is still in play, it could be bumped back to the offending side or drifted to the blocking side with speed and momentum reduced. Whichever happens the party with the ball formulates a new offense with a new sequence of pass-set-attack. The rally might continue back and forth several times until the point is decided.

In the model, we assumed there is one setter on both sides exclusively setting for the team. In this setup, all setting actions are performed by the setter, therefore, due to no consecutive touch, setters are not involved in passing and attacking. Further we assumed for each passing and attacking, one player is randomly selected from the team except setter. Each player has their own unique statistics in probability of outcomes for each action. For example, some players have a higher probability of 'Ace' in attacking but also higher chance of 'Error' in pass and some players are more balanced in each of the five skills.

Whenever a player is selected for each action, here unique statistics are used to determine the random outcome in simulation. However, as blocking involves multiple players, mostly from 2 to 3, it's hard to determine the probability of outcome for every scenario involved given different combinations of players. Therefore, although each player has her own statistics for blocking, we used team average as a proxy for every blocking action.

Results of Our Simulation

Analysis of the System Output:
Given a set of 6 players from our team and another set of 6 players from the opposing team, we simulate 100 games and the system output will include the winning percentage for our team and average length of the game in seconds.

Target Audience: Coach and Sponsor

1. The coach would like the winning percentage to be as high as possible.
2. The sponsor would like the length of a game to be as long as possible, given that the team is winning for the majority of the time.

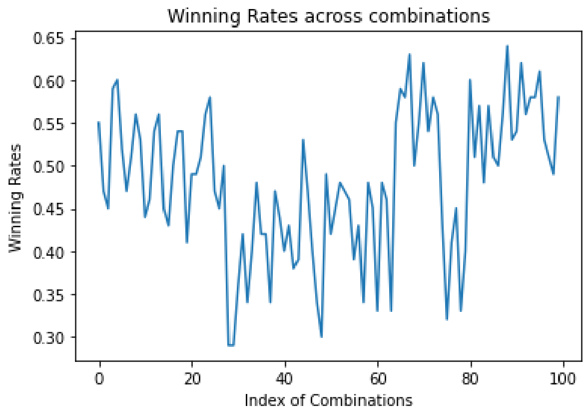
Optimization Proposal:
Every night our team is facing a different opponent, and each opposing team has a different set of 6 players. Given that our opponent tonight picked their roster by experience and we know their fixed set of 6 opponents ahead of the game, we can quickly run our simulations to adjust our optimal roster that best fits our interests.

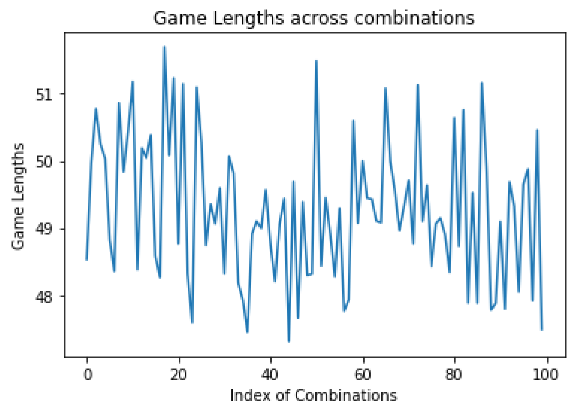
We put the winning percentage as the priority, but it doesn't always have to be the highest.
Given a lower bound for the winning percentage that is required by the coach, we pick the team combination that plays the game to the longest duration, because that will give our sponsor more brand exposure if our players played a little longer.

Therefore, we can run our simulations on all different combinations of 6 players out of 12 players in our team while keeping the opposing team's selection of 6 players fixed, assuming that we are the only team that is using this simulation technique.

There are 924 possible combinations in total. We only select the first 100 combinations as our choices, and simulate the games with our opposing team. We will analyze the top 10 winning rates first, and then we will compare the lengths of those games and pick the combination that leads to the longest game to maximize the interest of our sponsors.

Optimization Results:





We ran the optimization process and computed the Top 10 winning rates with their game lengths. According to the plots above, we can clearly see that the different combinations do affect the result of the game. The winning rate can go as high as 64% and as low as 29%.

However, the length of the game turns out to be trivial in our measures, because the interval of the Game Lengths is very narrow and Top 10 rosters all last around 49 seconds in playing time.

Therefore, we will go with the combination that has the highest winning rate at 64%.
The optimal roster we get is: *Kyle Piekarski, Quinn Isaacson, Oscar Fiorentino, George Huhmann, Casey McGarry, Liam Maxwell.*

Discussion and Conclusions

Contributions of our model and result:
Our model provides a simple simulation. There are lot's of assumptions made, which is discussed in the limitations below. But, due to the modularity of our model, it is simple to remove some assumptions.

For example, when a player performs an action on a ball, our model does not consider the distance between the player and the ball. We only use the distance to see if the ball will be picked up by somebody near. But, the probability of a successful action does not depend on this distance between the two, only the rating. You can easily add in this dependency by adding the probability model of choice (say a 2D gaussian centered around the player's position), and then normalize the model.

We also assumed only single blockers - no double or triple blocks. Incorporating this isn't too simple, however. But our model contributes a great foundation for ideas to form and assumptions to peel.

Limitations of our model and result:

Though the model constructed is a genuine approximation of a real game, it suffers from several limitation:

1) Some assumptions do not hold in real games, for example, setters do sometimes pass, attackers have different chances to attack given their position and blocking action is much more complicated than using team average statistics as an approximation.
2) The simulation is finished as soon as possible while a real game usually lasts between 25-35 minutes if considering all the timeouts called and natural breaks between points.
3) Many more factors are not considered in this simulation, for example, as time elapses players have more chances of errors and players might under or over perform than its past record in a specific game.

Role and contribution of players:

Gong Cheng: Integrated each action process into a simulation model for the whole game, generated simulation outputs for each round of game.

Ziyang Zhang: Analyze the system output of the simulation. Define the goals target audience and run the optimization process. Interpret the optimization results.

Michael Grandel: Set up the problem statement and probability models, gathered the data, built initial simulation via trajectories.

Links:
Jupyter Notebook:
https://colab.research.google.com/drive/1li2srDPNDRqUS5JQYnNt_uqIC3VeYTyw?usp=sharing
Dataset: https://drive.google.com/file/d/1itj7Kllfn_YwAMisEsYUsgCPlhBAEs7v/view?usp=sharing
Video Presentation:
https://columbia.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=44a50cf4-ce44-4451-813b-aca00050804a