Using Paired Comparison Models to Predict Hitbox Advantage

Jordan Love

March 2019

1 Background

SaltyBet is an online, 24/7 nonstop, A.I driven street fighter game where viewers are given fake "Salty Bucks" to bet on their favorite characters. SaltyBet is hosted on Twitch, a popular video game streaming website where users can stream and commentate while they play. SaltyBet was one of many to creatively use the stream to foster more viewer engagement. Most notably, the popular "Twitch Plays Pokemon" stream cites as an inspirational source for a non-standard format. [Ref] While an official launch date for SaltyBet is not currently known, it has been running since at least [DATE] 2014 [Twitter reference]. Since then, the twitch stream has maintained a fairly consistent average of 400 users over the past year as reported by SullyGnome, a third-party twitch statistics [Reference]. A typical screen during a match is shown in Figure 1.

Shortly after its inception, SaltyBet added a premium feature which allowed users to access all previous match data. This spawned the creation of many viewers opting to scrape and then use this data to build bots which automatically bet on characters. [Reference] These bots consist of many different types

of algorithms. One of the more popular bots applies a genetic algorithm to determine which character within a match will win. [Reference] Among the bots programmed to predict winners, none found apply a probabilistic approach to modeling characters latent ranking or incorporate immutable features of each character.

Each character consists of several features or traits which define the performance of the character. SaltyBet operates off of the MUGEN engine which was developed by "elecbyte" in early 2002. This engine clones the basic features of classic Street Fighter series of games originally developed by Capcom. [Reference] The engine also comes with specifications in order to create customer characters. Since then, a large number of characters have been created by users to use and compete in the MUGEN world. SaltyBet itself employs 5,777 unique characters during its matches. [REF] Each character must consist of a set of images which define the characters movement and moveset. A moveset describes all of the actions and motions a character can make to attack another character. The image used to define a character defines its hitbox: the area on the screen where a character can receive damage. Several example characters are shown in Figure 2. Notice specifically the large discrepancy in hitbox size between many of the characters. Each character is also equipped with an AI script which defines how the character will attack and respond to other attacks.

The goal of this paper is to develop a probabilistic framework for evaluating the latent strength of characters and include additional information about the characters to improve predictive accuracy. Specifically, a focus will be given to identifying hitbox disparities between characters through this model to determine if this is a substantial predictor and at what level of the disparity does it arise. To do this, we will perform a brief review of Paired-Comaprison (PC) Models and examine similar approaches in other popular sports. We will focus

on Bayesian methods as this is of interest for future modeling efforts.

2 The Data

For this project, historic matches were scraped from the SaltyBet website through the premium functionality provided. There were a total of TODO: X matches between Y characters. The scraping script was written using Python and the Selenium module [X], the code appendix contains code for scraping each match and character. Due to the restrictions placed on the number of server calls each user can make to the SaltyBet server per hour, scraping for the entire dataset took approximately TODO: Z days. Once scraped, the data were formatted and into a PostgreSQL database.

Within saltybet, characters are divided into five distinct tiers: X, S, A, B, and P. These tiers are assigned based on the performance of each character previously. Characters are promoted or demoted based on their performance directly after a match. There are three distinct types of matches: Matchmaking, Tournament, and Exhibition. The matchmaking mode algorithmically chooses players to match up against each other where the odds are approximately equal of each character winning. Tournament mode is a random set of 16 characters from a specific tier who fight each other in a single-elimination tournament. Finally, exhibition mode is a set of viewer-requested matches which also allows teams of characters to compete. Exhibition mode games are typically chosen by viewers to force edge-case behavior of the characters. Many times, viewer requested matches result in a server crash due to intense computational loads.

Characters were scraped individually and stored with the following information: Author, Life, Meter, Hitbox Width, and Hitbox Height. The scraping process for characters was similar to the process for scraping match data and the code appendix contains a modified script for character data scraping. Of all

the characters scraped, TODO: X had no image associated with their profile. For the purpose of this project, we excluded these characters. The number of matches affected by this removal is TODO: Z. Figure 2 showed an example of differing hitboxes, but

3   Models

The goal of Paired-Comaprison (PC) Models are to quantify the probability of a subject choosing one option over another or, in sports, one team winning over another. Each PC model has some formulation of the latent "ranking" either of strength of team or underlying preference. The general form of a PC model has the form of equation 1. In this case, $f(x)$ represents a transformation of the structure of the latent strength.

$$P(A > B) = f(x)$$

One of the first models to address these issues was the Thurstone-Molster TODO: Check spelling— model. In this model, the underlying strength or preference is modeled as a normal distribution. Thurstone and Mollster derived several simplifications of this model to ease computation. One of the most computational simple choices is Case V of the Thurstone and Mollster Model. In this model, variances are assumed to be equal and correlation is TODO: Blah. Use of the normal distribution allows for us to derive a theoretically simple equation for the paired comparison.

$$A \sim N(\mu_a, \sigma_a), B \sim N(\mu_b, \sigma_b)$$

$$P(A > B) = P(A - B > 0) = A - B \sim N(\mu_a - \mu_b, \sigma_a + \sigma_b) = \int_0^\infty$$

The last line of the above derivation shows that in order to compute the probability of interest with these underlying assumptions, we must compute the "error" function. This function is well known in probability and statistics, but no closed form solution exists. This creates a computational issue immediately.

The next development within Paired-Comparison Models was the Bradley-Terry model developed by TODO:. This model changed the underlying strength or preference distribution to be distributed as a Gumbel or Extreme Value Distribution. Using this assumption, a concise result is obtained through an similar process for the Thurstone-Mollster model. Using the CDF method to compute the difference between the two Gumbel distributions:

$$P(A > B) = P(A - B > 0) = A - B \sim Logistic(\mu_a - \mu_b, \beta) = TODO :$$

This reduces to a convient logistic function which can be evaluated more quickly than the required error function evaluation of the Thurstone-Mollster model. While these models make differing assumptions regarding the underlying strength or preference distribution, the results are notably neglible [TODO: Reference from Glickman]. Figure 3 shows a comparison of shapes of the CDFs. The logistic distibution has slightly longer tails than the normal distribution and is more sloped at zero than the normal distribution. For the purposes of this paper, we will focus on the Bradley-Terry formulation of the Paired-Comparison model. It is worth noting that a bayesian approach to the Thurstone-Mollster model exists and a Gibbs-sampling procedure is described in Yao [REFERENCE]. Simulation study 1 in Section 3 addresses model basics and a comparison between

these two models.

3.1 Adding Additional Predictors

In many cases, there are additional predictors which assist in identifying which preference will be chosen or team will win. In sports, a home-field advantage terms is included in the model. [Cricket and Baseball R package] This term enters the model as a categorical variable describing which team is home and which is away with the home teaming being coded as a 1. In general, we would like to add a variety of covariates to the model. The general formula for adding covariates to a Bradley-Terry model is discussed in detail in [REFERENCE] and is shown below in equation 4. [1]

References

[1]   Heather Turner and David Firth. "Bradley-Terry Models in R: The BradleyTerry2 Package". In: *Journal of Statistical Software* 48.9 (2012), pp. 1–21. URL: http://www.jstatsoft.org/v48/i09/.