April 15, 2019

1 Background & Motivation

Sports prediction and betting have history, especially in the United States. While the prediction of the outcomes of sports has always existed, more statistical approaches have only become more popular with the revolutionary use of data in many sports. The rise of Sabremetrics within Baseball has acted as a catalyst for many other sports to begin accepting data-driven prediction and prescription. Sports betting alone has become a large political issue within the previous two years as large online fantasy sports sites such as DraftKings and FanDuel gained popularity. While the prediction of sports games will continue, sports betting while remain a contenious issue for the foreseable future in many states within the US. However, the statistical question of predicting outcomes and then assigning value to the prediction is still inherently interesting and has many parallels within financial markets.

The motivation for this project has three key aspects. The first is to explore the sparsity and connectivity reqiurements on the datasets which are

used in Paired-Comparison models. These requirements will be formulated in the language of Graph theory. Algorithms which can be used as a diagnostic for paired-comparison datasets are discussed, and a simulation related to the dataset analyzed in this project is analyzed. The second is to retain the context of prediction and prescription as found in many fantasy sports while also using a novel dataset which has the potential for real-time prediction. The final motivating factor is to use models which integrate with existing decision optimization methods. We will begin by describing the source of the dataset for this project: SaltyBet.

1.1 What is SaltyBet?

SaltyBet is an online, nonstop, "Street Fighter" game where A.I. driven characters fight against each other and human viewers are given fake "Salty Bucks" to bet on the outcome of each match. [**?**] SaltyBet is hosted on the Twitch platform, a popular video game streaming website where users can stream and commentate in real-time. SaltyBet was among the first to creatively use the platform to provide a unique viewer expierence by the entire stream automated. It has served as a forerunner in this form of stream automation with the even more popular stream, "Twitch Plays Pokemon", citing SaltyBet an inspiration. [**?**] An official launch date for SaltyBet cannot be verified. However, it has been running since at least April 26th, 2013 based on its accompanying twitter account which tweets important match outcomes. [**?**] Since then, the Twitch stream has maintained a fairly consis-

tent average of 400 users over the past year as reported by SullyGnome, a third-party Twitch platform statistics website. [?] A typical screen during a live match on SaltyBet.com is shown in Figure 1.

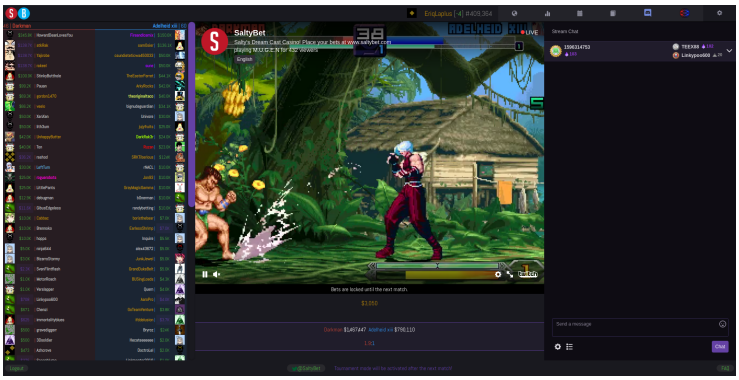

Figure 1: Typical screen at SaltyBet.com of a match in progress – The left vertical bar shows the users who have bet on the current match alongside their bet amount, the center vertical bar contains the match in progress and match odds, the right vertical bar shows the chat where users communicate.

Shortly after its inception, SaltyBet added a premium feature which allowed viewers to access all previous match data. This spawned the creation of many automated "betting bots" as viewers began to scrape and use the previous match data as a training set. [?, ?] These bots have consisted of many different types of algorithms. One of the more popular bots applies a genetic algorithm to rank and then predict the outcome of each match. [?] Among the bots reviewed for this project, none were found to apply a probabilistic approach to modeling each character's latent strength or incorporate other features of each character.

Within SaltyBet, each character consists of several features or traits which

define the performance of the character. SaltyBet itself operates off of the MUGEN engine which was developed by "elecbyte" in early 2002. [**?**] This engine clones the basic features of the classic Street Fighter series of games originally developed by Capcom beginning in 1987. The engines is designed with specifications to allow anyone to create custom characters. Since MUGEN's launch, a large number of characters have been created by the surrounding community. Of these created characters, 5,777 have fought at least one match on SaltyBet.com. The definition of each character must consist of a set of images which define the characters movement or "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 also defines its "hitbox": the area on the screen where a character can receive damage from other characters. Two example characters are shown in Figure 2. Notice specifically the large discrepancy in the hitbox height between the character. Each character is also equipped with an AI script which defines how the character will attack and respond to other attacks. Finally, there are numeric values describing the attack strength, health, and meter (a measure of accessiblity to highly performant attacks) associated with each character. Within the SaltyBet community, there exists is a large number of theories indicating discrepancies in the size of each character's hitbox can be highly predictive of match outcomes. [**?**] One goal of this project is to examine this hitbox advanrage hypothesis.

(a) 69 x 117 pixel hitbox

(b) 18 x 18 pixel hitbox

1.2 Bayesian Methods

In order to address the hitbox advantage hypothesis, it is necessary to develop a probabilistic framework for evaluating the latent strength of characters and include additional information about the characters withint he model. To do this, we will use Paired-Comparison Models. Specifically, a focus will be given to identifying hitbox advantages between characters through this model and determine at what level of hitbox differential advantages begin to arise. To do this, we will perform a brief review of Paired-Comaprison (PC) Models and extensions for modeling additional information between characters..

It is of interest in this project to employ a Bayesian framework for a number of reasons. Bayesian methods often have large computational cost depending on this complexity of the model, but there are also many advantages. The advantages relevant to this project are 1. a natural estimation of variance, 2. a natural recursive formula for online estimation, and 3. a natural integration into a Markov Decision Process.

1.2.1 A Natural Estimation of Variance

While it is beyond the scope of this project to detail the differences between Frequentist and Bayesian methods, we will discuss specifically the advantage of natural estimation of variability. One of the primary differences between Bayesian and Frequentist methods is that the parameter under estimation is assumed to be random as opposed to fixed. This allows the workflow of a Bayesian modeling problem to be interpretted, through use of Bayes' rule, in terms of probability distribtuions throughout. While not always trivial to obtain the resulting posterior distribution of a parameter within a complex model, all information surrounding the parameter is contained within the samples of the posterior distribution. This includes variability which is simply a transformation of the samples of the posterior distribution.

Within this and similar contexts, the interest is not only in predicting a point estimate describing which character or sports team is estimated to be most likely to win but also quantifying our uncertainty around the estimate. This is not unlike financial markets where we are not only concered with the overall performance but also the potential risk for large swings within the market. This allows us to make proper decision about how to prescribe actions based on the estimates from the model. Since one of the ideal applications of this model would be to form a "trading strategy", Bayesian modeling allows access to all necessary information.

1.2.2 A Natural Recursive Formula

Another advantage of Bayesian modeling is the natural recursive formula which can be formed by using the posteriors estimated at time $t-1$ as the prior distributions at time t. Many other statistical and machine learning techniques do allow this recursive estimation procedure to be implemented easily. Most notably, Recursive Least Squares, an estimation technique found commonly in Signal Processing possesses a similar framework. [CITE] However, as with many other estimation techniques, Bayesian methods typically encompass a larger class of estimation procedures with specific prior choices being commonly known in literature (Regularized Least Squares, LASSO) [CITE]. This is no different for Recursive Least Squares with D.S.G. Pollock describing the relationship from a bayesian perspective. [CITE] Since one of the motivating factors of the SaltyBet dataset is the potentially real-time prediction of each match, Bayesian modeling is a natural approach for this problem.

1.2.3 Natural Integration with Markov Decision Processes

Markov Decision Processes (MDPs) are a formulation of the problem of the optimally moving around a space given some set of actions and known rewards. This optimization problem is often intractable to specify aprior due to the unknown rewards associated with each action and lack of observability of the entire system. This leads to a class of MDPs known as Partially Observable Markov Decision Processes (POMDPs) where the estimation of the

current state of the system and optimal policy are estimated simultaneously. POMDPs are notoriously difficult to solve optimally, but the framework allows for many numerical approximation methods to be applied. This framework can be seen in many contexts such as Air Traffic Control, Surveliance, and Robotics. [].

Since MDPs and POMDPs are both built first mathematically on the theory of probability, Bayesian modeling provides a seamless integration into these methods for two reasons. These two methods have already been described in previous advantages. The first is that we have a natural interpretation of variability in the language of probability constructed directly into our estimation procecure, the second is that the goal of both MDPs and POMDPs is typically not to make a single decision but to make multiple decisions over time. This allows the natural recursiveness of bayesian methods again to seamlessly integrate with this method. For this reason, Bayesian methods are often the de-facto estimation procedures when dealing with these problems.

While implementing a POMDP is beyond the scope of this project, this framework was an important deciding factor in what model to choose to use as a natural extension from estimating the latent strength of characters and predicting outcomes is to operatialize and optimize both which bets to make and how much to bet based on the information available.

TODO: Need to transfer citations.

1.3 Data Collection

For this project, historic matches were scraped from the SaltyBet website through the premium account functionality provided. There were a total of TODO: X matches between Y characters. The scraping script was written using Python and the Selenium module [CITE]. The code appendix contains the code which was used to web-scrape the data from each match.

1.4 Struture of this Project

This paper is divided into four primary sections. The first section describes the literature surrounding the Bradley-Terry models and the introduction of additional information. A seperate subsection within models describes the various estimation techniques including Maximum Likelihood Estimation and Bayesian Estimation. The second primary section discusses the assumptions of connectivity within paired-comparison models. In this section, we describe algorithms for determining if a dataset is sufficiently connected and what options are available if this assumption is not satisfied. In this section, we perform a graph-based simulation to determine the probaiblity of connectivity within the dataset being analyzed. In the third primary section, we perform three seperate analyses using methods discussed in the model section and discuss their differences. We also describe all data cleaning steps taken and the results of the prediction of each of the three models fit on 500 additional matches. Finally, the concluding section concisely describes the

results of the analysis and what future work may be performed to improve the work of this project.

2 Temporary Work

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 web-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