

My title...

Jordan R. Love

Department of Mathematical Sciences
Montana State University

May 3, 2019

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

Jordan R. Love

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date

Andrew B. Hoegh
Writing Project Advisor

Date

Mark C. Greenwood
Writing Projects Coordinator

Contents

1	Background	3
2	The Data	5
3	Models	6
3.1	Adding Additional Predictors	9
3.1.1	”Home-Field” Advantage	9
4	Estimating Strength Parameters	10
4.1	Maximum Likelihood Estimation	10
4.2	Minorization-Maximization Algorithms	10
4.3	Bayesian Estimation Methods	11
5	Comparison Graph Connectivity	12
6	References - Left in for... reference	16
7	Appendix-R Code	17

Abstract

Abstract goes here.

April 5, 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 custom 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:

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

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 distribution 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 term 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.

3.1.1 "Home-Field" Advantage

Agresti (1990) TODO: Add Reference discusses the inclusion of an effect which weights the team which has an advantage due to playing in their home stadium. This modifies the likelihood function of the Bradley-Terry model in the following way. Agresti includes a factor, θ , for the home team which is multiplied to their estimated latent strength within the model.

TODO: Discuss what the home-field advantage model accomplishes

In the model section, we have introduced the original paired comparison model, the Thurstone Model, and discussed the assumptions associated with a specific case of the model (Case V) in depth. We then introduced the

Bradley-Terry model by noting the motivation of the model was opposite that of the Thurstone model: instead of a convenient latent distribution being chosen, a desirable difference CDF was chosen and distributions which provided this results were found: the Gumbel or Extreme Value Distribution. Finally, we discussed the modifications required to add covariates and a "home-field advantage" factor to the Bradley-Terry model.

4 Estimating Strength Parameters

4.1 Maximum Likelihood Estimation

The original algorithm for estimating probabilities of paired comparisons was developed before Thurstone formulated a model in full. Zermelo in 1929 developed and proved an algorithm which converged to a unique set of parameter estimates given certain conditions were met. These conditions are discussed more fully in section X (TODO:) as an analysis of the comparison graph. The algorithm described by Zermelo is described below:

TODO: Algorithm Box

TODO: Discuss Algorithm Intuitively and results and connection to MLE

4.2 Minorization-Maximization Algorithms

While the algorithm described by Zermelo handles the most basic case, it was not extended to consider covariate models or "home-field advantage"

models. In this case, a more general theory of estimation algorithms surrounding Bradley-Terry models were developed. Lange, Hunter and Yang (2000) showed the algorithm developed by Zermelo is a specific case of a more general class of algorithms known as Minorization-Maximization (MM) Algorithms. One well known special case extending from this class of algorithms is the Expectation-Maximization (EM) Algorithm. Heiser (1995) TODO: describes in detail the connection between the MM and EM algorithms.

Since the algorithm proved by Zermelo is a special case of MM algorithms, the estimation procedure only changes in notation between the original and latest literature. Instead, we provide the MM algorithm estimation procedure for the "home-field advantage" model discussed previously and note the key differences.

TODO: Algorithm Box for MM Algorithm for "home-field advantage"

TODO: Discuss key differences

4.3 Bayesian Estimation Methods

While a review of classical statistical methods have been review previously, the goal of this project is to develop a Bayesian view of Bradley-Terry models with the intent for these models to be used in conjunction with the mathematics of Markov Decision Processes. For this, we refer to the work of Caron and Doucet TODO: add reference. Their paper "Efficient Bayesian Inference for Generalized Bradley-Terry Models" describes the construction of Gibbs

samplers for both the typical comparison and "home-field advantage" models through the reformulation of the likelihood function via different latent variables.

In the original case, the latent variables of interest are the latent strength distributions which are denoted as λ_i in the above discussion. Instead of introducing this model, Caron and Doucet instead introduce the latent variable $Z_{i,j} = \min(Y_{kj}, Y_{ki})$ where each Y_{kj}, Y_{ki} is a realization from the underlying strength distribution of competitors indexed by i and j , respectively. This allows the difference between two competitors to be captured in the new latent variable $Z_{i,j}$ as opposed to separate latent variables λ_i and λ_j . However, by properties of the exponential distribution, we have that:

$$Z_{i,j} \sim \text{Gamma}(n_{ij}, \lambda_i + \lambda_j)$$

TODO:

5 Comparison Graph Connectivity

One key assumption when using Bradley-Terry models for paired comparisons is the data is in such a format the comparisons of interest can be estimated. In many cases when a paired-comparison model is employed, all items being compared have been compared to each other at least once and in each comparison at least one has been chosen over the other. One efficient way to formulate these assumptions is through graph theory.

An first principles view of graph theory is beyond the scope of this paper, but the author refers interested readers to West (TODO:). A graph is defined as nodes and edges. Nodes represent atomic items such as sports teams or choices where edges may represent matches between teams or preferences between choices. A graph can be an undirected or directed graph. In the case of an undirected graph, a single edge is placed between two nodes indicating a general relationship. Within an undirected graph, consider moving between two nodes whenever any edge connects them. A directed graph adds an additional layer for direction. For the purposes of this paper, a directed edge from between two nodes indicates the outcome of a match where the direction will extend from the winning node to the defeated node. A directed graph restricts movement between nodes to movements allowed by the directed edges.

Formally, we will define a graph as an object consisting of two sets: nodes and edges. We will index the nodes as $1 \leq i < j \leq k$, where k represents the number of nodes in the graph. We will refer to edges as $e_{i,j}$, where this represents the directed edge from i to j unless specified we are working in an undirected setting. We will define the **comparison graph** as the directed graph consisting of all characters as nodes and all matches played between each character as directed edges. Figure X shows a sample comparison graph.

TODO

The original constraint for paired-comparison data to be suitable analyzed such that all comparisons can be estimated was formulated by Ford

in 1957. In graph theoretic terms, the condition was that the comparison graph constructed from the data be **strongly connected**. This condition necessitates that within a directed graph any node can be reached by starting at any other node in the graph and moving along directed edges.

In order to evaluate the feasibility of this constraint within a dataset of paired-comparisons, we can use well known algorithms to determine the connectivity of the comparison graph. Algorithm X (TODO) walks through the steps of determining connectivity within a graph. The primary notion of the algorithm is to determine what nodes can be accessed from all other nodes and if not how many disconnected components exist.

TODO: Algorithm X

While strong connectivity of the comparison graph allows comparisons without modification, Yan (2014) discusses how using a singular perturbation method can allow for comparisons to be made when the graph is only **weakly connected**. Weakly connected graphs are directed graphs whereby if all directed edges are converted to undirected edges, the graph is connected or all nodes can be reached from any other node. Figure Y shows an example of this situation in detail. In Figure Ya, we see the original directed graph with node A having only directed edges away from it. This can be known in other graph theoretic literature as a "source." However, if we remove the direction from each of the edges, we see in Figure Yb that the undirected version of this graph is connected. One intuitive reason for why weakly connected graphs are not sufficient without modification is that one team is undefeated

and without an additional measure of "strength of win", the magnitude of how much the latent strength of this team differs from the next best team is unknown.

The modification discussed by Yan (2014) is by adding a penalized term to the likelihood in such a way that a "pseudo-loss" to at least one other team is shown. This term can be re-interpreted as a specific prior distribution to allow for the penalized term to exist.[?]

6 References - Left in for... reference

References

- [1] Ella Kaye and David Firth. *BradleyTerryScalable: Fits the Bradley-Terry Model to Potentially Large and Sparse Networks of Comparison Data*. R package version 0.1.0. 2017. URL: <https://CRAN.R-project.org/package=BradleyTerryScalable>.

7 Appendix-R Code

All that sweet sweet R code.

Verbatim would also work here I think.