Ranking and Prediction for Cycling Canada

by

Richard R. C. Hsia

B.Sc., University of British Columbia, 2016

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Fall 2017

Approval

Name: **Richard R. C. Hsia**

Degree: **Master of Science (Statistics)**

Title: **Ranking and Prediction for Cycling Canada**

Examining Committee:

Chair: Jinko Graham
Professor

Tim Swartz
Senior Supervisor
Professor

Joan Hu
Supervisor
Professor

Dave Clarke
External Examiner
Assistant Professor
Biomedical Physiology and Kinesiology

Date Defended: **December 14, 2017**

Abstract

In efforts to improve Canadian performance in the men's Elite UCI Mountain Bike World Cup, researchers from the Canadian Sport Institute Ontario (CSIO) presented to us a specific problem. They had a wealth of race data but were unsure how to best extract insights from the dataset. We responded to their request by building an interactive user interface with R Shiny to obtain rider rankings. Estimation was carried out via maximum likelihood using the Bradley-Terry model. We improved on the existing literature, proposed an exponentially weighted version of the model, and determined an optimal weighting parameter through cross-validation involving performance of future races. Therefore, the proposed methods provide forecasting capability. The tuned Bradley-Terry estimation performed better than the UCI point-based ranking in terms of predictive error. This implementation of the Bradley-Terry model with a user-friendly graphical interface provides broader scientific audiences easy access to Bradley-Terry ranking for prediction in racing sports.

Keywords: cycling; ranking; prediction; Bradley-Terry; pairwise-comparisons

Acknowledgements

I would first like to thank my senior supervisor, Professor Tim Swartz. My experience as a graduate student would not be the same without your mentorship and guidance.

My gratitude also goes out to all of my professors here at Simon Fraser University. I want to extend special thanks to Professor Joan Hu for serving on my committee and Professor Rachel Altman for helping me get a kickstart on my project. I also want to thank Dr. Qian (Michelle) Zhou for originally getting me interested in the statistics program here.

Many thanks to Dr. Dave Clarke for serving on my committee and also for getting us in contact with Canadian Sport Institute Ontario (CSIO). I would also like to thank Mike Patton and Emily Wood of CSIO who provided us with the data.

I want to express my deepest appreciation to my family. To my wife, thank you for being my second brain. To my parents, thank you for your constant support and understanding. To my brother, you didn't really help at all.

Lastly, I want to thank my friends and fellow students for their support—from the one who asks too many questions to the one who is always late.

Table of Contents

Approval ii

Abstract iii

Acknowledgements iv

Table of Contents v

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Motivation 1

1.2 Literature review 2

1.3 Approach 3

2 Data 5

3 Models 9

3.1 Formulation 9

3.2 Ability estimation 11

3.3 Weighted Bradley-Terry 12

3.4 Cross-validation 14

4 Predictions and results 16

4.1 Cross-validation and predictions 16

4.2 Point-based ranking comparison 19

4.3 App for Cycling Canada 21

5 Discussion 26

Bibliography 28

Appendix A Cross-validation code 29

Appendix B Shiny app code **35**

List of Tables

Table 2.1 Example of Excel data from CSIO. 6
Table 2.2 Example of data format after conversion. 6

Table 4.1 Actual results and ranking ($\hat{\theta} = 0.074$) for the top 20 finishers of race 27. 18
Table 4.2 Actual results and ranking ($\hat{\theta} = 0.074$) for the top 20 finishers of race 28. 19
Table 4.3 UCI Mountain Bike World Cup cross-country points system. 20
Table 4.4 Top ten riders at the end of the 2014 season according to the UCI point-based ranking. 20

List of Figures

Figure 2.1 A histogram of the number of Elite competitor race finishes for the 2014 Mountain Bike World Cup season. 7

Figure 2.2 A histogram of the number of Canadian Elite competitor race finishes for the 2014 Mountain Bike World Cup season. 8

Figure 3.1 Probability density function of the Gumbel distribution with different parameters. 10

Figure 4.1 Scatterplot of weighting parameter $0 < \theta < 1$ and MAE and RMSE. The black line is a corresponding lowess curve. 17

Figure 4.2 Scatterplot of weighting parameter $0.0 < \theta < 0.1$ and MAE and RMSE. The black line is a corresponding lowess curve. 17

Figure 4.3 Scatterplot of weighting factor w_t and the number of races t prior to the most recent race when $\hat{\theta} = 0.074$. 17

Figure 4.4 Relationship between the number of top racers considered and the MAEs of the Bradley-Terry and point-based rankings. 21

Figure 4.5 Shiny user interface for selecting CSV input file. 22

Figure 4.6 Drop-down menus for choosing data filtering options. 23

Figure 4.7 Shiny user interface with data subset preview of the 2014 Elite World Cup season. 24

Figure 4.8 Progress bar informing the user of algorithm details. 24

Figure 4.9 Example of a Bradley-Terry prediction based on all seven Elite World Cup races of the 2014 season with weighting parameter $\theta = 0.074$. . 25

Chapter 1

Introduction

Researchers from Canadian Sport Institute Ontario (CSIO) approached us to discuss how we might be able to contribute to their statistical analysis efforts. CSIO is a national sports centre dedicated to putting more Canadian athletes on the international podium by applying sport science, sport medicine, athlete/coach and staff development. The institute is heavily invested in Canadian performance at Olympic, Paralympic Games and World Championships. For example, CSIO is a vital partner of the renowned Own the Podium organization.

1.1 Motivation

For the last few years, CSIO scientists have been involved with Cycling Canada, a national sport organization for the promotion of cycling in Canada. CSIO has been providing performance analysis support at various international competitions, such as the Mountain Bike World Cup held by Union Cycliste Internationale (UCI). This series is composed of a combination of cross-country and downhill disciplines, each with an Elite Men and Elite Women category. Athletes compete in several rounds during each season, with points awarded according to their placing in each event.

Much of Cycling Canada's recent focus has been on Canadian performance in the Elite cross-country events. While Canadian women routinely win or reach the podium, the men's podium has been dominated by the Swiss rider Nino Schurter and Frenchman Julien Absalon for the past 15 years. They share a combined 12 series winner titles from 2003 to 2017. The last Canadian man to win the cross-country series was Roland Green in 2001.

To assist in their cycling analytics endeavours, CSIO presented to us a specific problem. Their researchers and coaches received race data in a particular format not conducive to manipulation and inference. They were also unsure how to best utilize the available data to obtain insights. We were tasked to design a clean, intuitive user interface to perform rider ranking and predictive analyses "under the hood" for the men's cross-country events.

CSIO researchers did not believe that individual race times were noteworthy, and suggested that they should not form part of the analysis. Thus, the baseline model is a multinomial distribution on $n!$ permutations of race order, where n is the number of racers. However, the observations are sparse because races are few compared to the number of permutations of race order. For example, in a set of 10 races with 80 racers each, we only observe a maximum of 10 of the possible 80! orderings. There would be no observations for the other $80! - 10$ possible orderings, therefore we would have no idea of their probabilities. Furthermore, not every cyclist competes in every race. Therefore, some alternative is needed, such as a reduced model or data reduction.

1.2 Literature review

In recent years, many approaches have been implemented to perform ranking and prediction on data across a variety of sports [1, 4, 6, 7, 9]. The Bradley-Terry model [2] is one of the most commonly applied models, especially in match-based team sports such as basketball, where one team competes with only one other team at the same time [4]. In each paired-comparison, there is a clear winner and loser. The Bradley-Terry model treats the outcome of each comparison as an independent Bernoulli random variable.

While paired-comparison methods such as the Bradley-Terry model have been widely applied to produce rankings in team sports, their use has been limited in racing sports. This scarcity may perhaps be because paired-comparisons are not as obvious in racing sports as in team sports. Anderson [1] constructed a Bradley-Terry model to produce a ranking for Formula One racers, the highest class of single-seat auto racing, where he compared $\binom{n}{2}$ pairs per race. This may have been in error, as it is obvious that the outcomes of such $\binom{n}{2}$ paired-comparisons are not independent, which is a fundamental assumption of the Bradley-Terry model. This may explain why Anderson's findings expressed some unusual probabilistic results.

Due to the shortage of quality publications on Bradley-Terry implementation in racing sports, especially bicycling, the UCI Mountain Bike World Cup dataset presented a unique challenge. Most racing sports including Formula One and the Mountain Bike World Cup employ a point-based ranking system. That is, competitors accumulate points throughout a season based on their performances in individual races. Competitors and fans alike often misleadingly interpret these rankings as measures of ability.

In many racing sports, points are not awarded to the bottom half of the finishers. For example, in Formula One, points are only awarded to the top ten finishers in decreasing increments. The drivers out of the top ten do not receive any points, and would thus be ranked equally, which would not be representative of their true abilities. Further, not all competitors compete in all the races throughout a season. A point-based ranking system would favour mediocre but reliable performance over a few spectacular race outcomes. These are just a few of the many issues that may exist with interpreting point-based rankings as measures of ability.

1.3 Approach

The Bradley-Terry model remedies these issues by ranking competitors based on their estimated true abilities. In this project, we aimed to provide a reliable implementation for producing rankings in racing sports accessible to broader scientific audiences, with the intention of using the rankings for prediction in future races. We achieved this by creating a Shiny application that analyses a variety of inputs, allows user specification of certain parameters, and generates a Bradley-Terry ranking. The algorithms implemented include a basic Bradley-Terry model and an exponentially weighted version which allows for a user-defined weighting constant.

Consider a race with cyclists $i_1, i_2, \ldots, i_n$ where we are interested in the probability $P(i_1 > i_2 > \ldots > i_n)$. The probabilistic notation specifies the finishing order of the cyclists where $i_j > i_{j+1}$ denotes that cyclist j defeated cyclist $j+1$. In our Bradley-Terry implementation, we have $n-1$ comparisons that correspond to the probabilities $P(i_1 > i_2), P(i_2 > i_3), \ldots, P(i_{n-1} > i_n)$. Therefore, there is an implicit assumption that $P(i_1 > i_2 > \ldots > i_n) = P(i_1 > i_2) \times \cdots \times P(i_{n-1} > i_n)$. Similar to the assumption made by Anderson [1], our assumption is incorrect, but "not as incorrect" as Anderson [1]. The difficulty in our assumption can be seen by expressing

$$P(i_1 > i_2 > \ldots > i_n) = P(i_1 > i_2 \mid i_2 > i_3 > \ldots > i_n) P(i_2 > i_3 > \ldots > i_n).$$

It seems to us that $P(i_1 > i_2 \mid i_2 > i_3 > \ldots > i_n) \neq P(i_1 > i_2)$, which is the underlying premise of the Bradley-Terry model as paired outcomes are assumed independent.

We believe that $P(i_1 > i_2 \mid i_2 > i_3 > \ldots > i_n) \neq P(i_1 > i_2)$ since the condition $i_2 > i_3 > \ldots > i_n$ indicates that i_2 has performed very well in the race. We therefore expect $P(i_1 > i_2 \mid i_2 > i_3 > \ldots > i_n) < P(i_1 > i_2)$. The expected consequence of this assumption is that the resulting probabilities obtained by the Bradley-Terry estimation algorithm may be more extreme for the top-end racers in the positive sense and more extreme for the bottom-end racers in the negative sense. We do not plan on using existing Bradley-Terry inferential capabilities (e.g. the construction of confidence intervals). Rather, we view the Bradley-Terry algorithm applied to the $n-1$ comparisons as a black-box procedure that produces ranks. Later, we introduce a cross-validation procedure that tests the rankings in terms of future race prediction. This is done by training on a subset of the data to determine optimal weighting constants based on mean absolute error (MAE) and root mean squared error (RMSE).

Chapter 2 describes and provides further context for the Mountain Bike World Cup data. Chapter 3 outlines the statistical models. Chapter 4 presents predictions and results. Chapter 5 offers a brief discussion on findings and a discussion of future research.

Chapter 2

Data

CSIO researchers presented us with a dataset containing four years of UCI Mountain Bike World Cup races. The data spanned from 2011-2014 and covered 32 races. The data included race results for all the races held in the Elite, Junior, and U23 leagues in the cross-country discipline, totalling 7,625 rows. Other descriptors included race level, race date, race location, rider name, rider nationality, and UCI unique identifier code. As previously mentioned, CSIO did not believe that race times were important and excluded such information from the dataset.

After the end of every season, a special World Championship race is held. This race differs from World Cup season races in many aspects. For example, the competitors in the World Championships represent national teams rather than commercial teams. This leads to coaching, training, and strategical differences compared to World Cup races. In the Championship race, a wider range of riders are invited. This leads to the phenomenon where at least 10% of the riders in the Championship race have only participated in past championship races, and have never raced in any of the season races. Furthermore, the results of the Championship race do not count toward a racer's season point-based ranking. CSIO researchers were interested in World Cup ranking and prediction. Therefore, of the 32 races included in the dataset, only the 28 World Cup races were of analytical focus.

The data arrived from CSIO as an Excel spreadsheet (.xlsx) in a highly vertical format where each row only contained one race result for one racer (Table 2.1). Our algorithm converted the data from this redundant format into one that was more concise and conducive to analysis where each row contained all the race results for one racer (Table 2.2). If a racer did not compete in or finish a race, the entry was coded with a 0. The graphical user interface for the algorithm also provided users the ability to choose the specific league and races to be included in the data conversion. Four of the original columns were unnecessary for our analyses and were not included in the data format conversion.

The Mountain Bike World Cup season typically runs from April through August, with the number of races each season ranging from six to eight. The aforementioned World Championship race is held after each season, typically in September. The Mountain Bike

race_level	RACE_ DATE	RACE_ LOCATION	RIDER_ CLASS	GENDER	RIDER	FINAL_ RANK	COUNTRY	UCI_CODE
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Absalon Julien	1	France	FRA19800816
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Schurter Nino	2	Switzerland	SUI19860513
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Fontana Marco Aurelio	3	Italy	ITA19841012
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Milatz Moritz	4	Germany	GER19820624
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Fumic Manuel	5	Germany	GER19820330
World Championship	6-Sep-14	Hafjell, Norway	Elite	Men	Mantecon Gutierrez Sergio	6	Spain	ESP19840925

Table 2.1: Example of Excel data from CSIO.

Riders	13-Apr-14	27-Apr-14	24-May-14	31-May-14	3-Aug-14	10-Aug-14	24-Aug-14	6-Sep-14
Absalon Julien	1	1	5	1	2	2	2	1
Fumic Manuel	2	20	11	7	11	6	3	5
Marotte Maxime	3	3	15	9	42	18	15	17
Cink Ondrej	4	8	46	0	0	17	22	12
Hermida Ramos José Antonio	5	5	6	14	8	16	10	9
Schurter Nino	6	0	1	2	1	1	1	2

Table 2.2: Example of data format after conversion.

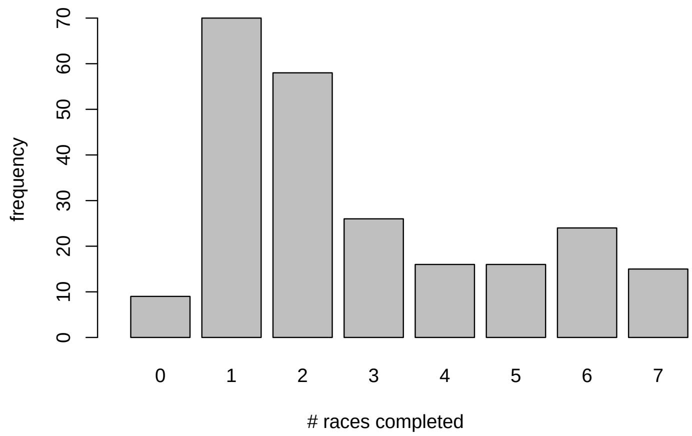

Figure 2.1: A histogram of the number of Elite competitor race finishes for the 2014 Mountain Bike World Cup season.

World Cup series is an international event with races being held in cities around the world. For example, the 2014 season featured races held in South Africa, Australia, Czech Republic, Germany, Canada, United States, France, and Norway. The wide variety of race factors including event elevation, climate, and surroundings that racers experience can have an impact on race performance.

Furthermore, not all riders compete and finish in every event of the season. Many do not participate in races due to injury, some are substitute riders racing in a single race, and others may not finish a race due to mechanical failures and punctures. Therefore, not all racers face each other for an equal number of races. For example, of the 233 Elite racers who competed in the 2014 season, 70 riders finished only one out of the seven races. Meanwhile, only 16 racers completed all seven races of the season, including the season champion Julien Absalon. The result is a heavily right-skewed distribution of race finishes (Figure 2.1). For comparison, 13 Canadian men competed in the 2014 season. The competitor with the most finishes was Raphael Gagne with six, while the majority of Canadian riders finished only two races (Figure 2.2).

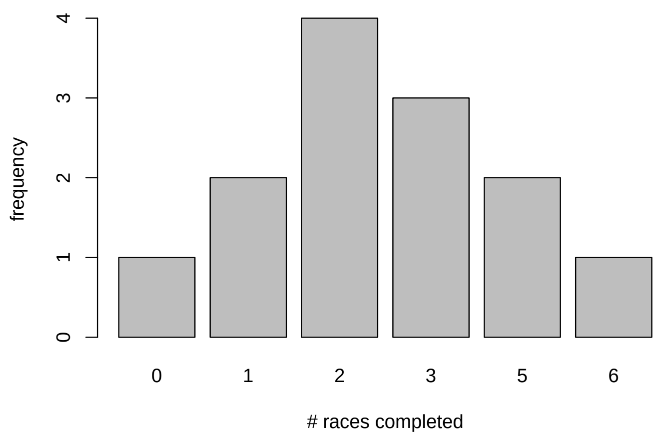

Figure 2.2: A histogram of the number of Canadian Elite competitor race finishes for the 2014 Mountain Bike World Cup season.

Chapter 3

Models

The estimation of competitors' true abilities was carried out via maximum likelihood estimation using the Bradley-Terry model.

3.1 Formulation

Suppose that competitor j possesses a latent ability score α_j representing their inherent "strength", which is assumed to be constant across the analysed races. The performance of rider j in race r is subject to a variety of stochastic factors ε_{rj} including but not limited to those mentioned in Chapter 2. The Bradley-Terry model assumes that the ε_{rj}'s are identically and independently distributed standard Gumbel random variables with cumulative distribution function

$$F(\varepsilon_{rj}) = \exp\left[-\exp\left(-\varepsilon_{rj}\right)\right] \qquad -\infty < \varepsilon_{rj} < \infty. \tag{3.1}$$

The standard Gumbel is a special case of the two parameter Gumbel distribution (also known as the log-Weibull distribution and the double exponential distribution). The Gumbel distribution has cumulative distribution function

$$F(x) = \exp\left[-\exp\left(-\left(x-\mu\right)/\beta\right)\right] \qquad -\infty < x < \infty,$$

where $-\infty < \mu < \infty$ and $\beta > 0$. Properties of the Gumbel distribution include the mode μ, the mean $\mu+\gamma\beta$ where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant, and standard deviation $\beta\pi/\sqrt{6}$. The standard Gumbel distribution corresponds to the case where $\mu = 0$ and $\beta = 1$. Figure 3.1 shows the density functions of various Gumbel distributions.

The resulting performance for rider j in race r can then be expressed as

$$m_{rj} = \alpha_j + \varepsilon_{rj}. \tag{3.2}$$

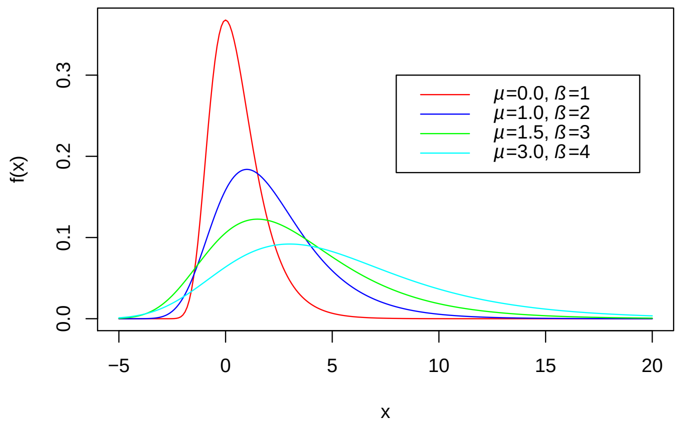

Figure 3.1: Probability density function of the Gumbel distribution with different parameters.

The continuous performance variable m_{rj} is latent (i.e. unobserved) and in the context of cycling could be interpreted as the time to complete the race or some transformation of time. Combining Equations (3.1) and (3.2), the distribution of m_{rj} is also a Gumbel random variable with distribution function

$$F(m_{rj}) = \exp\left[-\exp\left(-\left(m_{rj} - \alpha_j\right)\right)\right].$$

The difference in performance between riders j and k is given by $m_{rj} - m_{rk}$ which is simply a difference between two Gumbel random variables. The difference follows a logistic distribution with distribution function

$$F(m_{rj} - m_{rk}) = \frac{1}{1 + \exp\left[-\left(\left(m_{rj} - m_{rk}\right) - \left(\alpha_j - \alpha_k\right)\right)\right]}.$$

Thus, the probability that rider j beats k is

$$\begin{aligned} P(m_{rj} - m_{rk} > 0) &= 1 - \frac{1}{1 + \exp\left[-\left(\left(m_{rj} - m_{rk}\right) - \left(\alpha_j - \alpha_k\right)\right)\right]} \\ &= \frac{\exp(\alpha_j - \alpha_k)}{1 + \exp(\alpha_j - \alpha_k)} \\ &= \frac{\exp(\alpha_j)}{\exp(\alpha_j) + \exp(\alpha_k)}. \end{aligned}$$

The latent ability score could then be interpreted as a measure of the competitor's relative tendency of beating another competitor, averaged over a set of races. It can be observed here that the α_j's are not identifiable. That is,

$$\alpha_j' = \alpha_j + c \qquad \forall j$$

gives the same likelihood, where c is a constant. Therefore, the likelihood function for the Bradley-Terry model is

$$L(\alpha) = \prod_{(j,k)} \frac{\exp(\alpha_j)}{\exp(\alpha_j) + \exp(\alpha_k)} \tag{3.3}$$

where the product is taken over a set of paired-comparisons in race r. As previously discussed in Section 1.3, the set was chosen as the $n-1$ sequential race order comparisons.

3.2 Ability estimation

Estimation of the α's is then a maximum likelihood problem, where the values of α are sought that maximize the probability of the observed race outcomes

$$(i_1 > i_2), (i_2 > i_3), \ldots, (i_{n-1} > i_n).$$

Recall again that the sequential ordering of pairwise-comparisons implies an independence that may not be true. Many iterative optimization techniques may be employed to estimate these parameters [4, 6].

Here, we consider a simple approach. Let the probability that rider j beats k in race r be simplified as

$$p_{rjk} = P(m_{rj} - m_{rk} > 0) = \frac{\exp(\alpha_j)}{\exp(\alpha_j) + \exp(\alpha_k)}.$$

The Bradley-Terry likelihood (3.3) can then be expressed as

$$\begin{aligned} L(\alpha) &= \prod_{(j,k)} p_{rjk} \\ &= \prod_{i=1}^{n-1} (p_{rjk})^{y_i} (1 - p_{rjk})^{1-y_i}, \end{aligned} \tag{3.4}$$

where $\mathbf{y} = (y_1, \ldots, y_{n-1})^{\mathrm{T}}$ is a vector of 1's corresponding to the sequential paired comparisons. Notice that (3.4) is now identical to the likelihood function for logistic regression. This renders a convenient way to implement the algorithm in R with `glm` [1, 8, 9].

Specifically, estimation is achieved with `glm(Y~X-1, family=binomial(logit))`. Note that `X-1` denotes regression without the intercept term. Here, `Y` is a vector of ones $\mathbf{y}$, and `X` is a data matrix $\mathbf{X}$. This $\mathbf{X}$ matrix is defined such that each column corresponds to a racer $1, \ldots, n$ and each row corresponds to a paired-comparison. Each row records the outcome of one paired-comparison, assigning 1 to the winner and -1 to the loser, with zeroes everywhere else. The resulting matrix contains only two non-zero values in each row. Riders who failed to finish a race were assigned zeroes for all of their column values in that race. Finally, one competitor should be removed from the regression to act as a baseline. This is done by setting that racer's column values all to zeroes. The estimated ability α of this competitor would be zero, with all the other racers' abilities relative to the baseline rider. The difference of the estimated α's between riders are unaffected by the rider chosen as baseline.

To illustrate, consider a set of three races and $n = 5$ competitors A, B, C, D, and E in each race. Let the finishing order in the first race be A, B, C, D, E, the finishing order in the second race E, D, C, B, A, and in the third race A, C, E, B, with rider D failing to finish. Thus, the corresponding $\mathbf{X}$ would resemble

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{r=1} \\ \mathbf{X}_{r=2} \\ \mathbf{X}_{r=3} \end{bmatrix} = \begin{array}{c} \begin{array}{ccccc} \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{race 1} \\ \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \end{bmatrix} \text{race 2} \\ \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \text{race 3} \end{array} .$$

Note that baseline rider E is removed from the analysis by setting its column values to zero. Also note that $\mathbf{X}_{r=3}$ has three rows because only four riders finished the race.

The resulting estimated coefficients correspond to estimates of the α's and can be extracted with `model$coefficients`.

3.3 Weighted Bradley-Terry

In order to explore the time series effects on rider ability estimates, we developed an exponentially weighted version of the classical Bradley-Terry model. We believe that we are the first to do this in the context of Bradley-Terry models in racing sports. Let

$$w_t = \exp(-\theta t) \tag{3.5}$$

be the weighting factor, where $\theta \geq 0$ is a tuning parameter and

$$t = r_{\max} - r$$

is the number of races prior to the most recent one. Thus, the most recent race would correspond to $t = 0$ and have a weighting factor of $w_t = 1$. The weights w_t for $t > 0$ are decreasing in t. Note that although we have used the variable t to suggest time, the races are not necessarily evenly spaced and therefore t is not a true time variable.

This is analogous to the well-known technique of exponential smoothing for time series data, first introduced by Brown [3]. Exponential smoothing is often presented in time series applications as a geometric progression $1, (1-a), (1-a)^2, \ldots, (1-a)^{t_{\max}}$, where $0 < a < 1$ is the smoothing factor. This geometric series is practically a discrete version of the exponential decay function (3.5). The rationale behind exponential weights is the belief that older observations may be less relevant for prediction than more current ones.

Instead of assigning 1 and -1 respectively for the winner and loser of each paired-comparison at time t, the winner is assigned w_t and the loser $-w_t$. This regime assigns decrementing weights to races further back in time. If $\theta = 0$, $w_t = 1$ and the model returns to that of the classical Bradley-Terry model, weighting all races equally. Suppose $\theta = 0.1$. Then in a season with eight races, the weights would be $w_0, \ldots, w_7 = 1.00, 0.90, 0.82, 0.74, 0.67, 0.61, 0.55, 0.50$. That is, a win accumulated in the first race of the season would be worth half that of a win in the eighth race. Likewise, a loss would only have half the impact on the Bradley-Terry ranking compared to a loss in the eighth race. Now suppose $\theta = 1$, then $w_0, \ldots, w_7 = 1.000, 0.368, 0.135, 0.050, 0.018, 0.007, 0.002, 0.001$ which quickly leads to vanishing weights. Naturally, the next step is to determine an optimal value of θ.

To illustrate, consider again a set of three races and $n = 5$ competitors A, B, C, D, and E in each race. Let the finishing order in the first race be A, B, C, D, E, the finishing order in the second race E, D, C, B, A, and in the third race A, C, E, B, with rider D failing to finish. For weighting, let $\theta = 0.1$. Thus, the corresponding $\mathbf{X}$ would resemble

$$
\mathbf{X} = \begin{bmatrix} \mathbf{X}_{r=1} \\ \mathbf{X}_{r=2} \\ \mathbf{X}_{r=3} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{t=2} \\ \mathbf{X}_{t=1} \\ \mathbf{X}_{t=0} \end{bmatrix} = \begin{array}{c} \begin{array}{ccccc} \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \end{array} \\ \begin{bmatrix} 0.82 & -0.82 & 0 & 0 & 0 \\ 0 & 0.82 & -0.82 & 0 & 0 \\ 0 & 0 & 0.82 & -0.82 & 0 \\ 0 & 0 & 0 & 0.82 & 0 \end{bmatrix} \text{race 1} \\ \begin{bmatrix} 0 & 0 & 0 & -0.90 & 0 \\ 0 & 0 & -0.90 & 0.90 & 0 \\ 0 & -0.90 & 0.90 & 0 & 0 \\ -0.90 & 0.90 & 0 & 0 & 0 \end{bmatrix} \text{race 2} \\ \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \text{race 3} \end{array}
$$

Again, note that baseline rider E is removed from the analysis by setting its column values to zero. Also note that $\mathbf{X}_{r=3}$ has three rows because only four riders finished the race.

While the Bradley-Terry model assumes that competitor strengths are constant, our weighted approach addresses this constraint and recognizes that strengths may vary through time. We are also aware of other methods allowing for time-varying abilities such as the dynamic Bradley-Terry or Bayesian models where α_j's update after every race.

3.4 Cross-validation

An optimal value of θ can be determined through cross-validation. The two-step process involved estimating an optimal $\hat{\theta}$, then using $\hat{\theta}$ to validate performance.

Over the four years and 28 World Cup races worth of results, we determined an optimal $\hat{\theta}$ based on using races $1, \ldots, 26$ to predict race 27. Our analysis considered only racers who participated in the 27th race and also at least one of the first 26, i.e.

$$\{\text{analysed riders}\} = \{\text{riders in first 26 races}\} \cap \{\text{riders in 27th race}\}.$$

From the subset of intersecting riders, an $\mathbf{X}$ matrix was constructed based on the sequentially ordered paired-comparison outcomes from the first 26 races. Considering testing θ at 0.01 intervals from 0 to 1,

$$\theta_1, \ldots, \theta_{101} = 0.00, 0.01, \ldots, 0.99, 1.00.$$

We created 101 $\mathbf{X}$ matrices each with the corresponding weighting factors w_t as described in (3.5). Rankings were obtained by performing Bradley-Terry estimation on each of the $\mathbf{X}$'s. Then, riders were ranked according to their estimated strength parameters $\hat{\alpha}_j$.

For each θ, every rider was assigned a predicted ranking based on the Bradley-Terry ability estimation results. For each θ, rider j's predicted ranking $\hat{i}_j$ for the 27th race was compared with their actual ranking i_j in the 27th race. Model accuracy was assessed with the mean absolute error

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n} |i_j - \hat{i}_j|$$

and the root mean squared error

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(i_j - \hat{i}_j)^2}.$$

An optimal $\hat{\theta}$ was chosen that minimized both MAE and RMSE.

With the optimal $\hat{\theta}$, we validated the tuned model by using races $1, \ldots, 27$ to predict race 28. This time, our analysis considered only racers who participated in the 28th race and racers for whom rankings were predicted. Our procedure to exclude race 28 from estimation of $\hat{\theta}$ was based on the principle that data should not be used both for estimation and validation. Finally, model performance was measured with MAE and RMSE for validation.

Chapter 4

Predictions and results

Our approach provides a reliable implementation of Bradley-Terry ranking for Cycling Canada based on two stages. First, we determined an optimal value of θ through cross-validation, as described in Section 3.4. Then, we designed a Shiny application which provided a graphical user interface for producing predictions using cross-validation.

4.1 Cross-validation and predictions

Using races $1, \ldots, 26$ to predict race 27, the data matrix $\mathbf{X}$ had 997 rows and 89 columns. That is, our analysis considered the 89 racers who participated in the 27th race and also at least one of the first 26. Then, 997 sequential paired-comparisons were constructed based on the race results of the 89 racers. Testing θ at 0.01 intervals from 0 to 1, we constructed 101 $\mathbf{X}$ matrices with different θ's.

Performing Bradley-Terry estimation on each of the $\mathbf{X}$'s produced "weighted" rankings for each θ. Comparing the weighted rankings with the actual ranking in the 27th race, the MAE and RMSE were calculated for each θ. Figure 4.1 shows the relationship between θ and MAE and RMSE. There is some variation in the plots, but they are not representative of real trends. Therefore, a lowess curve with a bandwidth of 1/3 is applied as a smoother for visualization.

A priori, we did not know the range of θ leading to good predictions (i.e. small MAE and RMSE). It is apparent from Figure 4.1 that the interesting portion of the plots lies in the interval $[0.0, 0.1]$. We therefore repeated the cross-validation exercise testing 101 θ values equally spaced in the range $[0.0, 0.1]$. The resulting plots are given in Figure 4.2.

Based on the lowess curve, $\theta = 0.075$ produced the lowest MAE of 10.1 while $\theta = 0.073$ produced the lowest RMSE of 14.5 in predicting race 27. Therefore, the midpoint $\hat{\theta} = 0.074$ was determined to be the optimal value. This corresponds to weights $w_0, \ldots, w_{25} = 1.00, \ldots, 0.16$, as illustrated in Figure 4.3. The actual race results and predicted ranking at $\hat{\theta} = 0.074$ for the top 20 finishers in race 27 are shown in Table 4.1.

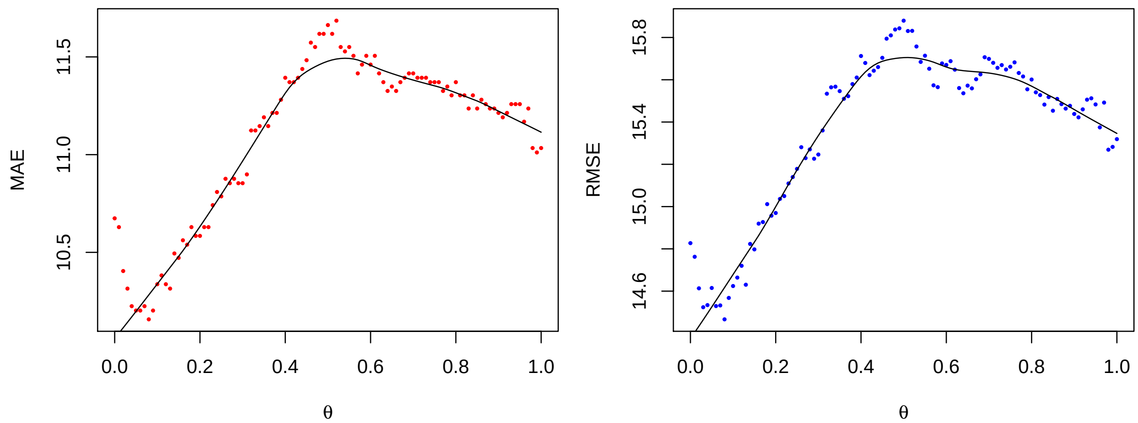

Figure 4.1: Scatterplot of weighting parameter $0 < \theta < 1$ and MAE and RMSE. The black line is a corresponding lowess curve.

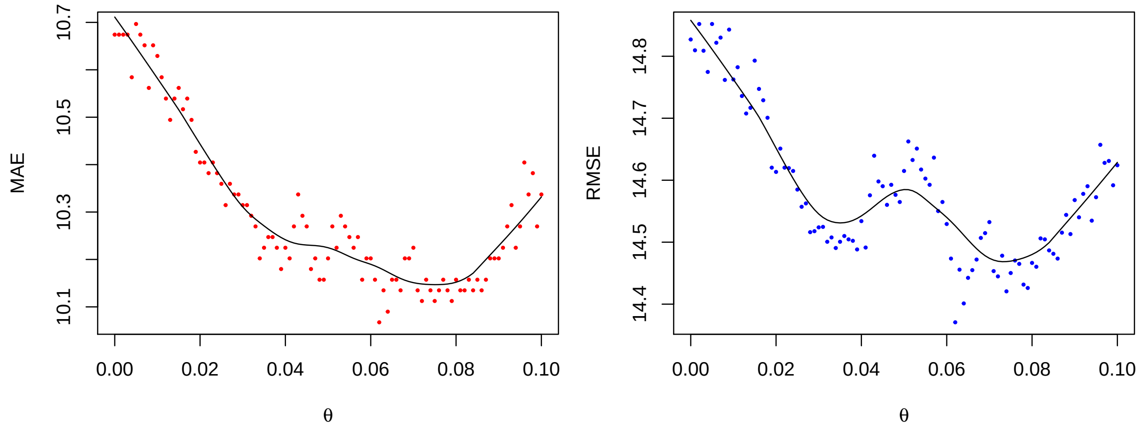

Figure 4.2: Scatterplot of weighting parameter $0.0 < \theta < 0.1$ and MAE and RMSE. The black line is a corresponding lowess curve.

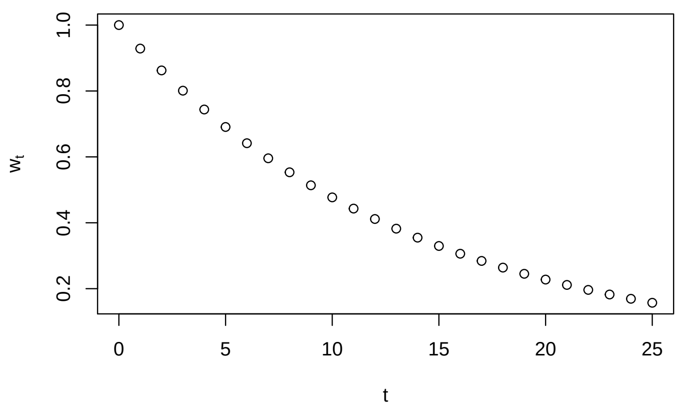

Figure 4.3: Scatterplot of weighting factor w_t and the number of races t prior to the most recent race when $\hat{\theta} = 0.074$.

Rider	Actual	Ranking $\hat{\theta} = 0.074$	Difference
Schurter Nino	1	1	0
Absalon Julien	2	2	0
Flückiger Lukas	3	12	-9
McConnell Daniel	4	6	-2
Flückiger Mathias	5	9	-4
Fumic Manuel	6	8	-2
Mantecon Gutierrez Sergio	7	13	-6
Tempier Stéphane	8	5	3
Vogel Florian	9	11	-2
Näf Ralph	10	17	-7
Giger Fabian	11	14	-3
Kabush Geoff	12	18	-6
Zandstra Derek	13	52	-39
Van Houts Rudi	14	20	-6
Fontana Marco Aurelio	15	16	-1
Hermida Ramos José Antonio	16	4	12
Cink Ondrej	17	7	10
Marotte Maxime	18	15	3
Lindgren Emil	19	19	0
Tiberi Andrea	20	23	-3

Table 4.1: Actual results and ranking ($\hat{\theta} = 0.074$) for the top 20 finishers of race 27.

With the optimal $\hat{\theta} = 0.074$, we continued the analysis by using races $1, \ldots, 27$ to predict race 28. The data matrix $\mathbf{X}$ had 1051 rows and 88 columns. That is, our analysis considered the 88 racers who participated in the 28th race and also in at least one of the first 27. Then, 1051 sequential paired-comparisons were constructed based on the race results of the 88 racers.

For validation, Bradley-Terry estimation was performed on the $\mathbf{X}$ with $\hat{\theta} = 0.074$. The predicted rankings (Table 4.2) produced MAE and RMSE of 12.4 and 18.0, respectively. These two values are higher than what was obtained in Figure 4.2, which emphasizes the importance of separating estimation from validation. The MAE of 12.4 corresponds to predictive ranking errors on average of around 12 positions. However, the MAE of the top 20 racers is only 6.6, which is considerably smaller than that of all 88 riders. The MAE further reduces to 4.3 when considering the top 10 racers. This is likely due to the stratification in true rider abilities. That is, the top riders' performances are much more consistent and superior than the rest of the field.

For example, it can be observed that Luca Braidot finished race 28 with a result (16) much higher than predicted (59). This may seem surprising at first, but taking a look at his race history provides some explanation. Braidot raced in the U23 league prior to the 2014 season, and performed well in the 2013 season with three top five finishes. Recently having been upgraded to the Elite league, his best finish was in 17th, but his consistency

Rider	Actual	Ranking $\hat{\theta} = 0.074$	Difference
Schurter Nino	1	1	0
Absalon Julien	2	2	0
Fumic Manuel	3	8	-5
Flückiger Lukas	4	5	-1
Kerschbaumer Gerhard	5	20	-15
McConnell Daniel	6	6	0
Näf Ralph	7	17	-10
Flückiger Mathias	8	4	4
Vogel Florian	9	10	-1
Hermida Ramos José Antonio	10	3	7
Lindgren Emil	11	21	-10
Milatz Moritz	12	15	-3
Giger Fabian	13	14	-1
Mantecon Gutierrez Sergio	14	11	3
Marotte Maxime	15	12	3
Braidot Luca	16	59	-43
Fanger Martin	17	24	-7
Fontana Marco Aurelio	18	13	5
Van Houts Rudi	19	18	1
Ettinger Stephen	20	32	-12

Table 4.2: Actual results and ranking ($\hat{\theta} = 0.074$) for the top 20 finishers of race 28.

suffered. In race 26 and 27 he finished in 105th and 50th, respectively. It is apparent that Braidot had the ability to finish in the top 20, but his recent races were disappointing. Due to the weighting of the Bradley-Terry estimation, his earlier good performances were downweighted compared to the recent poor outcomes.

4.2 Point-based ranking comparison

Further insight on field stratification may be gained by studying the UCI point-based ranking. As mentioned in Section 1.1, the series has long been dominated by the riders Nino Schurter and Julien Absalon. Their performances have been so commanding that according to the point-based system (Table 4.3), other racers in the top ten fail to even accumulate half of their points in a season (Table 4.4). For example, series winner Julien Absalon and runner-up Nino Schurter respectively accrued 1490 and 1330 points in the 2014 season. However, fifth place Stéphane Tempier came away with only 785 points. Rider points quickly dropped off from there.

However, as discussed in Section 1.2, many issues exist with interpreting the rankings as measures of athlete ability in point-based rankings of racing sports. Thus, predicting future performance based on point-based rankings may produce misleading results. The Bradley-Terry model remedies these concerns by ranking competitors based on their estimated true

Pos	Pts	Pos	Pts	Pos	Pts	Pos	Pts	Pos	Pts	Pos	Pts
1	250	11	90	21	64	31	44	41	27	51	17
2	200	12	85	22	62	32	42	42	26	52	16
3	160	13	80	23	60	33	40	43	25	53	15
4	150	14	78	24	58	34	38	44	24	54	14
5	140	15	76	25	56	35	36	45	23	55	13
6	130	16	74	26	54	36	34	46	22	56	12
7	120	17	72	27	52	37	32	47	21	57	11
8	110	18	70	28	50	38	30	48	20	58	10
9	100	19	68	29	48	39	29	49	19	59	9
10	95	20	66	30	46	40	28	50	18	60	8

Table 4.3: UCI Mountain Bike World Cup cross-country points system.

Pos	Athlete	Points
1	Julien Absalon	1490
2	Nino Schurter	1330
3	Daniel McConnell	970
4	Manuel Fumic	856
5	Stéphane Tempier	785
6	Mathias Fluckiger	785
7	José Antonio Hermida	767
8	Lukas Fluckiger	709
9	Sergio Mantecon Gutierrez	683
10	Maxime Marotte	668

Table 4.4: Top ten riders at the end of the 2014 season according to the UCI point-based ranking.

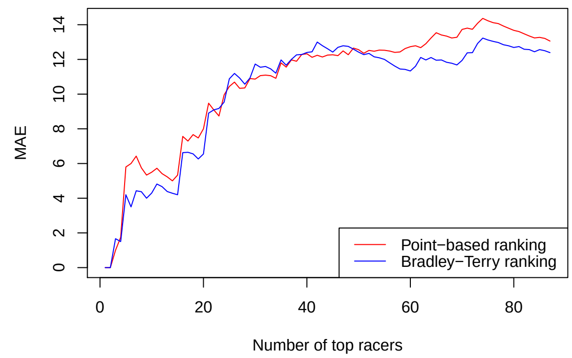

Figure 4.4: Relationship between the number of top racers considered and the MAEs of the Bradley-Terry and point-based rankings.

abilities. To demonstrate, we took the same 88 racers from the validation step and ranked them according to their UCI point-based ranking prior to race 28.

Prediction of the 28th race based on these rankings produced MAE of 13.1. The MAE was 8.0 for the top 20 racers and 5.5 for the top 10. Comparing with the results in Section 4.1, Bradley-Terry prediction of the same race produced MAE of 12.4. The Bradley-Terry prediction MAE was 6.6 for the top 20 racers and 4.3 for the top 10. Evidently, the Bradley-Terry model performed better in terms of lower MAE than the point-based rankings in predicting race 28 results, especially for the top racers. Figure 4.4 shows the relationship between the number of top racers considered and the MAEs of the Bradley-Terry and point-based rankings.

4.3 App for Cycling Canada

Shiny is an R package for building interactive apps for data analysis. It allows statistical computing to be performed behind the scenes in R, while interacting with broader scientific audiences through a graphical user interface. Therefore, the user need not understand the code nor model intricacies in order to use the app and benefit from the results.

The code for our app was written in a single .R script file with both the user interface and server components. The user interface allows users to specify a host of input parameters. To begin, the user would choose a CSV file from their computer (Figure 4.5). In the case of the Cycling Canada Excel spreadsheets, they had to be converted to CSV prior to selection, a very simple procedure.

As presented in Chapter 2, the dataset contained four years of race results for all the races held in the Elite, Junior, and U23 leagues in the cross-country discipline, totalling 7,625 rows. This includes both World Cup and World Championship races. If the file is successfully uploaded in the correct format, the selection screen expands to show further options as shown in Figure 4.6. The user could filter the dataset by league, race type, and

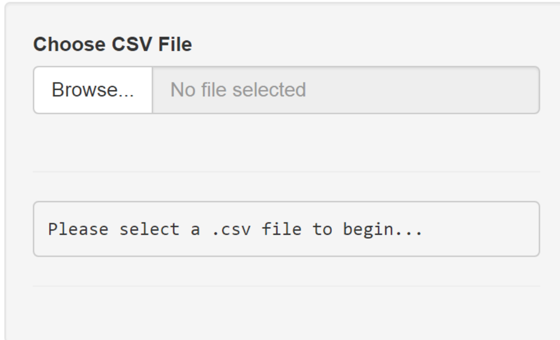

Figure 4.5: Shiny user interface for selecting CSV input file.

race date. The application automatically scans the dataset for league, race type, and race date values and presents them in drop-down menus.

If valid filters are applied, the interface (below the selection inputs) would show the number of riders and races in the selected data subset (Figure 4.7). These values could be used for user information or data verification. A preview of the subset is also presented to the right of the options bar. The data preview allows the user to further sort and search the subset for data verification.

For example, Figure 4.7 shows the user interface after selecting the 2014 World Cup season of the Elite league. It can be seen that the data summary shows 233 riders and seven races, matching that of the example in Chapter 2. This assures the user that the data was read correctly. Furthermore, the right side of the user interface shows a preview of the selected data for this subset.

Upon verifying the selected data, the user has the option of specifying a weighting parameter θ in the numeric entry field at the bottom of the options bar. The default value is set to 0, corresponding to the classical Bradley-Terry model, weighting all races equally. Of course, our cross-validation procedure recommends $\theta = 0.074$. Once satisfied with the chosen options, the user could press on the “Run Bradley-Terry Ranking” button and the application would perform the requested analysis in the background.

Depending on the size of the dataset to be analysed, analysis typically takes several seconds to a few minutes to complete. Throughout the process, a progress bar at the bottom right of the window provides details on the task being carried out (Figure 4.8). If for any reason the analysis needed to be stopped prior to completion, the user could simply press on the “X” in the progress bar to abort the process. Upon successful completion of the Bradley-Terry estimation, a table of predicted rankings is presented to the right of the options bar, replacing the data preview table (Figure 4.9). Similarly, this table is sortable and searchable. Note that some rider names have accented letters rendered incorrectly. This is an artefact originating from the Cycling Canada Excel file.

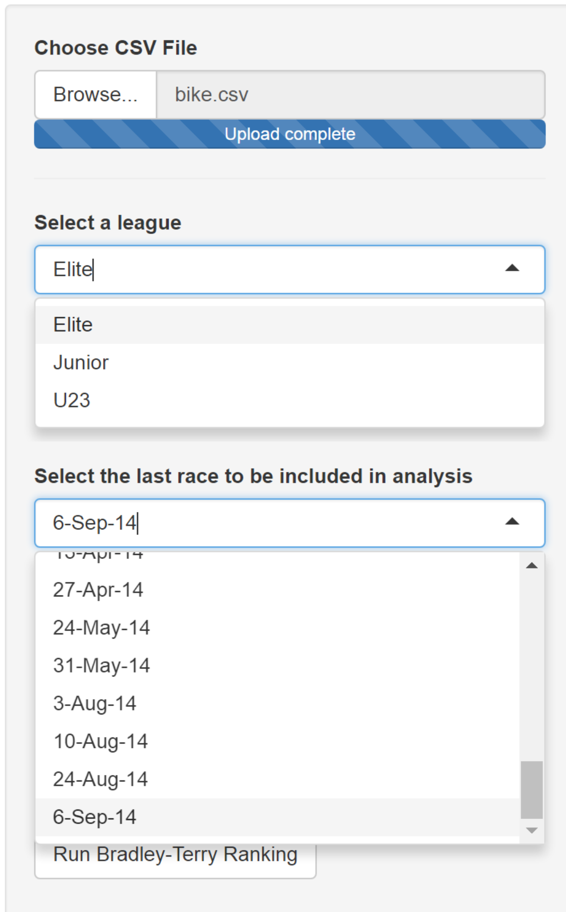

Figure 4.6: Drop-down menus for choosing data filtering options.

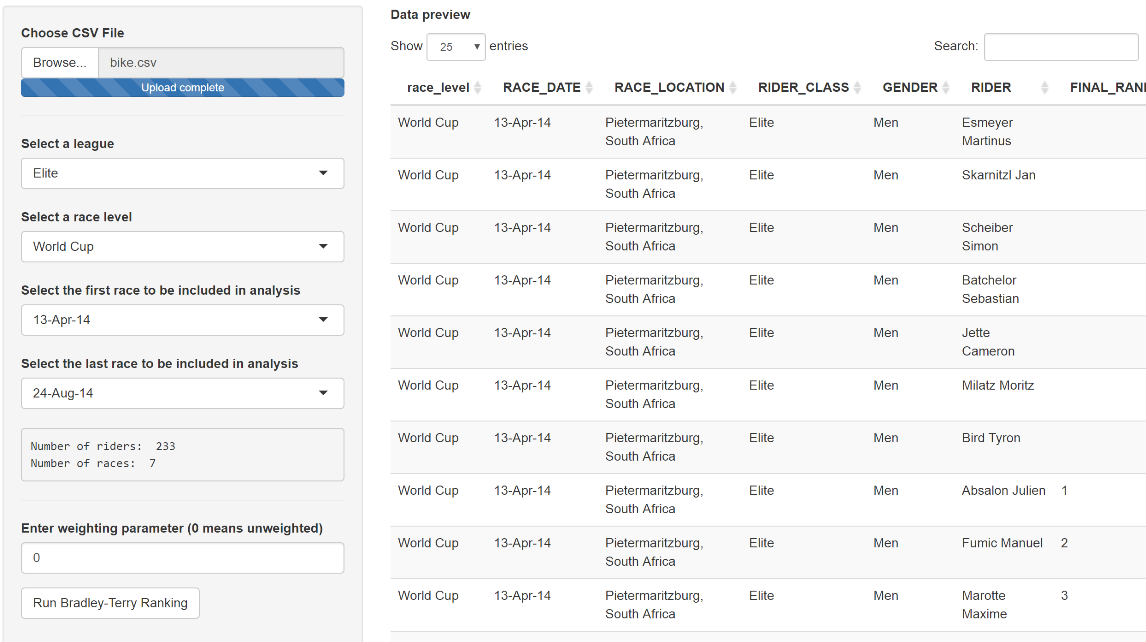

Figure 4.7: Shiny user interface with data subset preview of the 2014 Elite World Cup season.

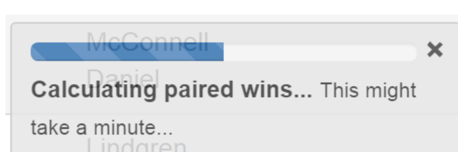

Figure 4.8: Progress bar informing the user of algorithm details.

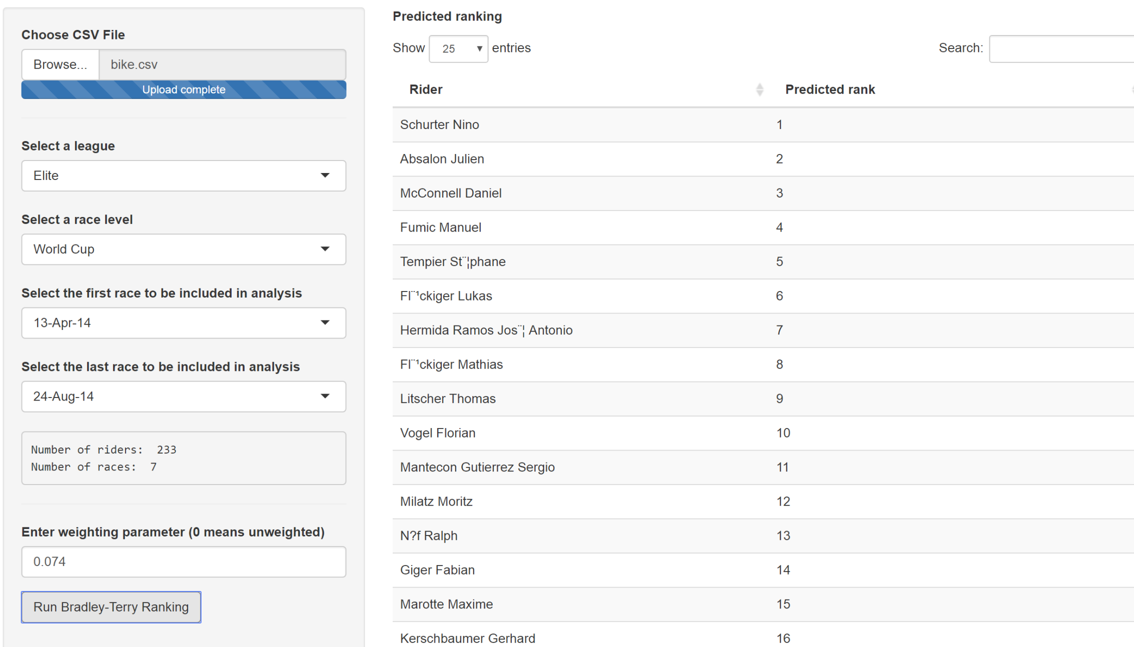

Figure 4.9: Example of a Bradley-Terry prediction based on all seven Elite World Cup races of the 2014 season with weighting parameter $\theta = 0.074$.

Chapter 5

Discussion

In our application of Bradley-Terry ranking for the Elite men's UCI Mountain Bike World Cup race data, we improved on the flawed implementation by Anderson [1] and we proposed a weighted version of the classical Bradley-Terry model. Combining these statistical techniques with a graphical user interface, we created a successful example of a user-friendly yet computationally versatile interface with Shiny.

Through cross-validation, we estimated an optimal weighting parameter of $\hat{\theta} = 0.074$. Our predicted rankings performed better in terms of predictive error than the UCI point-based ranking. Furthermore, the MAE of the Bradley-Terry estimation decreased as we considered the top racers. To coaches and sports scientists, the performance of these top racers is often of higher importance than the rest of the field.

As shown in this project, the Bradley-Terry model merits further research in the context of prediction in racing sports. Discussed in Section 1.3, our probabilistic assumption is incorrect, but "not as incorrect" as previous literature. Work may be done to further refine the probabilistic assumption, while not sacrificing computational ease. It may also be beneficial to develop alternative weighting schemes and compare their performance with the exponentially weighted Bradley-Terry.

For cross-validation, we split our data based on 28 races and measured accuracy with MAE and RMSE. The cross-validation process may benefit from using data from more races or implementing some other model evaluation metrics. It would also be possible to perform validation on more races, perhaps the final three races. The R code for cross-validation is included in Appendix A and can serve as a starting point for future development.

Mentioned in Section 3.3, we used the variable t to suggest time. The races are not necessarily evenly spaced and therefore t is not a true time variable. Further work can be done to implement t as a true time variable such as the number of days prior to the most recent race. Its performance can then be compared with that of the current weighting scheme. It might be reasonable to instead consider a time subscript α_{jt} on the ability score, but this greatly increases the parameter space and would make it difficult to obtain estimates.

Further, while CSIO researchers did not believe that individual race times were noteworthy for analysis, we believe that some form of race time analysis may produce interesting results. That is, racers who finish within a certain time difference δ of another may be treated as having tied. The Bradley-Terry model can be extended to accommodate such ties [5], and further cross-validation can be performed to determine an optimal δ.

Regarding the Shiny app, work can be done to increase its robustness in reading different data inputs. It would also be interesting if the user could choose a certain subset of riders they would like to be analysed. This would allow for predictions based on custom start lists and hypothetical scenarios. The R code for the Shiny app is included in Appendix B and can serve as a starting point for future development.

Bibliography

[1] A Anderson. Maximum likelihood ranking in racing sports. *Applied Economics*, 46(15): 1778–1787, 2014.

[2] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[3] Robert G Brown. Exponential smoothing for predicting demand, 1956.

[4] Manuela Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, pages 412–433, 2012.

[5] Roger R Davidson. On extending the bradley-terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

[6] David R Hunter. Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, pages 384–406, 2004.

[7] Dae-Ki Kang and Myong-Jong Kim. Poisson model and bradley terry model for predicting multiplayer online battle games. In *Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on*, pages 882–887. IEEE, 2015.

[8] Stan Lipovetsky and W Michael Conklin. Thurstone scaling via binary response regression. *Statistical Methodology*, 1(1):93–104, 2004.

[9] David Mease. A penalized maximum likelihood approach for the ranking of college football teams independent of victory margins. *The American Statistician*, 57(4):241–248, 2003.

Appendix A

Cross-validation code

```
options(useFancyQuotes = FALSE)
library(stringr)
library(hellno)
library(plotrix)

# Data directory
bike <- read.csv("C:/Users/Richard/Dropbox/Documents/SFU/Project/bike.csv",
                 header=TRUE,stringsAsFactors=FALSE)
# [1] "race_level"    "RACE_DATE"     "RACE_LOCATION" "RIDER_CLASS"   "GENDER"        "RIDER"
# [7] "FINAL_RANK"    "COUNTRY"       "UCI_CODE"

############################################################################################
# 1. DATA CLASSIFICATION ###################################################################
############################################################################################

### Append RACE_YEAR onto dataset
data <- cbind(as.numeric(paste("20",str_sub(bike$RACE_DATE,-2),sep="")),bike)
colnames(data)[1] <- "RACE_YEAR"
years <- as.numeric(paste("20",unique(str_sub(data$RACE_DATE,-2)),sep=""))

### Split data by RIDER_CLASS
elite <- data[data$RIDER_CLASS=="Elite",]; rownames(elite) <- NULL
junior <- data[data$RIDER_CLASS=="Junior",]; rownames(junior) <- NULL
U23 <- data[data$RIDER_CLASS=="U23",]; rownames(U23) <- NULL

### Unique riders in each class
elite.riders <- unique(elite$RIDER)
junior.riders <- unique(junior$RIDER)
U23.riders <- unique(U23$RIDER)

### Split each class by RACE_YEAR (automatic year detection)
for (i in years[1]:years[length(years)]){
  elite.temp <- paste("elite.",i,sep="")
  assign(elite.temp, data.frame(elite[elite$RACE_YEAR==i,],row.names=NULL))
  junior.temp <- paste("junior.",i,sep="")
  assign(junior.temp, data.frame(junior[junior$RACE_YEAR==i,],row.names=NULL))
  U23.temp <- paste("U23.",i,sep="")
  assign(U23.temp, data.frame(U23[U23$RACE_YEAR==i,],row.names=NULL))
  print(c(elite.temp,junior.temp,U23.temp))
}

# Data by class and year
# [1] "elite.2011"  "junior.2011" "U23.2011"
# [1] "elite.2012"  "junior.2012" "U23.2012"
# [1] "elite.2013"  "junior.2013" "U23.2013"
# [1] "elite.2014"  "junior.2014" "U23.2014"

### Race dates by class and year
for (i in years[1]:years[length(years)]){
  elite.temp <- paste("elite.",i,".races",sep="")
  assign(elite.temp, unique(elite[elite$RACE_YEAR==i,]$RACE_DATE))
  elite.races <- unique(elite$RACE_DATE)
  junior.temp <- paste("junior.",i,".races",sep="")
  assign(junior.temp, unique(junior[junior$RACE_YEAR==i,]$RACE_DATE))
  junior.races <- unique(junior$RACE_DATE)
  U23.temp <- paste("U23.",i,".races",sep="")
  assign(U23.temp, unique(U23[U23$RACE_YEAR==i,]$RACE_DATE))
  U23.races <- unique(U23$RACE_DATE)
  print(c(elite.temp,junior.temp,U23.temp))
}

# [1] "elite.2011.races"  "junior.2011.races" "U23.2011.races"
# [1] "elite.2012.races"  "junior.2012.races" "U23.2012.races"
```

```
# [1] "elite.2013.races"  "junior.2013.races" "U23.2013.races"
# [1] "elite.2014.races"  "junior.2014.races" "U23.2014.races"

### Unique riders in each class and year
for (i in years[1]:years[length(years)]){
  elite.temp <- paste("elite.",i,".riders",sep="")
  assign(elite.temp, unique(as.data.frame(lapply(paste("elite.",i,sep=""),get))$RIDER))
  junior.temp <- paste("junior.",i,".riders",sep="")
  assign(junior.temp, unique(as.data.frame(lapply(paste("junior.",i,sep=""),get))$RIDER))
  U23.temp <- paste("U23.",i,".riders",sep="")
  assign(U23.temp, unique(as.data.frame(lapply(paste("U23.",i,sep=""),get))$RIDER))
  print(c(elite.temp,junior.temp,U23.temp))
}

# Unique riders by class and year
# [1] "elite.2011.riders"  "junior.2011.riders" "U23.2011.riders"
# [1] "elite.2012.riders"  "junior.2012.riders" "U23.2012.riders"
# [1] "elite.2013.riders"  "junior.2013.riders" "U23.2013.riders"
# [1] "elite.2014.riders"  "junior.2014.riders" "U23.2014.riders"

###############################################################################################
# 2. DATA FORMATTING ##########################################################################
###############################################################################################

### Format all Elite races and riders into horizontal format
df.elite <- matrix(nrow=length(elite.riders),ncol=length(elite.races)+1)
df.elite[,1] <- elite.riders
df.elite <- as.data.frame(df.elite,stringsAsFactors=FALSE)

for (i in 1:length(elite.riders)){
  for (j in 1:length(elite.races)){
    if (length(elite[elite$RACE_DATE==elite.races[j]
                     & elite$RIDER == elite.riders[i],]$FINAL_RANK) > 0){
      df.elite[i,j+1] <- elite[elite$RACE_DATE==elite.races[j]
                               & elite$RIDER == elite.riders[i],]$FINAL_RANK
    }
  }
}

df.elite <- data.frame(cbind(data.frame(df.elite[,1]),
                             data.frame(lapply(df.elite[,2:dim(df.elite)[2]], as.numeric))))
colnames(df.elite) <- c("Riders",elite.races)

### Format all junior races and riders into horizontal format
df.junior <- matrix(nrow=length(junior.riders),ncol=length(junior.races)+1)
df.junior[,1] <- junior.riders
df.junior <- as.data.frame(df.junior,stringsAsFactors=FALSE)

for (i in 1:length(junior.riders)){
  for (j in 1:length(junior.races)){
    if (length(junior[junior$RACE_DATE==junior.races[j]
                      & junior$RIDER == junior.riders[i],]$FINAL_RANK) > 0){
      df.junior[i,j+1] <- junior[junior$RACE_DATE==junior.races[j]
                                 & junior$RIDER == junior.riders[i],]$FINAL_RANK
    }
  }
}

df.junior <- data.frame(cbind(data.frame(df.junior[,1]),
                              data.frame(lapply(df.junior[,2:dim(df.junior)[2]], as.numeric))))
colnames(df.junior) <- c("Riders",junior.races)

### Format all U23 races and riders into horizontal format
df.U23 <- matrix(nrow=length(U23.riders),ncol=length(U23.races)+1)
df.U23[,1] <- U23.riders
df.U23 <- as.data.frame(df.U23,stringsAsFactors=FALSE)

for (i in 1:length(U23.riders)){
  for (j in 1:length(U23.races)){
    if (length(U23[U23$RACE_DATE==U23.races[j]
                   & U23$RIDER == U23.riders[i],]$FINAL_RANK) > 0){
      df.U23[i,j+1] <- U23[U23$RACE_DATE==U23.races[j]
                           & U23$RIDER == U23.riders[i],]$FINAL_RANK
    }
  }
}

df.U23 <- data.frame(cbind(data.frame(df.U23[,1]),
                           data.frame(lapply(df.U23[,2:dim(df.U23)[2]], as.numeric))))
colnames(df.U23) <- c("Riders",U23.races)

### Checksums
sum(df.elite[,2:(length(elite.races)+1)],na.rm=T) == sum(elite$FINAL_RANK,na.rm=T)
sum(df.junior[,2:(length(junior.races)+1)],na.rm=T) == sum(junior$FINAL_RANK,na.rm=T)
sum(df.U23[,2:(length(U23.races)+1)],na.rm=T) == sum(U23$FINAL_RANK,na.rm=T)

###############################################################################################
# 3. CROSS-VALIDATION #########################################################################
###############################################################################################

# 0. DATA FORMATTING ##########################################################################
```

```
# options(warn=1)

raw <- rbind(elite)

data <- raw
riders <- unique(data$RIDER)
races <- unique(data$RACE_DATE)
data <- matrix(nrow=length(riders),ncol=length(races)+1)
data[,1] <- riders
data <- as.data.frame(data,stringsAsFactors=FALSE)

# Format race data horizontally
for (i in 1:length(riders)){
  for (j in 1:length(races)){
    if (length(raw[raw$RACE_DATE==races[j]
                   & raw$RIDER == riders[i],]$FINAL_RANK) > 0){
      data[i,j+1] <- raw[raw$RACE_DATE==races[j]
                         & raw$RIDER == riders[i],]$FINAL_RANK
    }
  }
}

data <- data.frame(cbind(data.frame(data[,1]),
                         data.frame(lapply(data[,2:dim(data)[2]],
                                           as.numeric))))
colnames(data) <- c("Riders",races)
data[is.na(data)] <- 0 # Set all NAs as 0

### 'data' is all races

# Checksum
sum(data[,2:(length(races)+1)],na.rm=T) == sum(elite[elite$RACE_YEAR <= 2014,]$FINAL_RANK,na.rm=T)

# Remove all world champsionship races
data <- cbind(data[,1:8], data[,10:17], data[,19:24], data[26:32])

# Subset test and validate sets

# race 27
Y_test <- data.frame(cbind(data[,1], as.numeric(data[,(dim(data)[2]-1)])),stringsAsFactors=F)
colnames(Y_test) <- c("Rider", "Actual")
order.Y_test <- order(as.numeric(Y_test$Actual), decreasing=F)
Y_test <- Y_test[order.Y_test,]
Y_test <- Y_test[Y_test$Actual!=0,]
rownames(Y_test) <- c()

# race 28
Y_val <- data.frame(cbind(data[,1], as.numeric(data[,dim(data)[2]])),stringsAsFactors=F)
colnames(Y_val) <- c("Rider", "Actual")
order.Y_val <- order(as.numeric(Y_val$Actual), decreasing=F)
Y_val <- Y_val[order.Y_val,]
Y_val <- Y_val[Y_val$Actual!=0,]
rownames(Y_val) <- c()

# race 1-26
X_test <- data
X_test <- X_test[,1:(dim(X_test)[2]-2)]

# Subset intersecting riders 1-26 and 27
X_test <- X_test[X_test$Riders %in% Y_test$Rider,]
rownames(X_test) <- c()
X_test <- X_test[which(rowSums(X_test[,2:dim(X_test)[2]])>0),]
rownames(X_test) <- c()

Y_test <- Y_test[Y_test$Rider %in% X_test$Riders,]
Y_test$Actual <- seq(1,dim(Y_test)[1])
rownames(Y_test) <- c()

test_riders <- length(unique(X_test$Riders))
test_pairs <- 0
test_races <- dim(X_test)[2]-1

# Calculate number of paired-comparisons 1-26
for (i in 1:test_races){
  test_pairs <- test_pairs + sum(as.numeric(X_test[,(i+1)])!=0) - 1
}

# race 1-27
X_val <- data
X_val <- X_val[,1:(dim(X_val)[2]-1)]

# Subset intersecting riders 1-27 and 28
X_val <- X_val[X_val$Riders %in% Y_val$Rider,]
rownames(X_val) <- c()
X_val <- X_val[which(rowSums(X_val[,2:dim(X_val)[2]])>0),]
rownames(X_val) <- c()

Y_val <- Y_val[Y_val$Rider %in% X_val$Riders,]
Y_val$Actual <- seq(1,dim(Y_val)[1])
rownames(Y_val) <- c()
```

```

val_riders <- length(unique(X_val$Riders))
val_pairs <- 0
val_races <- dim(X_val)[2]-1

# Calculate number of paired-comparisons 1-27
for (i in 1:val_races){
  val_pairs <- val_pairs + sum(as.numeric(X_val[,(i+1)])!=0) - 1
}

# 1. ESTIMATION ##########################################################################

# Predicted ranking 1-30
pred <- c()
for (w in seq(0,0.1,0.001)){
  # Calculate paired wins
  X <- matrix(0, nrow=test_pairs, ncol=test_riders)
  colnames(X) <- unique(X_test$Riders)
  X_row <- 1

  for (r in 1:test_races){
    places <- max(as.numeric(X_test[,(r+1)]))
    for (i in 1:(places-1)){
      win <- which(X_test[,r+1]==i)
      lose <- which(X_test[,r+1]==min(X_test[,r+1][X_test[,r+1]>i])) # next largest
      if (length(win)>0 & length(lose)>0){
        X[X_row, win] <- exp(-w*(test_races-r))
        X[X_row, lose] <- -exp(-w*(test_races-r))
        X_row <- X_row + 1
      }
    }
  }

  X[,dim(X)[2]] <- 0

  Y <- rep(1,test_pairs)
  model <- glm(Y~X-1, family=binomial(logit), control=glm.control(maxit=500))

  rank <- data.frame(summary(model)$coefficients[,1])
  rank <- data.frame(cbind(substr(rownames(rank),2,100), format(round(unname(rank),2),nsmall=2)))
  rownames(rank) <- c()
  colnames(rank) <- c("Rider", "Estimate")
  order.rank <- order(rank$Estimate,decreasing=TRUE)
  rank <- rank[order.rank,]
  #rank <- rank[rank$Rider %in% actual$Rider,] # remove riders from predicted not in actual
  rownames(rank) <- c()

  pred <- cbind(pred,as.character(rank[,1]))
  print(w)
}

pred <- cbind(seq(1,dim(pred)[1]),pred)
pred <- data.frame(pred)
colnames(pred) <- c("Place", seq(0,0.1,0.001))

# Comparison of actual with predicted rankings
comp <- Y_test
for (i in 1:length(seq(0,0.1,0.001))){
  pred.vec <- c()
  for (j in 1:length(Y_test$Rider)){
    if (length(pred[,1][pred[,(i+1)]==Y_test$Rider[j]])==0){
      pred.rank <- NA
      pred.vec <- c(pred.vec, pred.rank)
    } else {
      pred.rank <- pred[,1][pred[,(i+1)]==Y_test$Rider[j]]
      pred.vec <- c(pred.vec, pred.rank)}
  }
  comp <- cbind(comp, pred.vec)
}
colnames(comp) <- c("Rider", "Actual", seq(0,0.1,0.001))
comp <- comp[complete.cases(comp),]
rownames(comp) <- c()
comp_test <- comp#[1:45,]

# Calculate MAEs
mae <- c()
rmse <- c()
for (i in 1:length(seq(0,0.1,0.001))){
  err <- 1/dim(Y_test)[1] * sum(abs(as.numeric(as.character(comp_test[,2]))
                                    -as.numeric(as.character(comp_test[,(i+2)]))))
  mae <- c(mae, err)
  err2 <- sqrt(1/dim(Y_test)[1] * sum((as.numeric(as.character(comp_test[,2]))
                                       -as.numeric(as.character(comp_test[,(i+2)])))^2))
  rmse <- c(rmse, err2)
}

par(mfrow=c(1,2),mar=c(4,4,2,1))
plot(seq(0,0.1,0.001),mae,col="red",ylab="MAE",xlab=expression(theta),cex=0.5,pch=20)
mae_lowess <- lowess(seq(0,0.1,0.001),mae,f=1/3)
lines(mae_lowess)
plot(seq(0,0.1,0.001),rmse,col="blue",ylab="RMSE",xlab=expression(theta),cex=0.5,pch=20)
```

```r
rmse_lowess <- lowess(seq(0,0.1,0.001),rmse,f=1/3)
lines(rmse_lowess)

seq(0,0.1,0.001)[which(min(mae_lowess$y)==mae_lowess$y)]
seq(0,0.1,0.001)[which(min(rmse_lowess$y)==rmse_lowess$y)]
min(mae_lowess$y)
min(rmse_lowess$y)

# 2. VALIDATION ##############################################################################

# Predicted ranking 1-27
pred <- c()
for (w in seq(0,0.1,0.001)){
  # Calculate paired wins
  X <- matrix(0, nrow=val_pairs, ncol=val_riders)
  colnames(X) <- unique(X_val$Riders)
  X_row <- 1

  for (r in 1:val_races){
    places <- max(as.numeric(X_val[,(r+1)]))
    for (i in 1:(places-1)){
      win <- which(X_val[,r+1]==i)
      lose <- which(X_val[,r+1]==min(X_val[,r+1][X_val[,r+1]>i])) # next largest
      if (length(win)>0 & length(lose)>0){
        X[X_row, win] <- exp(-w*(val_races-r))
        X[X_row, lose] <- -exp(-w*(val_races-r))
        X_row <- X_row + 1
      }
    }
  }

  X[,dim(X)[2]] <- 0

  Y <- rep(1,val_pairs)
  model <- glm(Y~X-1, family=binomial(logit), control=glm.control(maxit=500))

  rank <- data.frame(summary(model)$coefficients[,1])
  rank <- data.frame(cbind(substr(rownames(rank),2,100), format(round(unname(rank),2),nsmall=2)))
  rownames(rank) <- c()
  colnames(rank) <- c("Rider", "Estimate")
  order.rank <- order(rank$Estimate,decreasing=TRUE)
  rank <- rank[order.rank,]
  #rank <- rank[rank$Rider %in% actual$Rider,] # remove riders from predicted not in actual
  rownames(rank) <- c()

  pred <- cbind(pred,as.character(rank[,1]))
  print(w)
}

pred <- cbind(seq(1,dim(pred)[1]),pred)
pred <- data.frame(pred)
colnames(pred) <- c("Place", seq(0,0.1,0.001))

# Comparison of actual with predicted rankings
comp <- Y_val
for (i in 1:length(seq(0,0.1,0.001))){
  pred.vec <- c()
  for (j in 1:length(Y_val$Rider)){
    if (length(pred[,1][pred[,(i+1)]==Y_val$Rider[j]])==0){
      pred.rank <- NA
      pred.vec <- c(pred.vec, pred.rank)
    } else {
      pred.rank <- pred[,1][pred[,(i+1)]==Y_val$Rider[j]]
      pred.vec <- c(pred.vec, pred.rank)}
  }
  comp <- cbind(comp, pred.vec)
}
colnames(comp) <- c("Rider", "Actual", seq(0,0.1,0.001))
comp <- comp[complete.cases(comp),]
rownames(comp) <- c()
comp_val <- comp#[1:10,]

# Calculate MAEs
mae <- c()
rmse <- c()
for (i in 1:length(seq(0,0.1,0.001))){
  err <- 1/dim(comp_val)[1] * sum(abs(as.numeric(as.character(comp_val[,2]))
                              -as.numeric(as.character(comp_val[,(i+2)]))))
  mae <- c(mae, err)
  err2 <- sqrt(1/dim(comp_val)[1] * sum((as.numeric(as.character(comp_val[,2]))
                                  -as.numeric(as.character(comp_val[,(i+2)])))^2))
  rmse <- c(rmse, err2)
}

par(mfrow=c(2,1),mar=c(4,4,2,1))
plot(seq(0,0.1,0.001),mae,col="red",ylab="MAE",xlab=expression(theta),cex=0.5,pch=20)
lines(lowess(seq(0,0.1,0.001),mae,f=1/3))
plot(seq(0,0.1,0.001),rmse,col="blue",ylab="RMSE",xlab=expression(theta),cex=0.5,pch=20)
lines(lowess(seq(0,0.1,0.001),rmse,f=1/3))

seq(0,0.1,0.001)[which(min(mae)==mae)]
```

```
seq(0,0.1,0.001)[which(min(rmse)==rmse)]
min(mae)
min(rmse)

mae[length(seq(0,0.074,0.001))]
rmse[length(seq(0,0.074,0.001))]
```

Appendix B

Shiny app code

```
options(useFancyQuotes = FALSE)
library(stringr)
library(shiny)

# Define UI for data upload app ----
ui <- fluidPage(
  # App title ----
  titlePanel("Bradley-Terry Ranking v2.2"),
  # Sidebar layout with input and output definitions ----
  sidebarLayout(
    # Sidebar panel for inputs ----
    sidebarPanel(
      # Input: Select a file ----
      fileInput("file1", "Choose CSV File",
                accept = c("text/csv",
                           "text/comma-separated-values,text/plain",
                           ".csv")),
      # Horizontal line ----
      tags$hr(),
      uiOutput("leaguebox"),
      uiOutput("levelbox"),
      uiOutput("startrace"),
      uiOutput("endrace"),
      verbatimTextOutput("dimensions"),
      tags$hr(),
      uiOutput("theta"),
      uiOutput("runbutton"),
      p(),
      verbatimTextOutput(("process"))
    ),
    # Main panel for displaying outputs ----
    mainPanel(
      uiOutput("mainHeader"),
      # Output: Data file ----
      dataTableOutput("contents")
    )
  )
)

# Define server logic to read selected file ----
server <- function(input,output,session) {
  df <- reactive({
    data <- read.csv(input$file1$datapath, header = TRUE, stringsAsFactors = FALSE)
    bike <- cbind(as.numeric(paste("20",str_sub(data$RACE_DATE,-2),sep="")),data)
    colnames(bike)[1] <- "RACE_YEAR"
    bike
  })

  output$mainHeader <- renderUI({
    if((flag$x)!=1){
      p(strong("Data preview"))
    } else {
      p(strong("Predicted ranking"))
    }
  })

  flag <- reactiveValues(x=0,ranking=NULL)

  output$contents <- renderDataTable({
    if(is.null(input$file1)) {
      return(NULL)
    } else if (is.null(input$file1)==FALSE & flag$x!=1){
      return(filtered()[,2:dim(filtered())[2]])
```

```
    } else if (flag$x==1){
      return(flag$ranking)
    }
  })

  output$leaguebox <- renderUI({
    if(is.null(input$file1)) return(NULL)
    selectInput("league","Select a league",unique(df()$RIDER_CLASS),"pick one")
  })

  output$levelbox <- renderUI({
    if(is.null(input$file1)) return(NULL)
    selectInput("level","Select a race level",unique(df()$race_level),"pick one")
  })

  output$startrace <- renderUI({
    if(is.null(input$file1)) return(NULL)
    selectInput("first","Select the first race to be included in analysis",
                unique(df()[df()$RIDER_CLASS==input$league & df()$race_level==input$level,]
                       $RACE_DATE),"pick one")
  })

  output$endrace <- renderUI({
    if(is.null(input$file1)) return(NULL)
    selectInput("last","Select the last race to be included in analysis",
                unique(df()[df()$RIDER_CLASS==input$league & df()$race_level==input$level,]
                       $RACE_DATE),"pick one")
  })

  filtered <- reactive({
    temp <- data.frame(matrix(0,ncol=dim(df())[2]),stringsAsFactors = FALSE)
    colnames(temp) <- colnames(df())
    for (i in match(input$first,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE)):
         match(input$last,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE))){
      temp <- rbind(temp, df()[df()$RIDER_CLASS==input$league
                       & df()$RACE_DATE==unique(df()[df()$RIDER_CLASS==input$league,]
                                                $RACE_DATE)[i],])
    }
    temp <- temp[-1,]
    temp <- temp[temp$race_level==input$level,]
  })

  riders <- reactive({
    unique(filtered()$RIDER)
  })

  numRiders <- reactive({
    length(riders())
  })

  races <- reactive({
    unique(filtered()$RACE_DATE)
  })

  numRaces <- reactive ({
    length(races())
  })

  output$dimensions <- renderPrint({
    if(is.null(input$file1)) return(cat("Please select a .csv file to begin..."))
    if(match(input$first,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE)) >
       match(input$last,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE))){
      return(cat("You have selected a start date after \nthe end date, please revise to continue..."))
    }
    cat("Number of riders: ", numRiders(), "\nNumber of races: ", numRaces())
  })

  output$theta <- renderUI({
    if(is.null(input$file1)) return(NULL)
    if(match(input$first,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE)) >
       match(input$last,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE))){
      return(NULL)
    }
    #if(is.null(input$runtype)) return(NULL)
    numericInput("theta","Enter weighting parameter (0 means unweighted)",value=0,min=0,step=0.01)
  })

  output$runbutton <- renderUI({
    if(is.null(input$file1)) return(NULL)
    if(match(input$first,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE)) >
       match(input$last,unique(df()[df()$RIDER_CLASS==input$league,]$RACE_DATE))){
      return(NULL)
    }
    #if(is.null(input$runtype)) return(NULL)
    actionButton("runbt","Run Bradley-Terry Ranking")
  })

  observeEvent(input$runbt, {
      progress <- Progress$new(session, min=0, max=10)
      on.exit(progress$close())

```

```
progress$set(message = "Initializing...",value=1)

df.run <- matrix(nrow=numRiders(),ncol=numRaces()+1)
df.run[,1] <- riders()
df.run <- as.data.frame(df.run,stringsAsFactors=FALSE)

progress$set(message = "Formatting...",value=2)

# Converting to horizontal format
for (i in 1:numRiders()){
  for (j in 1:numRaces()){
    if (length(filtered()[filtered()$RACE_DATE==races()[j]
                     & filtered()$RIDER == riders()[i],]$FINAL_RANK) > 0){
      df.run[i,j+1] <- filtered()[filtered()$RACE_DATE==races()[j]
                             & filtered()$RIDER == riders()[i],]$FINAL_RANK
    }
  }
}
df.run <- data.frame(cbind(data.frame(df.run[,1]),
                           data.frame(lapply(df.run[,2:dim(df.run)[2]], as.numeric))))
colnames(df.run) <- c("Riders",races())
df.run[is.na(df.run)] <- 0

# Conversion checksum
checksum <- sum(df.run[,2:(numRaces()+1)],na.rm=T) == sum(filtered()$FINAL_RANK,na.rm=T)
if (checksum==TRUE){
  progress$set(message = "Checksum passed...",value=3)
}
try (if (checksum == FALSE){
  output$process <- renderPrint({
    cat("Starting...\nFormatting...\nChecksum failed...\nreview data file or contact admin.")
  })
  stop()
})

# Subset test and validate sets
progress$set(message = "Calculating pairs...",
             detail="This usually takes several seconds...",
             value=4)

X_test <- df.run
test_riders <- length(unique(X_test$Riders))
test_pairs <- 0
test_races <- dim(X_test)[2]-1

# Calculate number of paired-comparisons
for (i in 1:test_races){
  test_pairs <- test_pairs + sum(as.numeric(X_test[,(i+1)])!=0) - 1
}

# Begin B-T estimation
X <- matrix(0, nrow=test_pairs, ncol=test_riders)
colnames(X) <- unique(X_test$Riders)
X_row <- 1


# Creating data Matrix X
progress$set(message = "Calculating paired-comparison outcomes...",
             detail="This might take a minute...",
             value=5)

for (r in 1:test_races){
  places <- max(as.numeric(X_test[,(r+1)]))
  for (i in 1:(places-1)){
    win <- which(X_test[,r+1]==i)
    lose <- which(X_test[,r+1]==min(X_test[,r+1][X_test[,r+1]>i])) # next largest
    if (length(win)>0 & length(lose)>0){
      X[X_row, win] <- exp(-input$theta*(test_races-r))
      X[X_row, lose] <- -exp(-input$theta*(test_races-r))
      X_row <- X_row + 1
    }
  }
}

X[,dim(X)[2]] <- 0
# Calculate paired wins
progress$set(message = "Fitting model...",
             detail="This takes the longest...",
             value=6)

Y <- rep(1,test_pairs)
model <- glm(Y~X-1, family=binomial(logit), control=glm.control(maxit=500))

progress$set(message = "Extracting ability estimates...",
             value=9)

rank <- data.frame(summary(model)$coefficients[,1])
rank <- data.frame(cbind(substr(rownames(rank),2,100), format(round(unname(rank),2),nsmall=2)))
rownames(rank) <- c()
colnames(rank) <- c("Rider", "Estimate")
order.rank <- order(rank$Estimate,decreasing=TRUE)
```

```
      rank <- rank[order.rank,]
      rownames(rank) <- c()
      print(head(rank))
      show <- data.frame(rank$Rider,rownames(rank))
      colnames(show) <- c("Rider", "Predicted rank")
      flag$ranking <- show
      flag$x <- 1
  })
}

# Create Shiny app ----
shinyApp(ui, server)
```