Neural Comput & Applic (2008) 17:175–186
DOI 10.1007/s00521-006-0080-8

ORIGINAL ARTICLE

A Bradley–Terry artificial neural network model for individual ratings in group competitions

Joshua E. Menke · Tony R. Martinez

Received: 5 June 2006 / Accepted: 10 November 2006 / Published online: 11 January 2007

Abstract A common statistical model for paired comparisons is the Bradley–Terry model. This research re-parameterizes the Bradley–Terry model as a single-layer *artificial neural network* (ANN) and shows how it can be fitted using the delta rule. The ANN model is appealing because it makes using and extending the Bradley–Terry model accessible to a broader community. It also leads to natural incremental and iterative updating methods. Several extensions are presented that allow the ANN model to learn to predict the outcome of complex, uneven two-team group competitions by rating individuals—no other published model currently does this. An incremental-learning Bradley–Terry ANN yields a probability estimate within less than 5% of the actual value training over 3,379 multi-player online matches of a popular team- and objective-based first-person shooter.

Keywords Bradley–Terry model · Paired comparisons · Neural networks · Delta rule · Probability estimates

1 Introduction

The Bradley–Terry model is well known for its use in statistics for paired comparisons [1–4]. It has also been applied in machine learning to obtain multi-class probability estimates [5–7]. The original model states that:

$$Pr(A \text{ defeats } B) = \frac{\lambda_A}{\lambda_A + \lambda_B}, \tag{1}$$

where λ_A and λ_B are both positive and subjects A and B compete in a paired-comparison. λ_A and λ_A represent the strengths of subjects A and B.

Bradley–Terry maximum likelihood models are often fit using methods like Markov Chain Monte Carlo (MCMC) integration or fixed-point algorithms that seek the mode of the negative log-likelihood function [4]. These methods however, ''are inadequate for large populations of competitors because the computation becomes intractable.'' [8]. The following paper presents a method for determining the ratings of over 4,000 players over 3,379 competitions.

Elo [9], a chess master and physics professor, suggested an efficient approach to update the ratings of thousands of players competing in thousands of chess tournaments. Now commonly referred to as the ELO Rating System, it has been used in the past by both the United States Chess Federation (USCF) and the World Chess Federation (FIDE). Most common large-scale rating systems in use today have roots in the ELO system. His method re-parameterizes the Bradley–Terry model by setting $\lambda_A = 10^{\theta_A}$ yielding:

$$Pr(A \text{ defeats } B) = \frac{1}{1 + 10^{-\frac{\theta_A - \theta_B}{400}}}. \tag{2}$$

The scale specific parameters are historical only and can be changed to the following which uses the natural base instead:

J. E. Menke (✉) · T. R. Martinez
Computer Science Department, Brigham Young University, 3361 TMCB, Provo, UT 84604, USA
e-mail: josh@axon.cs.byu.edu

$$Pr(A \text{ defeats } B) = \frac{1}{1 + e^{-(\theta_A - \theta_B)}}. \quad (3)$$

This is also the equation for the standard sigmoid used as a transfer function in *artificial neural network* (ANN) nodes, and therefore substituting w for θ in (3) yields the single-layer ANN in Fig. 1. This single-layer sigmoid node ANN can be viewed as a Bradley–Terry model where the input corresponding to subject A is always 1, and B, always –1. The weights w_A and w_B correspond to the Bradley–Terry strengths θ_A and θ_B—often referred to as ratings. The Bradley–Terry ANN model can be fit by using the standard delta rule training method for single-layer ANNs.

This ANN interpretation is appealing because many types of common extensions can be added simply as additional inputs to the ANN. It also makes the Bradley–Terry model accessible to anyone familiar with ANNs and provides natural incremental as well as iterative training methods (see Sect. 3). Bradley–Terry models are useful because they can be applied to any situation where the probability of one subject being ''better'' than another is needed. This can include not only sports, but obtaining multi-class probability estimates from binary classifiers in machine learning [5,1000,7], market predictions for economics, biostatistics [1], bibliometrics or deciding which publications are more significant [10], genetics [11], taste-testing, military encounters, and any other prediction that requires the probability of one subject being more desirable than another. In addition to making the usage of the Bradley–Terry model more accessible to the machine learning community, viewing the Bradley–Terry model as an ANN also makes research into extending and improving the model more accessible. This research demonstrates this by adding and proposing several extensions based on this ANN view of the Bradley–Terry model.

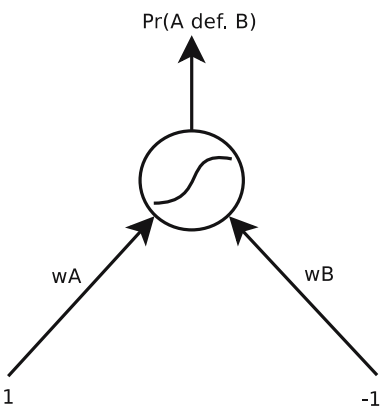

Fig. 1 The Bradley–Terry model as a single-layer ANN

The first extension presented is actually a generalization of the original Bradley–Terry model which provides for the case where the subjects being compared are groups and the goal is to obtain the ratings of the individuals in the group. The original model does not do this. The following presents a model that can give individual ratings from groups and shows further extensions of the model to handle ''home field advantage'', weighting individuals by contribution, dealing with variable-sized groups, and other issues. The probability estimates given by the final model are accurate to within less than 5% of the actual observed values over 3,379 multi-player online matches of a popular team- and objective-based first-person shooter. This is significant because of the difficulty in the predicting the outcome of these highly dynamic matches. They can be compared to a soccer game where either team can have as many players as they like so long as the combined number of players on the field is less than some maximum. To complicate matters further, the players are free to switch between teams at will. Furthermore, the soccer field does not have to be symmetrical: the field can be on an incline such that the ball naturally rolls towards one team's goal, and it can also include obstacles on one end that are not on the other. The pool of possible players also does not have a limit, ranging as high as 4,000, and the players are free to leave and join a match at any time. There are no other published models designed to handle this complex of a competition, and therefore nothing with which to compare the current results. The fact that the proposed model provides probability estimates within less than 5% of a match's actual outcome and that these estimates can be created in a short amount of time is in and of itself a significant result.

One other extension that will be presented is for dealing with the effects of the passage of time within a competition. There exist competitions and paired-comparisons in which the probability of a given competitor defeating the other is related to the amount of time that has passed in the competition. For instance, in chess, there may be players that win often when they are able to defeat their opponents early in the match, but who do not win as often when the match lasts longer than average. In the multi-player online game used for the results in this paper, one team is usually trying to prevent the other from achieving a known objective for a certain amount of time. When the objective is not accomplished in the usual amount of time, the probability of the achieving team winning is likely to be less than expected. Games and sports are not the only applications where the passage is significant. Modeling time in paired-comparisons can also be applied in education to determine how time affects

different teaching and testing methods, it can be applied in medicine to account for the passage of time given different treatments, it can be used in military applications for balancing exercises or determining how long a given encounter should last, and it can be used in any other paired-comparison or competition where time may play an important role. One of the extensions in the following paper uses an artificial neural network approach to learn how important the passage of time is for a given paired-comparison. Including this extension results in further a 16% decrease in error over the 3,379 matches.

This article will proceed as follows: Sect. 2 gives more related background on prior uses and methods for the fitting of Bradley–Terry models, Sect. 3 explains the proposed model, extensions, and how to fit it, Sect. 4 explains how the model is evaluated, Sect. 5 gives results and analysis of the model's evaluations, a further analysis of the meaning of the time weights is given in Sect. 7, a discussion of possible applications of the model are given in Sect. 6 and Sect. 8 gives conclusions and future research directions.

2 Background

The Bradley–Terry model was originally introduced by Bradley and Terry [1]. It has been widely applied in situations where the strength of a particular choice or individual needs to be evaluated with respect to another. This includes sports and other competitive applications. There are several reviews on extensions and methods of fitting Bradley–Terry models [1–4]. This paper differs from previous research in at least two ways. First, it proposes a method to determine individual ratings and rankings when the subjects being compared are groups. Recently, Huang et al. proposed similar extensions by modeling group ratings as the sum of the ratings of the individuals in each group [7, 12]. One of their papers models the increase in a larger group's rating linearly [7] and the other uses a similar model to that used here to model that increase exponentially [12]. We model group ratings as an exponentiated average of the individual ratings in a group, along with an exponentiated ''field'' weight that takes into account the number of individuals in each group. Like Huang et al.'s second model, ours assumes having more individuals in one group than another results in an exponential increase in the larger group's probability of winning. This is appealing in competitions where each individuals can act simultaneously.

The more common methods for fitting a Bradley–Terry model require that all of the comparisons to be considered involving all of the competitors have already taken place so the model can be fit simultaneously. This includes two different iterative methods proposed Huang et al. [7, 12]. While this can be more theoretically accurate, it becomes computationally intensive in situations involving thousands of comparisons and competitors. It would require storing and retaining information about every match that ever occurs and refitting the model after each match. This can be unreasonable if the system is meant to continually processes a potentially unlimited amount of matches. In addition, as the number of matches grows, it may take longer to fit the model than the length of a match. An average match in the game used here usually lasts at least 15 min, but can often last as little as 5. Once the number of matches becomes large enough so that iterative methods take longer than this to fit the model, the rating estimates will fall farther and farther behind. This will make them unusable for the real-time applications discussed in Sect. 6.

Elo [9], and also Glickman [8], both proposed methods for approximating an iterative fit by by adapting the Bradley–Terry model after every comparison without requiring storing match information. However, neither of their models has been extended to extract individual ratings from groups. The model presented here does allow individual ratings to be determined in addition to providing an efficient update after each comparison. Although adapting using only local information may result in theoretically less accurate estimates of the ratings than iterative methods, they do have the advantage of assuming a moving average of the ratings instead of a static one. This allows them to track time-varying changes in individual ratings. Over the course of thousands of matches, the true average of an individual rating may increase or decrease based on increased experience, or decrease in interest. Iterative methods such as those used by Huang et al. [7, 12] assume individuals have the same rating over all matches. While this assumption may hold over a reasonable set or hundreds of matches [12], it is less appropriate over a larger set or thousands of matches like that used here.

3 The ANN model

The section proceeds as follows: the first Sect. 3.1, presents the basic framework for viewing the Bradley–Terry model as a delta-rule trained single layer ANN, Sect. 3.2 will show how to extend the model to learn individual ratings within a group, Sect. 3.3 extends the model to allow different weights for each individual

based on their contribution, Sect. 3.4 shows how the model can be extended to learn ''home field'' advantages, Sect. 3.6 presents a heuristic to deal with uncertainty in an individual's rating, and Sect. 3.7 gives another heuristic for preventing rating inflation.

3.1 The basic model

The basis for the Bradley–Terry ANN Model begins with the ANN shown in Fig. 1. Assuming, without loss of generality, that the ANN model is designed to always predict the probability that subject A will win, the input for w_A will always be 1 and the input for w_B will always be –1. This assumption is a new approach for using ANNs because it suggests the use of interchangeable weights. It is an important part of viewing the Bradley–Terry model as an ANN because it allows a given subject to ''carry its own weight'' as its rating. That rating is ''plugged in'' as the winning or losing weight when that subject competes, and ignored otherwise. For example, given two subjects, 1 and 2, if subject 1 defeats subject 2, then w_A is set to the rating of subject 1, θ_1, and w_B is set to the rating of subject 2, θ_2. However, if subject 2 defeats subject 1, the opposite occurs: w_A is set to θ_2 because subject 2 is the winner. If a subject is not currently competing, then its weight or rating is ignored altogether—equivalent to setting the input for that weight to 0. Instead of the classical notion that weights are always associated with the same inputs, the weights are interchanged based on the outcome of the comparison. Another way of viewing this approach is that the weight for every subject is always present, but the inputs are different. If a subject wins, the input for its weight becomes 1. If that subject loses, its input is –1. If the subject is not competing, its input is 0. In addition, only two weights can be non-zero for a given comparison—in other words, only two subjects are compared at a time. There do exist extensions to the Bradley–Terry model allowing for more than one comparison at a time [4, 13], but that is beyond the scope of this work.

The equation for the model can be written as

$$\text{Output} = Pr(A \text{ defeats } B) = \frac{1}{1 + e^{-(w_A - w_B)}} \tag{4}$$

which is the same as (3), except w_A and w_B are substituted for θ_A and θ_B. This can be rewritten as:

$$\text{Output} = Pr(A \text{ defeats } B) = \frac{e^{w_A}}{e^{w_A} + e^{w_B}}, \tag{5}$$

which is the same as (1) with $\lambda_A = e^{w_A}$. This is a conditional exponential model, like that of [12].

Since we assume that w_A is always the rating of the winning subject, and w_B the rating of the losing subject, the likelihood of w_A and w_B over a set of m matches can be written as:

$$L(w_A, w_B) = \prod_{i=1}^{m} \frac{e^{w_A}}{e^{w_A} + e^{w_B}}. \tag{6}$$

where $w_{A_i}(w_{B_i})$ is the rating of the winning (losing) subject in match i. The negative log-likelihood is therefore:

$$L(w_A, w_B) = -\sum_{i=1}^{m} \log\left(\frac{e^{w_A}}{e^{w_A} + e^{w_B}}\right) \tag{7}$$

where the goal is finding the w_A and w_B for each i that minimize $L(w_A,w_B)$. As [12] stated, ''it is well known that the log-likelihood of a conditional exponential model is concave. Thus (its negative log-likelihood) is convex, so one can easily find a global minimum...'' Although [12] uses an iterative method to find the minimum of (7), they also mention that ''Standard optimization methods (e.g., gradient...) can be used'', which is what will be used here. This is done so that a sequential update can be derived that can minimize (7) over a potentially infinite set of matches.

The negative log-likelihood over the predictions of a set of Bernouli observations, like those given here, is also known as the cross-entropy of those predictions. A variation of the well-known delta rule for ANN training can be used to minimize the cross-entropy of the predictions of an ANN with respect to its weights. Further details and derivations can be found in [14–16]. For the ANN model used here, the delta rule update after a single match is

$$\Delta w_i = \eta \delta x_i \tag{8}$$

where η is a learning rate, δ is the error measured with respect to the output, and x_i is the input for w_i. The error, δ is measured here as

$$\delta = \text{Target Output} - \text{Actual Output}. \tag{9}$$

For the ANN Bradley–Terry model, the target output is always 1 given the assumption that A will defeat B. The actual output is the output of the single node ANN. Therefore, the delta rule error can be rewritten as

$$\delta = 1 - Pr(A \text{ defeats } B) = 1 - \text{Output}, \tag{10}$$

and the weight updates can be written as

$$\Delta w_A = \eta(1 - \text{Output}) \tag{11}$$

$$\Delta w_B = -\eta(1 - \text{Output}). \tag{12}$$

Here, x_i is implicit as 1 for w_A and –1 for w_B, again since A is assumed to be the winning subject. This formulation of the delta rule can be applied incrementally, updating the model after each competition, or iteratively over a training-set of paired-comparisons. The first approach may be less accurate, but is also less likely to overfit since it will never update based on the same comparison more than once. The iterative approach should only be used if the training-set can be assumed to contain a sampling of all possible competitions that matches all expected future competitions.

3.2 Individual ratings from groups

In order to extend the ANN model given in 1 to learn individual ratings, the weights for each group are obtained by averaging the ratings of the individuals in each group. Huang et al. [7, 12] modeled the strength of a given group by using the sum of the strengths of the individuals in the group. Their first model [7] assumes that when there are an uneven number of individuals in each group, the increase in rating from having a larger group is linear. Their second model [12] assumes the effect of a difference in the number of individuals is exponential. The former was appropriate for the application in [7] because they did not use their model for applications where there were an uneven number of individuals in each group, and therefore how their model handled uneven team numbers was not relevant. For the application in this paper, however, it is common to have competing groups of differing sizes. Instead of modeling the effect of an imbalance in numbers exponentially directly as Huang et al. do [12], the model we present uses an average instead so that the effect of a difference in the number of individuals can be handled separately. For example, the home field advantage formulation in Sect. 3.4 uses the relative number of individuals per group as an explicit input into the ANN, yielding an exponential increase in a group's likelihood of winning if it has more individuals than the other group. Averaging instead of summing the ratings is also analogous to normalizing the inputs—a common practice in efficiently training ANNs. Neither method changes the general form of the likelihood. The only difference is that each subject is now a group instead of an individual. Instead of inserting an individual's known rating θ_i in for the winning or losing weight, the average of a group's individual ratings is inserted for the winning or losing weight. Instead of each individual "carrying its own weight" as a rating, each individual carries its own rating, and that rating is combined with the other individuals within the group to create that weight. Therefore, the weights for the ANN model in 1 are obtained as follows:

$$w_A = \frac{\sum_{i \in A} \theta_i}{N_A} \tag{13}$$

$$w_B = \frac{\sum_{i \in B} \theta_i}{N_B}, \tag{14}$$

where $i \in A$ means individual i belongs to group A and N_A is the number of individuals in group A. With respect to the original Bradley–Terry model, the strength for a group A becomes:

$$\lambda_A = \exp\left(\frac{\sum_{i \in A} \theta_i}{N_A}\right). \tag{15}$$

Combining individual ratings in this manner means that the difference in the performance between two different individuals within a single competition can not be distinguished. However, after two individuals have compared within several different groups, their ratings can diverge.

The weight update for each individual θ_i is equal to the update for Δw_{group} given in (11):

$$\forall i \in A, \Delta\theta_i = \eta(1 - \text{Output}) \tag{16}$$

$$\forall i \in B, \Delta\theta_i = -\eta(1 - \text{Output}), \tag{17}$$

where A is the winning group and B is the losing group. In this model, instead of updating w_A and w_B directly, each individual receives the same update, therefore indirectly changing the average group ratings w_i by the same amount that the update in (11) does for Δw_A and Δw_B.

Notice that this model still assumes that groups are fixed within a comparison. Section 3.3 discusses a way to implicitly model individuals changing their groups within a single comparison.

3.3 Weighting player contribution

Depending on the type of competition, prior knowledge may suggest certain individuals within a group are more or less important despite their ratings. As an example, consider a public online multi-player competition where players are free to join and leave either team at any time. All else being equal, players that remain in the competition longer are expected to

contribute more to the outcome than those who leave early, join late, or divide their time between teams. Therefore, it would be appropriate to weight each player's contribution to w_A or w_B by the length of time they spent on team A and team B. In addition, the update to each player's rating should also be weighted by the amount of time the player spends on both teams. Therefore, w_A is rewritten as an average weighted by time spent in either group:

$$w_A = \sum_{i \in A} \theta_i \frac{t_A^i}{\sum_{j \in A} t_A^j}. \tag{18}$$

Here, t_A^i means the amount of time individual i spent in group A. The rating θ_i is therefore weighted by the amount of time individual i spends in group A relative to the total time spent by all individuals in group A. Weighting can be used for more than just the amount of time an individual participates in a competition. For example, in sports, it may have been previously determined that a given position is more important than another. This gives an average that is weighted by each individual's contribution compared to the rest of the individuals' contributions in the group.

3.4 Home field advantage

The common way of modeling home field advantage in a Bradley–Terry model is to add a parameter learned for each ''field'' into the rating of the appropriate group [17]. In the ANN model of Fig. 1, this would be analogous to adding a new connection with a weight of θ_f. If the advantage is in favor of the winning group, the input for θ_f is 1, if it is for the losing group, that input is –1. This would be one way to model home field advantage in the current ANN model. However, since the model uses group averages to form the weights, the home field advantage model can be expanded to include situations where there is an uneven amount of individuals in each group ($N_A \neq N_B$). Instead of having a single parameter per field, θ_{fg} is the advantage for group g on field f. The input to the ANN model then becomes 1 for the winning group (A) and –1 for the losing group (B). The parameters can be stored one per group per field, or each if it applies, each field can have a parameter for its home group, and one for any visitor. The following change to the chosen field inputs, f_A and f_B, extends this to handle uneven groups:

$$f_A = \frac{N_A}{N_A + N_B} \tag{19}$$

$$f_B = \frac{N_B}{N_A + N_B} \tag{20}$$

This is appealing because θ_{fg} then represents the worth of an average-rated individual from group g on field f.

Therefore, the field inputs are the relative size of each group. The full model including the field inputs then becomes:

$$\text{Output} = Pr(A \text{ defeats } B) = \frac{1}{1 + e^{-(w_A - w_B + \theta_{fA} f_A - \theta_{fB} f_B)}}. \tag{21}$$

The weight update after each comparison on field f then extends the delta-rule update from (11) to include the new parameters:

$$\Delta\theta_{fA} = \eta(1 - \text{Output}) f_A \tag{22}$$

$$\Delta\theta_{fB} = -\eta(1 - \text{Output}) f_B. \tag{23}$$

3.5 Taking into account time

One of the appealing properties of viewing a Bradley–Terry model as an ANN is that adding extensions is often as simple as including an additional input into the ANN. Time can therefore be accounted for by adding inputs x_{t_A} and x_{t_B} with corresponding time weights θ_{t_A} and θ_{t_B} into the ANN. x_{t_A} is the time input for subject A and x_{t_B} is the time input for subject B. Again, subject A is the winning subject and subject B, the losing subject. Since the ANN is being used to learn the significance of a paired difference, x_{t_A} should be positive and x_{t_B} negative. The magnitude of the input can be encoded appropriately to the given data to be fit. For example, if there is a known maximum time limit the time input can be the percent of that maximum that has passed. If time is dependant on the particular ''field'' the subjects are being compared on, then a time weight can be learned per subject per field. If it can be assumed that all visiting competitors are affected equally by time on a given ''home field'', each field can have one weight for its home subject, and one for any visitor. The time weight is updated in the same manner as the subject weights w_A and w_B by applying the delta rule:

$$\Delta\theta_{t_A} = \eta(1 - \text{Output}) x_{t_A} \tag{24}$$

$$\Delta\theta_{t_B} = -\eta(1 - \text{Output}) x_{t_B}. \tag{25}$$

The terms x_{t_A} and x_{t_B} are included here because they are not necessarily 1 or –1 as is the case with the subject inputs on weights w_A and w_B.

3.6 Rating uncertainty

One of the problems of the given model is that when an individual participates for the first time, their rating is assumed to be average. This can result in incorrect predictions when a newer individual's true rating is actually significantly higher or lower than average. Therefore, it would be appropriate to include the concept of uncertainty or variance in an individuals rating. Glickman [8] derived both a likelihood-based method and a regular-updating method (like Elo's) for modeling this type of uncertainty in Bradley–Terry models. However, his methods did not account for finding individual ratings within groups. Instead of extending his models to do so, we give the following heuristic which models uncertainty without negatively impacting the gradient descent method.

Given g_i, the number of times that individual i has participated in a comparison, let the *certainty* γ_i of that individual be:

$$\gamma_i = \frac{\min(g_i, G)}{G}, \tag{26}$$

where G is a constant that represents the number of comparisons needed before individual i's rating is fully trusted. The overall certainty of the comparison is then the average certainty of the individuals from both groups. If a weighting is used, then the certainty for the comparison is the weighted average of all the individuals in either group. The probability estimate then becomes:

$$\text{Output} = Pr(A \text{ defeats } B) = \frac{1}{1 + e^{-(c(w_A - w_B) + \theta_{fA} f_A - \theta_{fB} f_B)}}, \tag{27}$$

where c is the average certainty of all of the individuals participating in the comparison. This effectively stretches the logistic function, making the probability estimate more likely to tend towards 0.5—which is desirable in more uncertain comparisons. For example, assuming the map weights are close to 0, a comparison yielding a 70% chance of group A defeating group B with $c = 1.0$, would yield a 60% chance of group A winning if $c = 0.5$. The modified weight update appears as follows:

$$\forall i \in A, \Delta\theta_i = c\eta(1 - \text{Output}) \tag{28}$$

$$\forall i \in B, \Delta\theta_i = -c\eta(1 - \text{Output}). \tag{29}$$

Since this can also be seen to effectively lower the learning rate, η is in practice chosen to be twice as large for the individual updates as the field-group updates.

The down side to this method of modeling uncertainty is that is does not allow for a later change in an individuals rating due to long periods of inactivity or due to improving rating over time. This could be implemented with a form of certainty decay that lowered an individual's certainty value γ_i, over time. Also, notice a similar measure of uncertainty could be applied to the field-group parameters.

3.7 Preventing rating inflation

One of the problems with averaging (or summing) individual ratings is that there is no way to model a single individual's expected contribution based on their rating. An individual with a rating significantly higher or lower than a group they participate in will have an equal update to the rest of the group. This can result in inflating that individual's rating if they constantly win with weaker groups. Likewise, a weaker individual participating in a highly skilled but losing group may have their rating decreased farther than is appropriate because that weaker individual was not expected to help their group win as much as the higher rated individuals were. One heuristic to offset this problem is to use an individual's own rating instead of that individual's group rating when comparing that individual to the other group. This will result in individuals with ratings higher than their group's average receiving smaller updates than the rest of the group when their group wins, and larger negative updates when they lose. The opposite is true for individuals with ratings lower than their group's average. They will be larger when they win, and smaller when they lose. This attempts to account for situations where there are large differences in the expected worth of participating individuals. This substitution can be written as

$$w_A = \theta_i \tag{30}$$

if individual i is in the winning group, and

$$w_B = \theta_i \tag{31}$$

if individual i is on the losing group. The substitution is only used to calculate $Pr(A \text{ defeats } B)$ for the *Output* used in player i's update.

4 Experiments

The final model employs all of the extensions discussed in Sect. 3. The model was developed originally to rate players and predict the outcome of matches in the

World War II-based online team first-person shooter computer game Enemy Territory. With hundreds of thousands of players over thousands of servers worldwide, Enemy Territory is one of the most popular multi-player first-person shooters available. In this game, two teams of on average 4–20 players each, the Axis and the Allies, compete on a given ''map''. Both teams have objectives they need to accomplish on the map to defeat the other team. Whichever team accomplishes their objectives first is declared the winner of that map. Usually one of the teams has a time-based objective—they need to prevent the other team from accomplishing their objectives for a certain period of time. If they do so, they are the winner. In addition, the objectives for one team on a given map can be easier or harder than the other team's objectives. The size of either team can also be different and players can both enter, leave, and change teams at will during the play of the given map. These characteristics motivated the development of the above extensions because they are common in public Enemy Territory servers. Weighting individuals by time deals with players coming, going, and changing teams. Incorporating a group size-based ''home field advantage'' extension deals with both the uneven numbers and the difference in difficulty for either team on a given map. The time parameters allow the model to adapt to situation where the length of the match is unusual for the given teams. The certainty parameter was developed to deal with inflated probability estimates given several new and not fully tested players. Since it is common practice for a server to drop a player's rating information if they do not play on the server for an extended period of time, no decay in the player's certainty was deemed necessary.

The model was developed to predict the outcome for a given set of teams playing a match on a given map. Therefore, the individual and field-group parameters (in this case both an Axis and Allies parameter for each map) are updated after the winner is declared for each match.

In order to evaluate the effectiveness of the model, its extensions, and training method, data was collected from 3,379 competitions including 4,249 players and 24 unique maps. The data included which players participated in each map, how long they played on each team, which team won the map, and how long the map lasted. Custom software was implemented in Python to simulate the ANN and its training. The ANN model is trained as described in Sect. 3, using the update rule after every map. The model is evaluated five times, only adding the time parameters for the last run. The runs without the time parameters include one with no heuristics, one with only the certainty heuristic from Sect. 3.6, one with only the rating inflation prevention heuristic from Sect. 3.7, and one combining both heuristics. The final run includes all of the heuristics because they are shown to be effective, and it also includes the time parameters.

For the results given, the model's accuracy on a given match is always measured *before* being trained that match. This leave-one-out approach ensures the results are not biased in favor of the data used for training. It does, however, result in a bias that leads to less accurate results initially and then more accurate results near the end. This however, is appealing because it gives a measure of the performance of the model when used for the real-time application for which it was designed—namely continually ranking and rating players in this online game. When applied in the real-world, the model will be fit in this exact sequential manner, and therefore the results given are specifically appropriate to the application.

One way to measure the effectiveness of these runs would be to look at how often the team with the higher probability of winning does not actually win. However, this result can be misleading because it assumes a team with a 55% chance of winning should win 100% of the time—which is not desirable. The real question is whether or not a team given a P% chance of winning actually wins P% of the time. Therefore, the method chosen to judge the model's effectiveness uses a histogram to measure how often a team given a P% chance of winning actually wins. For the results in Sect. 5, the size of the histogram intervals is chosen to be 1%. This measure is called the *prediction error* because it shows how far off the predicted outcome is from the actual result. A prediction error of 5% means that when the model predicts team A has a 70% probability of winning, they really have a probability of winning between 65 and 75%. Or, in the histogram sense, it means that given all of the occurrences of the the model giving a team a 70% chance of winning, that team will actually win in 65–75% of those occurrences.

As mentioned in Sect. 1, there are no currently published models which with to compare this one, so none are given. It can only be said that given the complexity of the problem, producing reasonable estimates of match outcomes is, in and of itself, significant.

5 Results and analysis

The results are shown in Table 1. Each column gives the prediction error resulting from a different combi-

Table 1 Bradley–Terry ANN Enemy Territory prediction errors

Heuristic	None	Inflation	Certainty	Both	With time
Prediction error	0.1034	0.0635	0.0600	0.0542	0.0457

nation of heuristics. The first column uses no heuristics, the second uses only the inflation prevention heuristic (see Sect. 3.7), the third column uses only the rating certainty heuristic (see Sect. 3.6), the fourth column combines both heuristics, and the last column combines both heuristics along with taking into account the time parameters. Each heuristic improves the results and combining both gives a prediction error at 5.42%. Adding in the time parameters brings the error 16% lower to 0.0457. This value means that a given estimate is, on average, 4.57% too high or 4.57% too low. Therefore, when the model predicts a team has a 75% chance of winning, the actual average % chance of winning is between 70 and 80%.

One question is on which side does the prediction error tend for a given probability. To examine this, two plots are shown. Each plot gives the predictions plotted against the true values taking 20 5% histogram intervals of the results. Figure 2 shows the predictions for the final model without the time parameters, and Fig. 3 shows the predictions with the time parameters. The lines in the middle of each figure give the ideal, and therefore deviations from the line show the direction of the error for each prediction. Notice that with only two low-error exceptions on the extremes, the prediction error for the model without the time parameters tends to be positive when the prediction probability is low,

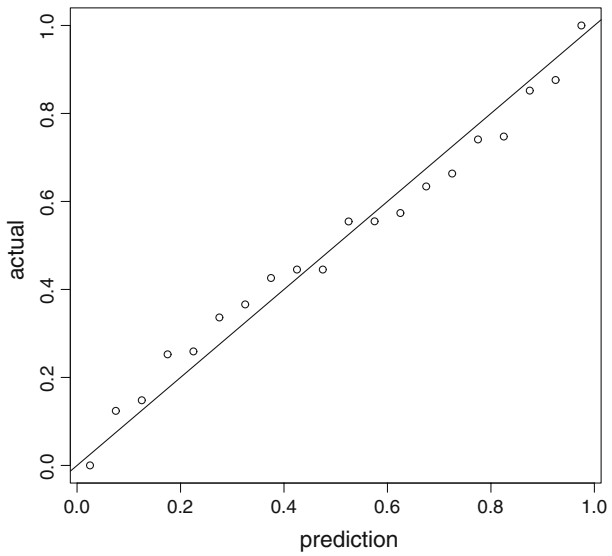

Fig. 2 Prediction versus true probability using 20 5% histogram intervals

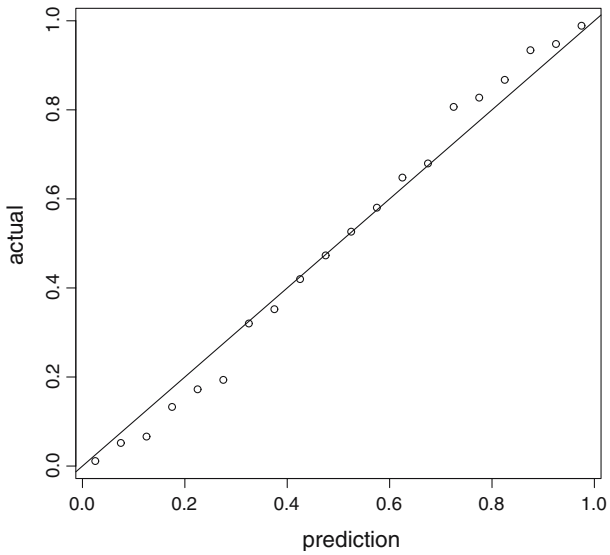

Fig. 3 Prediction versus true probability using 20 5% histogram intervals with the time-based parameters added

and negative when it is high. This means that the original model is slightly extreme on average. For example, if the model predicts a team is 75% likely to win, they will be on average only 70% likely to win. A team predicted to be 25% likely to win will be on average 30% likely to win. The model with the time parameters, however, has the opposite trend. It tends to have negative error when the prediction is low, and positive when it is high. This means it is more conservative. When the newer model predicts a team is 75% likely to win, that team is really, on average, closer to 80% likely to win. When it predicts a team in 25% likely to win, that team is closer to 20% likely to win. It is not surprising that taking into account time results in a more conservative model because it lowers the expectation of a superior team winning if that team is ''taking too long'' to win. Another interesting difference when taking into account time is the fact it is more accurate near 50% (and therefore more likely to predict ties correctly) and the clusters are ''tighter'', meaning it has a lower variance.

In addition to evaluating the model analytically, the ratings it assigns to the players can be assessed from experience. For the matches used for the results, the well-known best players also have the highest ratings according to the model. These players are well-known on for their tenacity in winning maps, and therefore the model's ranking appears to fit intuition gained from experience with these players. In summary, the model effectively predicts the outcome of a given map with a prediction error of less than 5%, and provides reasonable ratings for the players.

6 Application

One of the goals in creating an entertaining multi-player game or running an entertaining multi-player game server is keeping the gameplay fair for both teams. Players are more likely to continue playing a game or continue playing on a given game server if the competition is neither too easy nor too hard. When the gameplay is uneven, players tend to become discouraged and leave the server or play different games. This is where the rating system becomes useful. Besides being able to rate individual players and predict the outcome of matches in a multiplayer game like Enemy Territory, the predictions can also be used during a given map to balance the teams. If a team's probability of winning is greater than some chosen threshold, a player can be moved from that team to the other team in order to ''even the odds.'' The Bradley–Terry ANN model as adapted for Enemy Territory is now being run on hundreds of Enemy Territory servers involving hundreds of thousands of players world-wide. A large number of those servers have enabled an option that keeps the teams balanced, and therefore, in theory, keeps the gameplay entertaining for the players.

One extension of this system would be to track player ratings across multiple servers world-wide with a master server and then recommend servers to players based on the difficulty level of each server. This would allow players to find matches for a given online game in which the gameplay felt ''even'' to them. This type of ''difficulty matching'' is already used in several online games today, but only for either one-on-one type matches, or teams where the players never change. The given Bradley–Terry ANN model could extend this to allow for dynamic teams on public servers that maintain an even balance of play.

This can also be applied to improving groups chosen for sports or military purposes. Individuals can be rated within groups as long as they are shuffled through several groups and then, over time, groups can be either chosen by using the highest rated individuals, or balanced by choosing a mix for each group. The higher-rated groups would be appropriate for real encounters, whereas balanced groups may be preferred for practicing or training purposes.

7 Weight analyses

Besides the importance of the accuracy of the proposed model, also of interest are the values of the parameters themselves. Analyzing these parameters can lead to improvements in gameplay. For example, examining the time parameters and the field parameters can lead to insights that allow server administrators to more fairly balance matches, and allow level and map designers to either create more balanced maps, or recommend more appropriate time limits.

As an example, consider the time and field weights assigned to two maps shown in Tables 2 and 3. First, note that the time weights are symmetrical. This is not surprising because the weight updates are the same for the time weights. In fact, both the field and time weights could be stored in a single weight, but they are separated for ease of interpretation. Notice in the baserace map (Table 2) the field weights are close and the time weights are small. This exactly matches intuition about baserace because it is a perfectly symmetrical map. The objectives are the same for both sides, and the physical layout is also the same on both sides of the map. Therefore, the expectation is that the weights should be very similar, and time should not have a strong affect on the map. The ice map, on the other hand, is almost the opposite. It is far easier in practice for the Axis team to win the ice map—however, when they do win, they usually win quite early. The negative time weight for the Axis team suggests that they become less and less likely to win as time passes. In practice, this usually means the Allies have found a way to defend against the Axis on this map more effectively than usual. Notice, however, that the difference in the field weights is still greater than the difference in the time weights. This suggests that even though the Axis team will become *less* likely to win over time, they are still expected to win more often in general. This also suggests that the model has detected that the ice map's timelimit may be too long because the input for the time weights is the amount of time the map took over the timelimit. If shortened, the field and time parameters may come more into balance. This was seen using the team balancing system in practice.

As suggested above, the predictions from the Bradley–Terry model as applied to Enemy Territory

Table 2 Time and field weights for the baserace map

	Field	Time
Allies	1.86	0.12
Axis	2.01	–0.12

Table 3 Time and field weights for the ice map

	Field	Time
Allies	–0.97	1.04
Axis	1.78	–1.04

can be used to actively balance the teams in a given match. In earlier tests, balancing the team without taking time into account resulted in maps like ice being heavily stacked in favor of the defense. This is because without the time weight parameter, the model can only assume that without a greater number of players, the Axis team will win most of the time. Adding the time weight parameter to this system results in more balanced play. It still stacks the defense initially, however, as time passes, the time-weighted model detects the Axis team is not as likely to win as originally predicted, and moves more players off the defense. Even with the time weighting, however, it was not uncommon for the team balancing system to not allow *any* players on the Axis team for several minutes. This is due to the aforementioned domination of the time weights by the field weights. Here, the team balancing system is effectively giving the unfavored Allies team a ''head start'' While logically correct, this is less entertaining for the players, who prefer to have contestants to play against, regardless of the difficulty. Fortunately, the ''head start'' the team balancing system suggests also implies that the map's timelimit may be too long. By decreasing the time limit of the map by roughly the amount of time the balancing system kept the teams empty, the time weight can come more into balance with the field weight. This results in a more balanced experience for both teams. Therefore, the results of adding time weights not only improves the model's predictive power, but also leads to new insights that improve gameplay overall. This suggests the model can be used not only for prediction, but map creators can use it for creating more balanced and therefore more entertaining maps. Games that provide a more entertaining experience are generally more popular, which is beneficial to both the makers of the games, and the players who will have more people with which to participate.

Analyzing time weights can be used for more than just improving the balance of online games. It can be applied to any situation where determining the effects of time on a comparison or competition is important. In the military sense, it can be used to balance exercises or to determine how long to allow an encounter to continue. For sports it can be used to better determine how long a match should last. This can include team sports as well as other time involved matches like ice-skating where a skater has a set amount of time in their program. In education, it could be applied to how long it takes to learn a given concept when comparing different methods of teaching. In medicine, in can be applied to determine how time will affect the recovery or worsening of a given patient using different treatments. There are several significant applications where analyzing the effects of the passage of time can lead to new insights on important paired-comparisons.

8 Conclusions and future work

This paper has presented a method to learn Bradley–Terry models using an ANN. An ANN model is appealing because it makes using and extending the Bradley–Terry model accessible for a broader community. In addition, ANN re-parameterization also provides a training rule that can be used incrementally or iteratively. The basic model is extended to rate individuals where groups are being compared, to allow for weighting individuals, to model ''home field advantages'', to take into account the effects of time on gameplay, to deal with rating uncertainty, and to prevent rating inflation. The results when applied to a large, real-world problem including thousands of comparisons involving thousands of players yields probability prediction estimates within less than 5% of the actual values. In addition, analysis of the resulting parameters leads to new insights that suggest ways of improving the game's overall balance.

This type of individual rating out of groups can be important in many other applications. This can include rating individual players on sports teams, rating soldiers that participate in military group training, rating employees who participate in competing projects across different groups, rating competing manufacturing processes at the machine-level, and any other application where it is advantageous to rank individual subjects when they are only compared at group levels. In today's world, overall performance is measured more often than not at the group instead of individual level, and therefore approaches for rating in that context are becoming more and more important. The fact that the given method provided here performs well on an application as complex as public-style online gaming suggests it could perform well on other important applications as well.

One of the problems with the current model is that all associations are assumed to be additive and exponential. They are additive in the sense that the skill of a given group is proportional to the exponentiated sum of the skill of the individuals in that group. It does not take into account higher-level interactions between the individuals. It is exponential because the effects of time and the number of individuals per group is assumed to affect the model in an exponential fashion. This may be inaccurate if having more individuals implies more or less an exponential increase in the skill, or being in a

group twice as long affects the outcome in a non-exponential fashion. Two approaches to improving the robustness of the current model include:

1. Extending the current single-layer ANN to a multi-layer ANN.
2. Constructing a higher-level statistical model that allows for these additional complexities.

One of the nice properties of using an ANN as a Bradley–Terry model is that it can be easily extended from a single-layer model to a multi-layer ANN. No currently known Bradley–Terry model incorporates interaction effects between individuals within a group or in a competing group. Current models do not consider that individuals i and j may have a higher effective rating when playing together than apart. A multi-layer ANN is able to model these higher-order interactions by making each hidden node represent a subset of the stronger interactions. In addition, a multi-layer ANN is able to learn non-linear functions of its inputs, and therefore a multi-layer Bradley–Terry ANN can find if either or both of the group size and time inputs affect the outcome in a different fashion. The next Bradley–Terry model will be a multi-layer Bradley–Terry ANN constructed to determining these relationships.

One of the down sides of using a multi-layer ANN is that it will be harder to interpret the individual ratings. Therefore, statistical approaches including hierarchical Bayes will also be applied to see if a more identifiable model can be constructed. One way to develop an identifiable interaction model can extend from the fact that it is common for sub-groups of players to play in ''fire-teams'' or ''squads'' in games like Enemy Territory. An interaction model can be designed that uses the fire-teams as a basis without needing to consider the intractable number of all-possible interactions. For the time and group size effects, more robust, non-linear functions can be chosen and fit for combining these parameters. Using a statistical model would also make it more natural to account for the uncertainty in individual ratings. The uncertainty can be modeled directly as the variance in a given individual's rating. This more principled approach to handling uncertainty can then be used to replace the Bradley–Terry ANN model parameter c with an true estimate instead of a heuristic.

References

1. Bradley R, Terry M (1952) The rank analysis of incomplete block designs I: The method of paired comparisions. Biometrika 39:324–345
2. Davidson RR, Farquhar PH (1976) A bibliography on the method of paired comparisons. Biometrics 32:241–252
3. David HA (1988) The method of paired comparisons. Oxford University Press, Oxford
4. Hunter DR (2004) MM algorithms for generalized Bradley–Terry models. Ann Stat 32:386–408
5. Hastie T, Tibshirani R (1998) Classification by pairwise coupling. In: Jordan MI, Kearns MJ, Solla SA (eds) Advances in neural information processing systems, vol 10. The MIT Press, Cambridge
6. Zadrozny B (2001) Reducing multiclass to binary by coupling probability estimates. In: NIPS, pp 1041–1048
7. Tzu-Kuo Huang, Ruby C. Weng, Chih-Jen Lin (2006) Generalized Bradley–Terry models and multi-class probability estimates. J Mach Learn Res 7:85–115
8. Mark E Glickman (1999) Parameter estimation in large dynamic paired comparison experiments. Appl Stat 48(3):377–394
9. Arpad E Elo (1978) The rating of chess players: past and present. Arco Publishing, New York
10. Stigler S (1994) Citation patterns in the journals of statistics and probability. Stat Sci 9:94–108
11. Sham PC, Curtis D (1994) An extended transmission/disequilibrium test (tdt) for multiallele marker loci. Ann Hum Genetics 59(3):323–336
12. Tzu-Kuo Huang, Chih-Jen Lin, Ruby C Weng (2006) Ranking individuals by group comparisons. In: ICML '06: Proceedings of the 23rd international conference on Machine learning. ACM Press, New York, pp 425–432
13. Graves TL, Reese CS, Fitzgerald M (2003) Hierarchical models for permutations: analysis of auto racing results. J Am Stat Assoc 98(462):282–291
14. Hinton GE (1989) Connectionist learning procedures. Artif Intell 40(1–3):185–234
15. Van Ooyen A, Nienhuis B (1992) Improving the convergence of the backpropagation algorithm. Neural Netw 5:465–471
16. Hampshire JB II, Perlmutter BA (1990) Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function. In: Touretzky D, Elman J, Sejnowski T, Hinton G (eds) Proceedings of the 1990 connectionist models summer school. Morgan Kaufmann, San Mateo
17. Agresti A (1988) Categorical data analysis. Oxford University Press, Oxford