Efficient computation of rankings from pairwise comparisons

M. E. J. Newman
Center for the Study of Complex Systems
University of Michigan
Ann Arbor, MI 48109, USA

Abstract

We study the ranking of individuals, teams, or objects, based on pairwise comparisons between them, using the Bradley-Terry model. Estimates of rankings within this model are commonly made using a simple iterative algorithm first introduced by Zermelo almost a century ago. Here we describe an alternative and similarly simple iteration that provably returns identical results but does so much faster—over a hundred times faster in some cases. We demonstrate this algorithm with applications to a range of example data sets and derive a number of results regarding its convergence.

1. Introduction

The problem of ranking a set of individuals, teams, or objects on the basis of a set of pairwise comparisons between them arises in many contexts, including competitions in sports, chess, and other games, paired comparison studies of consumer choice, and observational studies of dominance behaviors in animals and humans (Zermelo, 1929; Bradley and Terry, 1952; Davidson and Farquhar, 1976; David, 1988; Cattelan, 2012). If a group of chess players play games against one another, for example, how can we rank the players, from best to worst, based on the outcome of those games? The outcomes may be contradictory or ambiguous—underdogs sometimes win and strong players sometimes lose—so we adopt a probabilistic model. In the most common version we assign a numerical score s_i to each individual i and the probability p_{ij} of i beating j is assumed to be some function of the difference in their scores $p_{ij} = f(s_i - s_j)$. The most popular choice of functional form is the logistic function $f(s) = 1/(1 + e^{-s})$, which gives

$$p_{ij} = \frac{e^{s_i}}{e^{s_i} + e^{s_j}}. \tag{1}$$

This is the Bradley-Terry model, first introduced by Zermelo (1929) and heavily studied in the years since, particularly following its rediscovery by Bradley and Terry (1952). For convenience, one often introduces the shorthand $\pi_i = e^{s_i}$ so that p_{ij} can also be written as

$$p_{ij} = \frac{\pi_i}{\pi_i + \pi_j}, \tag{2}$$

and we will do that here. Zermelo (writing in German) referred to the non-negative parameters π_i as *Spielstärken* or "playing strengths," although they are elsewhere variously called worth parameters, skill parameters, merit parameters, ratings, or simply weights. Following Zermelo, we will call them strengths.

Given the outcomes of a series of pairwise competitions between N competitors the strengths can be estimated straightforwardly. Commonly one makes a maximum-likelihood

estimate. Defining w_{ij} to be the total number of times i beats j, or zero if i and j never competed, it can be shown that the maximum-likelihood values of the strengths are given by a simple procedure: starting from any convenient initial values we iterate the equation

$$\pi_i' = \frac{\sum_{j=1}^{N} w_{ij}}{\sum_{j=1}^{N} (w_{ij} + w_{ji})/(\pi_i + \pi_j)} \tag{3}$$

until convergence is reached. This algorithm was also first described by Zermelo (1929) and we will refer to it as Zermelo's algorithm. An extraordinary number of papers have been written about it, its variants, its properties, and its applications.

Although widely used, however, Zermelo's algorithm is known to be slow to converge (Dykstra, 1956; Hunter, 2004). In this paper we study the alternate iteration

$$\pi_i' = \frac{\sum_{j=1}^{N} w_{ij}\pi_j/(\pi_i + \pi_j)}{\sum_{j=1}^{N} w_{ji}/(\pi_i + \pi_j)}. \tag{4}$$

We show that iteration of this equation solves the same problem and converges to the same solution as Zermelo's algorithm but does so significantly faster—over a hundred times faster in some cases. Given that Eq. (4) is also simple to implement we know of no reason not to favor it over Eq. (3).

In recent years a number of other authors have also considered alternative and potentially more efficient algorithms for ranking under the Bradley-Terry model. One promising approach employs spectral methods that estimate rankings based on the properties of random walks on the network of directed interactions between individuals (Maystre and Grossglauser, 2015; Negahban et al., 2017; Agarwal et al., 2018). Although they do not directly maximize likelihood under the Bradley-Terry model, these algorithms can be shown to converge closely to the maximum-likelihood solution. Some versions are quite numerically efficient, though they can also be complex to implement. Minorization-maximization (MM) algorithms, which optimize a minorizing proxy for the likelihood function, can also be applied to ranking problems. For the Bradley-Terry model the appropriate MM algorithm turns out to be exactly equivalent to Zermelo's algorithm and hence offers no speed improvement (Hunter, 2004), but techniques have been suggested for accelerating convergence (Vojnovic et al., 2019) and the MM formulation also provides an elegant route to developing algorithms for generalizations of the model. Perhaps more directly competitive with our approach is one of the simplest of methods: one can fit the Bradley-Terry model using Newton's method applied to the derivative of the likelihood. For small applications with only a few individuals or teams to be ranked this is typically the fastest approach, although for such small cases the difference may be moot. As the number N of individuals gets larger, however, Newton's method becomes impractical because the time taken per iteration of the algorithm scales as N^3, which quickly becomes prohibitive (Hunter, 2004). For Eq. (4) the time per iteration scales only as the number of pairwise competitions, making this approach faster for larger applications. Maximum-likelihood methods are not the only approach for fitting: one can also adopt Bayesian approaches, although these usually require Monte Carlo estimation and hence are not competitive in terms of speed (Davidson and Solomon, 1973; Caron and Doucet, 2012). Overall, the particular combination of simplicity and speed offered by Eq. (4) makes it an attractive approach for practical applications.

The rest of this paper is organized as follows. In Section 2 we describe and derive Zermelo's original ranking algorithm and the new algorithm proposed here. In Section 3 we prove that the new algorithm converges to the same unique maximum-likelihood solution as Zermelo's algorithm. In Section 4 we briefly discuss a larger family of ranking algorithms of which Zermelo's algorithm and our own are special cases. Within this family, the algorithm of this paper is the fastest to converge and hence it is our primary focus. In Sections 5 and 6 we describe two extensions of our approach, one to maximum a priori (MAP) estimation of rankings and the other to a (previously proposed) generalization of the Bradley-Terry model that allows for ties or draws in competitions. In Section 7 we apply our algorithms to a broad selection of example data, both real and synthetic, finding in every case that the algorithm of this paper is faster than Zermelo's, often by a wide margin. In Section 8 we give our conclusions. Some additional technical results are presented in appendices.

2. Iterative algorithms and the Bradley-Terry model

Consider a tournament where N players or teams play games of some kind against one another in pairs. We will initially assume that no ties or draws are allowed, so there is a clear winner and loser of every game. The case where ties are allowed is treated separately in Section 6. We assume that the probability p_{ij} that player i beats player j obeys Eq. (1) and we consider the strengths π_i to be a measure of the skill of the players, higher values indicating better players.

Note that the win probabilities p_{ij} are invariant under multiplication of all the π_i by any constant. One can remove this ambiguity by imposing any convenient normalization condition. Here we fix the geometric mean strength to be 1, which is equivalent to setting $\prod_i \pi_i = 1$. This choice has the nice effect that the probability p_1 of a player with strength π beating the average player with strength 1 is $p_1 = \pi/(\pi+1)$ and hence $\pi = p_1/(1-p_1)$. Thus the strength parameter has a simple interpretation: it is the odds of beating the average player.

2.1 Zermelo's algorithm

Suppose the tournament consists of a total of M games between pairs of players and let w_{ij} be the number of times that player i beats player j. Using these data we can make a maximum-likelihood estimate of the strengths π_i as follows. The likelihood of the observed games given the strengths (represented by a matrix $W = [w_{ij}]$ and vector $\pi = [\pi_i]$ respectively) is

$$P(W|\pi) = \prod_{ij} p_{ij}^{w_{ij}} = \prod_{ij} \left(\frac{\pi_i}{\pi_i + \pi_j} \right)^{w_{ij}}, \tag{5}$$

so that the log-likelihood is

$$\log P(W|\pi) = \sum_{ij} w_{ij} \log \frac{\pi_i}{\pi_i + \pi_j} = \sum_{ij} w_{ij} \log \pi_i - \sum_{ij} w_{ij} \log(\pi_i + \pi_j). \tag{6}$$

Differentiating with respect to π_i for any i and setting the result to zero we get

$$\frac{1}{\pi_i} \sum_j w_{ij} - \sum_j \frac{w_{ij} + w_{ji}}{\pi_i + \pi_j} = 0, \tag{7}$$

which can be rearranged to read

$$\pi_i = \frac{\sum_j w_{ij}}{\sum_j (w_{ij} + w_{ji})/(\pi_i + \pi_j)}. \tag{8}$$

In general this equation has no closed-form solution but it can be solved numerically by simple iteration: one picks a suitable set of non-negative starting values for the π_i—random values are often used—and then computes new values π_i' according to

$$\pi_i' = \frac{\sum_j w_{ij}}{\sum_j (w_{ij} + w_{ji})/(\pi_i + \pi_j)}. \tag{9}$$

Iterating this process, it can be proved subject to certain conditions that we converge to the global maximum of the likelihood and hence obtain an estimate of the strengths π_i (Zermelo, 1929; Ford, 1957; Hunter, 2004). The values can then be sorted in order to give a ranking of the players, or simply used in their raw form as a kind of rating. The iteration can be performed synchronously (all π_i updated at the same time) or asynchronously (π_i updated one by one in cyclic fashion), but it is generally believed that asynchronous updating is more efficient, since the update of any individual π_i benefits from the improved estimates of previously updated ones. In this paper we use asynchronous updates. Our convergence results in Sections 3 and 4 are also for the asynchronous case.

This iterative algorithm, first described by Zermelo (1929), is the standard method for calculating rankings within the Bradley-Terry model and has seen numerous applications over the years in a wide variety of contexts.

2.2 An alternative algorithm

In a sense, computing maximum-likelihood estimates for the Bradley-Terry model is a straightforward problem. As discussed in Section 3, the likelihood is concave and many standard convex optimization methods can be applied. Speed, however, is of the essence in practical applications of the model, so considerable effort has been exerted in recent years to find solution methods faster than Zermelo's algorithm (Maystre and Grossglauser, 2015; Negahban et al., 2017; Agarwal et al., 2018; Vojnovic et al., 2019). Some of these are quite complex, but here we propose a very simple approach that also turns out to be highly efficient.

Grouping the terms slightly differently, Eq. (7) can be rewritten as

$$\frac{1}{\pi_i} \sum_j w_{ij} \frac{\pi_j}{\pi_i + \pi_j} - \sum_j \frac{w_{ji}}{\pi_i + \pi_j} = 0, \tag{10}$$

which can be rearranged as

$$\pi_i = \frac{\sum_j w_{ij}\pi_j/(\pi_i + \pi_j)}{\sum_j w_{ji}/(\pi_i + \pi_j)}. \tag{11}$$

This suggests a different iterative algorithm for the Bradley-Terry model. Again we choose suitable starting values (for instance at random), then we iterate the form

$$\pi_i' = \frac{\sum_j w_{ij}\pi_j/(\pi_i + \pi_j)}{\sum_j w_{ji}/(\pi_i + \pi_j)} \tag{12}$$

4

to convergence. In Section 3 we prove that, like Zermelo's algorithm, this process converges to the global maximum of the likelihood.

One nice feature of this algorithm is that it is transparent from Eq. (12) that $\pi_i = 0$ for any individual who loses all their games and $\pi_i = \infty$ for any individual who wins all their games. Furthermore, the iteration converges to these values in a single step. The same values are also returned by the standard Zermelo algorithm, but it is less obvious from Eq. (9) that this is true—it is some work to demonstrate the result for the player who wins every game and moreover it takes the Zermelo algorithm an infinite number of iterations to reach the correct value instead of just one iteration. This is a special case of the more general finding, which we explore in this paper, that (12) converges faster than Zermelo's algorithm.

3. Convergence

In this section we prove that the iteration of Eq. (12) converges to the global maximum of the likelihood, Eq. (5), from any starting point, whenever a maximum exists.

Zermelo proved that the likelihood has only one stationary point for $\pi_i \geq 0$, corresponding to the global maximum, provided the π_i are normalized as discussed in Section 2 and the directed network of interactions (the network with adjacency matrix w_{ij}) is strongly connected, i.e., there is a directed path through the network from every individual to every other (Zermelo, 1929; Ford, 1957). If the network is not strongly connected then there are no stationary points and there is no maximum of the likelihood, and hence our problem has no solution. For the moment we will assume, as other authors have done, that the network is strongly connected and hence that there is a maximum of the likelihood, although we show how to relax this requirement in Section 5.

Since any fixed point of the iteration of Eq. (12) corresponds to a stationary point of the likelihood, and since the iteration generates non-negative values of π_i only (given non-negative initial values), it follows that if the iteration converges to a fixed point that point must be the global maximum. To prove that it converges to a fixed point it suffices to demonstrate that the value of the log-likelihood always increases upon application of Eq. (12) unless a fixed point has been reached, since the log-likelihood cannot increase without bound, being bounded above by the maximum.

We consider the asynchronous version of the iteration of Eq. (12) in which a single π_i is updated at each step, all others π_j remaining the same. The π_i are updated in order until all N have been updated. Consider the step on which a particular π_i is updated. We define a function $f(\pi_i)$ equal to the sum of the terms in the log-likelihood, Eq. (6), that depend on π_i:

$$f(\pi_i) = \sum_j w_{ij} \log \frac{\pi_i}{\pi_i + \pi_j} - \sum_j w_{ji} \log(\pi_i + \pi_j). \tag{13}$$

Noting that $\log x \leq x - 1$ for all real $x > 0$ and making the substitution $x \to x/y$, we derive the useful inequality

$$\log x \leq \log y + \frac{x}{y} - 1 \tag{14}$$

5

for all $x, y > 0$, or equivalently

$$\log y \geq \log x - \frac{x}{y} + 1, \tag{15}$$

with the exact equality holding if and only if $x = y$. This implies for any π_i and π'_i that

$$\log \frac{\pi'_i}{\pi'_i + \pi_j} \geq \log \frac{\pi_i}{\pi_i + \pi_j} - \frac{\pi_i/(\pi_i + \pi_j)}{\pi'_i/(\pi'_i + \pi_j)} + 1 = \log \frac{\pi_i}{\pi_i + \pi_j} + \frac{(\pi'_i - \pi_i)/\pi'_i}{(\pi_i + \pi_j)/\pi_j}, \tag{16}$$

and (14) implies that

$$\log(\pi'_i + \pi_j) \leq \log(\pi_i + \pi_j) + \frac{\pi'_i + \pi_j}{\pi_i + \pi_j} - 1 = \log(\pi_i + \pi_j) + \frac{\pi'_i - \pi_i}{\pi_i + \pi_j}. \tag{17}$$

Evaluating Eq. (13) at the point π'_i defined by Eq. (12) and applying these two inequalities, we find that

$$
\begin{aligned}
f(\pi'_i) &= \sum_j w_{ij} \log \frac{\pi'_i}{\pi'_i + \pi_j} - \sum_j w_{ji} \log(\pi'_i + \pi_j) \\
&\geq \sum_j w_{ij} \log \frac{\pi_i}{\pi_i + \pi_j} + \frac{\pi'_i - \pi_i}{\pi'_i} \sum_j w_{ij} \frac{\pi_j}{\pi_i + \pi_j} \\
&\qquad - \sum_j w_{ji} \log(\pi_i + \pi_j) - (\pi'_i - \pi_i) \sum_j \frac{w_{ji}}{\pi_i + \pi_j} \\
&= \sum_j w_{ij} \log \frac{\pi_i}{\pi_i + \pi_j} - \sum_j w_{ji} \log(\pi_i + \pi_j) \\
&\qquad + (\pi'_i - \pi_i) \left[\frac{1}{\pi'_i} \sum_j w_{ij} \frac{\pi_j}{\pi_i + \pi_j} - \sum_j \frac{w_{ji}}{\pi_i + \pi_j} \right] \\
&= f(\pi_i), \tag{18}
\end{aligned}
$$

where we have used Eq. (13) again, the term inside the square brackets vanishes because of (12), and the exact equality applies if and only if $\pi'_i = \pi_i$.

Thus $f(\pi_i)$ always increases upon application of Eq. (12) and hence so also does the log-likelihood, unless $\pi'_i = \pi_i$, in which case the log-likelihood remains the same but could still increase when one of the other π_i is updated. Only if $\pi'_i = \pi_i$ for all i does the log-likelihood not increase at all, but if this occurs then by definition we have reached a fixed point of the iteration, and hence we have reached the global maximum. This now guarantees convergence of the iterative algorithm of Eq. (12) to the global likelihood maximum.

4. Other iterative algorithms

Given the existence of two different iterations, Eqs. (9) and (12), that both converge to the same maximum-likelihood estimate, one might wonder whether there exist any others. In fact, it turns out there is an entire one-parameter family of iterations that includes (9) and (12) as special cases, and all of them converge to the same solution. Of these iterations,

Eq. (12) converges most rapidly and hence is our primary focus in this paper, but for the interested reader we discuss the full family briefly in this section.

For any α we can rewrite Eq. (10) in the form

$$\frac{1}{\pi_i} \sum_j w_{ij} \frac{\alpha \pi_i + \pi_j}{\pi_i + \pi_j} - \sum_j \frac{\alpha w_{ij} + w_{ji}}{\pi_i + \pi_j} = 0, \tag{19}$$

which we can solve by iterating

$$\pi_i' = \frac{\sum_j w_{ij}(\alpha \pi_i + \pi_j)/(\pi_i + \pi_j)}{\sum_j (\alpha w_{ij} + w_{ji})/(\pi_i + \pi_j)} \tag{20}$$

until convergence is achieved. When $\alpha = 1$ this procedure is equivalent to Zermelo's algorithm, Eq. (9). When $\alpha = 0$ it is equivalent to the algorithm presented in this paper, Eq. (12). For negative α the iteration does not generate positive values of π_i in general and hence is invalid, but for zero or positive α it gives a whole range of algorithms, all of which converge to the same maximum-likelihood solution as Zermelo's algorithm. When $0 \leq \alpha \leq 1$, convergence can be proved using a straightforward generalization of the method of Section 3 as follows.

Since the log-likelihood has only a single stationary point corresponding to the global maximum, and since any fixed point of (20) is a solution of (19) and hence corresponds to a stationary point of the log-likelihood, it follows that if (20) converges to a fixed point at all it must converge to the global likelihood maximum. To show that it converges to a fixed point it suffices, as previously, to show that the log-likelihood always increases upon application of (20) unless a fixed point has been reached. To do this we rewrite Eq. (13) for the terms in the log-likelihood that depend on π_i as

$$f(\pi_i) = \alpha \sum_j w_{ij} \log \pi_i + (1 - \alpha) \sum_j w_{ij} \log \frac{\pi_i}{\pi_i + \pi_j} - \sum_j (\alpha w_{ij} + w_{ji}) \log(\pi_i + \pi_j). \tag{21}$$

For any π_i, π_i' the inequality (15) implies that

$$\log \pi_i' \geq \log \pi_i - \frac{\pi_i}{\pi_i'} + 1 = \log \pi_i + \frac{\pi_i' - \pi_i}{\pi_i'}, \tag{22}$$

with the exact equality applying if and only if $\pi_i' = \pi_i$. Evaluating Eq. (21) at the point π_i' given by Eq. (20) and applying this inequality along with (16) and (17), we find for $0 \leq \alpha \leq 1$ that

$$\begin{aligned}
f(\pi_i') &= \alpha \sum_j w_{ij} \log \pi_i' + (1 - \alpha) \sum_j w_{ij} \log \frac{\pi_i'}{\pi_i' + \pi_j} - \sum_j (\alpha w_{ij} + w_{ji}) \log(\pi_i' + \pi_j) \\
&\geq \alpha \sum_j w_{ij} \left[\log \pi_i + \frac{\pi_i' - \pi_i}{\pi_i'} \right] + (1 - \alpha) \sum_j w_{ij} \left[\log \frac{\pi_i}{\pi_i + \pi_j} + \frac{(\pi_i' - \pi_i)/\pi_i'}{(\pi_i + \pi_j)/\pi_j} \right] \\
&\quad - \sum_j (\alpha w_{ij} + w_{ji}) \left[\log(\pi_i + \pi_j) + \frac{\pi_i' - \pi_i}{\pi_i + \pi_j} \right] \\
&= f(\pi_i) + (\pi_i' - \pi_i) \left[\frac{1}{\pi_i'} \sum_j w_{ij} \frac{\alpha \pi_i + \pi_j}{\pi_i + \pi_j} - \sum_j \frac{\alpha w_{ij} + w_{ji}}{\pi_i + \pi_j} \right] \\
&= f(\pi_i), \tag{23}
\end{aligned}$$

where we have employed Eq. (21) again, the term in square brackets in the penultimate line vanishes because of Eq. (20), and the exact equality applies if and only if $\pi_i' = \pi_i$. Thus $f(\pi_i)$ always increases upon application of (20) unless $\pi_i' = \pi_i$.

The rest of the proof follows the same lines as in Section 3 and hence convergence to the global likelihood maximum is established. As a corollary, this also provides an alternative proof of the convergence of Zermelo's algorithm (the case $\alpha = 1$) which is significantly simpler than the original proof given by Zermelo (1929) or the later proof by Ford (1957).

The method of proof used here does not extend to the case of $\alpha > 1$, because $1 - \alpha$ becomes negative and the inequality in (23) no longer follows from (16). It is still possible to prove convergence for $\alpha > 1$ but the proof is more involved. See Appendix A for details.

Numerical measurements, some of which are presented in Section 7, indicate that convergence of the algorithms of this section becomes monotonically slower with increasing α, so that the main algorithm presented in this paper, Eq. (12), which corresponds to the smallest allowed value of $\alpha = 0$, is the fastest, and it is on this case that we concentrate in the remainder of the paper. Some formal results on rates of convergence as a function of α are presented in Appendix B.

5. Prior on the strength parameters

Equation (12) provides a complete algorithm for fitting the Bradley-Terry model. In practice, however, pure maximum-likelihood fits such as this can be problematic for this model. In particular, as mentioned in Section 3, a likelihood maximum exists only if the network of interactions is strongly connected. If this condition is not met then the score parameters s_i will diverge and the algorithm of Eq. (12)—and indeed all maximum-likelihood methods for this model—will fail.

The root cause of this problem is that the maximum-likelihood fit effectively assumes a uniform (improper) prior on the s_i, which places all but a vanishing fraction of its weight on arbitrarily large values and, when coupled with a network that is not strongly connected, causes divergences. An effective solution is to impose a better-behaved prior on s_i and then compute a maximum a posteriori (MAP) estimate of the scores instead of a maximum-likelihood estimate (MLE). A range of priors have been proposed for this purpose (Davidson and Solomon, 1973; Caron and Doucet, 2012; Whelan, 2017) but arguably the most natural is a logistic prior. Recall from Section 2 that the probability p_1 of a player with strength π winning against the average player is $p_1 = \pi/(\pi + 1)$. In the absence of any evidence to the contrary, we assume this probability to be uniformly distributed between zero and one so that $P(p_1) = 1$, a least informative or maximum-entropy prior. Then the prior on the scores $s = \log \pi$ is

$$P(s) = P(p_1)\frac{dp_1}{ds} = \frac{dp_1}{d\pi}\frac{d\pi}{ds} = \frac{\pi}{(\pi + 1)^2} = \frac{1}{(e^s + 1)(e^{-s} + 1)}, \tag{24}$$

which is the logistic distribution. Combining this result with Eq. (5) we then get a posterior probability on the scores that is given, up to a multiplicative constant, by

$$P(s|W) \propto \prod_{ij} \left(\frac{e^{s_i}}{e^{s_i} + e^{s_j}} \right)^{w_{ij}} \prod_i \frac{1}{(e^{s_i} + 1)(e^{-s_i} + 1)}$$

$$= \prod_{ij} \left(\frac{\pi_i}{\pi_i + \pi_j} \right)^{w_{ij}} \prod_i \frac{\pi_i}{(\pi_i + 1)^2}. \tag{25}$$

Maximizing this posterior probability instead of the likelihood regularizes the values of the scores, preventing them from diverging. It also removes the invariance under multiplication of the π_i by a constant and hence eliminates the need to normalize them.

The iterative algorithm of Eq. (12) can be generalized straightforwardly to this MAP estimate. As observed by Whelan (2017), the prior for individual i can be thought of as

$$\frac{\pi_i}{(1 + \pi_i)^2} = \frac{\pi_i}{\pi_i + 1} \times \frac{1}{\pi_i + 1}, \tag{26}$$

which is precisely the probability that i plays two games against the average player (who has $\pi = 1$) and wins one of them and loses the other. Thus Eq. (25) can be thought of as the likelihood of a Bradley-Terry model in which we have added two fictitious games for each player, one won and one lost, and we can maximize this likelihood (and hence the posterior of Eq. (25)) using the same algorithm as before, merely adding these extra fictitious games to the data. This also means that our proof of convergence generalizes to the MAP case and that the network of interactions is now strongly connected, so the probability maximum always exists.

Alternatively, and perhaps more conveniently, we can derive an explicit algorithm for the MAP case by differentiating Eq. (25) with respect to π_i for any i, which leads to the iteration

$$\pi_i' = \frac{1/(\pi_i + 1) + \sum_j w_{ij}\pi_j/(\pi_i + \pi_j)}{1/(\pi_i + 1) + \sum_j w_{ji}/(\pi_i + \pi_j)}. \tag{27}$$

This is the generalization of Eq. (12) to the MAP case. It is completely equivalent to adding the fictitious games and has the same guaranteed convergence. One can also add the same prior to the traditional Zermelo algorithm of Eq. (9), which gives

$$\pi_i' = \frac{1 + \sum_j w_{ij}}{2/(\pi_i + 1) + \sum_j (w_{ij} + w_{ji})/(\pi_i + \pi_j)}. \tag{28}$$

In Section 7 we present the results of numerical experiments on the rate of convergence both of these MAP estimators and of the MLEs, using Eqs. (9), (12), (27), and (28).

6. Ties

Ties or draws can occur in certain types of competition, such as chess and soccer. There are a number of ways to generalize ranking calculations to include ties. The simplest is just to consider a tied game to be half of a win for each of the players. This approach is used for instance in the Elo chess rating system and can be trivially incorporated into

our calculations by modifying the values w_{ij}. A more sophisticated approach, however, incorporates the probability of a tie into the model itself. There is more than one way to do this (Rao and Kupper, 1967; Davidson, 1970). Here we employ the modification of the Bradley-Terry model proposed by Davidson (1970). One again defines strengths π_i for each player and the probabilities of a win p_{ij} and a tie q_{ij} between players i and j are

$$p_{ij} = \frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}, \qquad q_{ij} = \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}, \tag{29}$$

where $\nu > 0$ is a parameter which controls the overall frequency of ties and which we will estimate by maximum likelihood along with the strengths. Note that when $\pi_i = \pi_j$ we have $q_{ij} = \nu/(1+\nu)$ and hence $\nu = q_{ij}/(1 - q_{ij})$, so ν can be interpreted as the odds of a tie between evenly matched players.

The form (29) satisfies the obvious requirements that $p_{ij} + p_{ji} + q_{ij} = 1$ and $q_{ij} = q_{ji}$, and also has the intuitive property that the probability of a tie is greatest when the players are evenly matched and vanishes as π_i and π_j become arbitrarily far apart. As with the standard Bradley-Terry model, the probabilities p_{ij} and q_{ij} are invariant under multiplication of all π_i by a constant, and again we remove this ambiguity by normalizing them so that $\prod_i \pi_i = 1$.

Davidson (1970) proposed an iterative algorithm for computing maximum-likelihood estimates of the strengths and the parameter ν within this model. Defining w_{ij} as before to be the number of times i beats j and $t_{ij} = t_{ji}$ to be the number of ties, we can write the likelihood of a set of observations $W = [w_{ij}]$, $T = [t_{ij}]$ as

$$\begin{aligned} P(W, T|\pi, \nu) &= \prod_{ij} p_{ij}^{w_{ij}} \prod_{i<j} q_{ij}^{t_{ij}} \\ &= \prod_{ij} \left(\frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \right)^{w_{ij}} \prod_{i<j} \left(\frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \right)^{t_{ij}}, \end{aligned} \tag{30}$$

and the corresponding log-likelihood is

$$\begin{aligned} \log P(W, T|\pi, \nu) = \sum_{ij} \left(w_{ij} + \tfrac{1}{2}t_{ij} \right) \log \pi_i + \tfrac{1}{2} \log 2\nu \sum_{ij} t_{ij} \\ - \sum_{ij} \left(w_{ij} + \tfrac{1}{2}t_{ij} \right) \log \left(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j} \right). \end{aligned} \tag{31}$$

The combination $w_{ij} + \tfrac{1}{2}t_{ij}$ comes up repeatedly in the analysis so, following Davidson, we define the convenient shorthand $a_{ij} = w_{ij} + \tfrac{1}{2}t_{ij}$ and

$$\log P(W, T|\pi, \nu) = \sum_{ij} a_{ij} \log \pi_i + \tfrac{1}{2} \log 2\nu \sum_{ij} t_{ij} - \sum_{ij} a_{ij} \log \left(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j} \right). \tag{32}$$

Differentiating with respect to π_i and setting the result to zero gives

$$\frac{1}{\pi_i} \sum_j a_{ij} = \sum_j \left(a_{ij} + a_{ji} \right) \frac{1 + \nu\sqrt{\pi_j/\pi_i}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}. \tag{33}$$

10

This equation has no general closed-form solution for π_i but Davidson proposed solving it by the obvious iteration

$$\pi_i' = \frac{\sum_j a_{ij}}{\sum_j \left(a_{ij} + a_{ji}\right)\left(\dfrac{1 + \nu\sqrt{\pi_j/\pi_i}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}. \tag{34}$$

We can also calculate a maximum-likelihood estimate of the parameter ν by differentiating (32) with respect to ν to get

$$\frac{1}{2\nu}\sum_{ij} t_{ij} = \sum_{ij} a_{ij}\frac{2\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}, \tag{35}$$

which is again solved by iteration:

$$\nu' = \frac{\frac{1}{2}\sum_{ij} t_{ij}}{\sum_{ij} a_{ij}\left(\dfrac{2\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}. \tag{36}$$

Davidson used asynchronous updates in which one applies Eq. (34) to each π_i in turn, then applies (36) once to update ν, then repeats until convergence is achieved. This is a natural generalization of Zermelo's algorithm, Eq. (9), to situations where ties are allowed, and it includes Zermelo's algorithm as the special case when $\nu = 0$ and $t_{ij} = 0$. Davidson proved that the procedure always converges to the global likelihood maximum (when the maximum exists), but once again convergence can be slow in practice. Here we propose an alternative algorithm which generalizes Eq. (12) and is substantially faster.

Equation (33) can be rearranged in the form

$$\frac{1}{\pi_i}\sum_j a_{ij}\frac{\pi_j + \nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} = \sum_j a_{ji}\frac{1 + \nu\sqrt{\pi_j/\pi_i}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}, \tag{37}$$

which can be solved by iterating the equation

$$\pi_i' = \frac{\sum_j a_{ij}\left(\dfrac{\pi_j + \nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}{\sum_j a_{ji}\left(\dfrac{1 + \nu\sqrt{\pi_j/\pi_i}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}. \tag{38}$$

Similarly, writing $a_{ij} = w_{ij} + \frac{1}{2}t_{ij}$, Eq. (35) can be rearranged in the form

$$\frac{1}{2\nu}\sum_{ij} t_{ij}\frac{\pi_i + \pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} = \sum_{ij} w_{ij}\frac{2\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}, \tag{39}$$

which can be solved by iterating

$$\nu' = \frac{\frac{1}{2}\sum_{ij} t_{ij}\left(\dfrac{\pi_i + \pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}{\sum_{ij} w_{ij}\left(\dfrac{2\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right)}. \tag{40}$$

Equations (38) and (40) are the appropriate generalization of (12) to the case with ties and they include (12) as the special case when $\nu = 0$ and $t_{ij} = 0$. Again we recommend applying the equations asynchronously: one cycle of the algorithm involves updating each π_i in turn using Eq. (38) then applying Eq. (40) once to update ν, and repeating until convergence is achieved. As we show in Section 7, this procedure converges significantly faster than Davidson's algorithm.

The proof that Eqs. (38) and (40) do in fact converge to the likelihood maximum follows similar lines to that for the case without ties but the algebra is tedious so we omit it here. The interested reader can find it in Appendix C.

7. Results

The iterations (12) and (27) (for the case without ties) and (38) and (40) (with ties) converge significantly faster in typical applications than the traditional algorithm of Zermelo or its extension for the case where ties are allowed. In this section we illustrate the convergence rates with a selection of example applications to both real and synthetic data.

7.1 Computer-generated data

As our first example, we apply our algorithms to a collection of random computer-generated data sets. In these calculations we generated synthetic test data with $N = 1000$ players and $M = 50\,000$ games, for an average of 100 games per player. The players for each game are chosen uniformly at random (with replacement) and the winners of the games are chosen using the Bradley-Terry model itself: scores s_i for each player are drawn from a logistic distribution $P(s) = 1/[(e^s + 1)(e^{-s} + 1)]$ and then the winner of each game is chosen at random according to the probability p_{ij} of Eq. (1). In cases where the resulting network of interactions is not strongly connected, games are discarded and redrawn until a strongly connected network is achieved, to ensure that a likelihood maximum does exist as discussed in Section 3.

Figure 1a shows the results of applying both Zermelo's algorithm and the algorithm of this paper to one such synthetic network. Initial values of π_i were chosen randomly such that s_i is drawn from the logistic distribution $1/[(e^s + 1)(e^{-s} + 1)]$. (Other methods of choosing the initial values have been proposed and may improve performance in some cases (Dykstra, 1956), but we avoid these here to separate effects of the different algorithms from effects of the initial values.) Figure 1a shows how the log-likelihood, Eq. (6), converges to its final value on successive iterations of each algorithm. As we can see, the algorithm of this paper (top curve, solid points) comes close to the final value of the log-likelihood after only two iterations, while Zermelo's algorithm (bottom curve, open points) takes significantly longer. The other curves in the figure (gray points) show values for various algorithms in the family defined in Section 4, parametrized by the quantity α as in Eq. (20), and it appears that convergence becomes monotonically slower with increasing α, as mentioned previously in Section 4.

Arguably, however, Fig. 1a fails to truly show how much faster the algorithm of this paper is. Figure 1b gives an alternative view. In this plot we show the root-mean-square (RMS) difference (averaged over all players) between the current estimate of the probability $p_1 = \pi_i/(\pi_i + 1)$ of beating the average player and the final converged value. We choose this
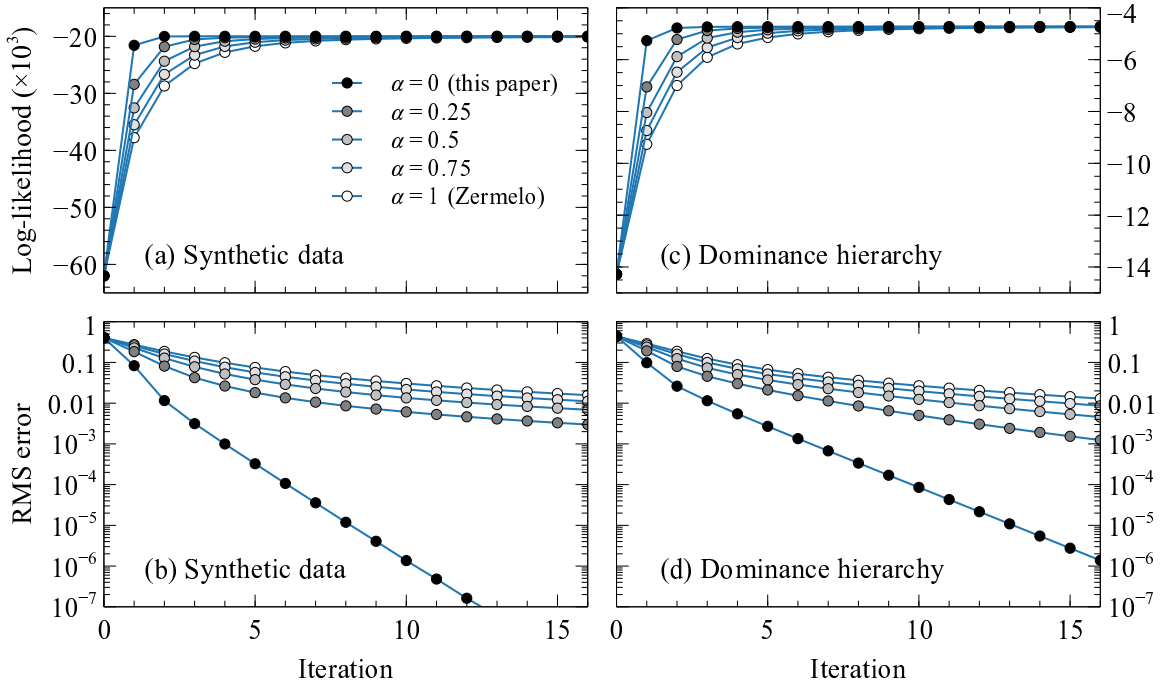
Figure 1: Convergence of the iterative algorithms studied here. (a) The log-likelihood, Eq. (6), for a synthetic network of 1000 players and 50 000 games. The plot shows the value on successive iterations for the algorithm of this paper, Zermelo's algorithm, and a selection of algorithms from the family defined in Section 4. (b) The root-mean-square (RMS) deviation from the final maximum-likelihood solution for the same synthetic data set. (c) The log-likelihood for the dominance hierarchy of vervet monkeys described in Section 7.2. (d) The RMS deviation for the same dominance hierarchy data set.

quantity because the probability of beating the average player is typically more uniformly distributed than the π_i themselves and hence the average over players is better behaved, minimizing effects of fluctuations and dependency on outliers. As the figure shows this quantity converges enormously faster for our algorithm than for Zermelo's algorithm, being already more than a thousand times better than the value for Zermelo's algorithm after less than ten iterations. Indeed there is a significant gulf in the speed of convergence between our algorithm and all of the others shown in the figure: for all the nonzero values of α convergence is much slower than it is for $\alpha = 0$.

To make these comparisons more quantitative, we have conducted extensive further tests using synthetic data of the type described here. In these tests we first iterate Eq. (12) until it converges to high precision, effectively solving for the maximum-likelihood solution at or close to the limits of numerical accuracy. Then, using either the Zermelo algorithm or our new algorithm, we measure the number of iterations needed to converge to this final

solution within a prescribed level of accuracy. Specifically we require that the probability $\pi_i/(\pi_i + 1)$ of beating the average player converge to within 10^{-6} from its final value for all i. This criterion is more stringent—and arguably more realistic—than criteria based on convergence of the value of the log-likelihood (Vojnovic et al., 2019).

For tests of the algorithm of Section 5 for MAP estimates with a logistic prior the same procedure was used to generate data and we compare the convergence of the new algorithm of Eq. (27) against the generalized Zermelo algorithm of Eq. (28). For the case with ties the procedure was similar, but wins, losses, and ties were generated according to the probabilities p_{ij} and q_{ij} of Eq. (29) with $\nu = \frac{1}{2}$ and we compare the convergence rate of Eqs. (38) and (40) against Eqs. (34) and (36) with an initial value of $\nu = 1$ in all cases.

All tests were averaged over 100 randomly generated data sets and the results are reported in Table 1. As these results show, the algorithm of this paper is much faster than Zermelo's algorithm. For the standard maximum likelihood estimate (the most common application) the difference is spectacular: the new algorithm is over a hundred times faster. Where Zermelo's algorithm takes an average of more than 1200 iterations to converge, the new algorithm takes just 12. For the MAP estimates the difference in running time is less extreme but still large—the new algorithm is over eight times faster than Zermelo's algorithm—while for the case with ties the new algorithm is an impressive 42 times faster.

7.2 Real-world data

In this section we present example applications to several real-world data sets and show that our algorithm also offers significant speed improvements in these settings. The data sets we study are as follows.

Wolves: A typical animal dominance hierarchy data set describing observations of subordinate behaviors among members of a family of 15 captive wolves in Arnhem, Netherlands as reported by van Hooff and Wensing (1987).

Vervet monkeys: A larger dominance hierarchy data set describing observations of agonistic interactions of various kinds among 63 wild vervet monkeys in the Samara Private Game Reserve in South Africa, as reported by Vilette et al. (2020). The original data set had 66 monkeys, but three were removed in order to ensure that the network of interactions was strongly connected, as discussed in Section 3.

American football: As an example of an application to sports competition, this data set describes professional American football games played in the US National Football League during a single season. Unlike association football, American football proceeds by a series of discrete plays in which the team currently in possession of the ball attempts to advance it up the field. This data set consists of individual plays in all games between the 32 teams in the league during the 2016 regular season, as compiled by Yurko et al. (2019). Only passing plays, running plays, punts, sacks, and field goals were used in the analysis. Other plays such as kickoffs and conversions were excluded. The team in possession of the ball is considered to have won a play if either (a) they score points on the play or (b) they advance the ball and retain possession; otherwise the other team wins the play.

Political figures: The results of an online paired comparison survey conducted by the Washington Post newspaper in 2010, in which readers were presented with pairs of promi-

| | Data | N | M | Iterations to reach convergence | | |
				Zermelo	This paper	Speed-up
Without ties MLE	Synthetic	1000	50 000	1270 ± 470	12 ± 2	$\times 104$
	Wolves	15	10 382	2410 ± 10	145 ± 1	$\times 17$
	Vervet monkeys	63	11 621	232 ± 8	19 ± 1	$\times 12$
	American football	32	35 741	49 ± 3	15 ± 1	$\times 3.4$
	Political figures	67	76 632	54 ± 3	8 ± 5	$\times 7.1$
	Photographs	9097	247 531	911 ± 4	22 ± 0	$\times 41$
Without ties MAP	Synthetic	1000	50 000	1560 ± 40	185 ± 18	$\times 8.5$
	Wolves	15	10 382	$49\,200 \pm 1700$	2200 ± 110	$\times 22$
	American football	32	35 741	$19\,200 \pm 1500$	6000 ± 600	$\times 3.3$
	Photographs	9097	247 531	1186 ± 3	82 ± 10	$\times 14$
With ties	Synthetic	1000	50 000	1130 ± 760	27 ± 8	$\times 42$
	Soccer	177	898	1650 ± 16	421 ± 5	$\times 3.9$
	School students	2155	8970	2770 ± 10	613 ± 1	$\times 4.5$
	Chess	14 852	623 727	1750 ± 90	162 ± 9	$\times 11$

Table 1: The number of iterations required for the algorithms discussed in this paper to converge in applications to real and synthetic (computer-generated) data. Results are averaged over 100 runs and rounded to three figures. N is the number of individuals or teams being ranked, M is the total number of interactions among all individuals, and the figures following "$\pm$" are standard deviations about the mean, giving an indication of the amount of variation in the results. "Speed-up" is the average factor by which the method of this paper improves upon the traditional Zermelo algorithm, or its generalizations for the MAP case and the case where ties are allowed.

nent political figures and asked to judge which had had the worse week in politics. The data were made available on the survey platform allourideas.org.

Photographs: Results from the IMDB-Wiki-SbS study of Pavlichenko and Ustalov (2021), a paired comparison study that asked participants to judge people's age from photographs. Participants were presented with 247 531 pairs of faces drawn from a pool of 9097 photographs and asked to judge which of the people depicted was older. In principle, a ranking of the results should then be able to order the people from (apparent) oldest to youngest. A small number of images were excluded from the data set for our calculations to ensure a strongly connected network.

Soccer: Wins, losses, and draws in 898 men's international association football matches between 177 different countries during the year 2011. Data from Mart Jürisoo at kaggle.com/martj42. The original network of matches was not strongly connected, so the data analyzed here represent only the largest strongly connected component of the network.

School students: These data describe declared friendships among 2155 students in a large US high school and its feeder middle school, from the National Longitudinal Study

of Adolescent Health (the "Add Health" study, Udry et al. 1997). If student i states that they are friends with student j but j does not reciprocate (something that occurs often in these data) we consider it a win for j; if i and j both state they are friends we consider it a tie. Although in principle friendships are not competitive, there is evidence that friendship patterns among school students do describe a clear hierarchy because students tend to claim friendship with others who have higher social status than themselves (Hallinan and Kubitschek, 1988). Thus ranking calculations applied to data like these can be used to infer social status (Ball and Newman, 2013). Treating a reciprocated friendship as a tie is arguably more correct than treating it as two separate wins: reciprocated friendships clearly violate the assumption of independence in the Bradley-Terry model without ties, since the two wins never go in the same direction, but there is no equivalent violation for the model with ties. The same approach could also be applied to other social networks that show similar reciprocity properties. The network of friendships for this data set was not strongly connected, so the data analyzed here represent only the largest strongly connected component of the network.

Chess: Wins, losses, and draws in chess matches between $14\,852$ expert players on the online chess server lichess.com during the month of July 2016. For a match to be included, both players must have had Elo ratings of 2000 or higher at the time of the match. A small number of players were removed to ensure the network of matches was strongly connected. The data are from lichess.com via kaggle.com/arevel. With over $600\,000$ matches, this is the largest data set considered here.

Figures 1c and 1d show an example of the convergence of the log-likelihood and RMS error during a single run using the vervet monkey data. The behavior is similar to that for the synthetic data in Figs. 1a and 1b: the log-likelihood converges most rapidly for the algorithm of this paper and significantly more slowly for Zermelo's algorithm. Other algorithms from the family defined in Section 4 fall between the two, and convergence appears to become monotonically slower as the parameter α of Eq. (20) increases. The RMS error shown in Fig. 1d once again shows very rapid convergence for the algorithm of this paper. All the other algorithms are substantially slower by this measure.

Complete results on time to convergence for the various data sets are presented in Table 1. The methodology for these calculations was the same as for the synthetic data: the parameters were first converged to high precision, then the results used to estimate the time to convergence in a second run of the calculation. Each calculation was replicated 100 times with random initial conditions in which the s_i were drawn from a logistic distribution as previously.

The overall picture for these runs is again similar to that for the synthetic data. In all cases the method of this paper outpaces the traditional Zermelo algorithm. For instance, for maximum-likelihood estimates in cases without ties the new algorithm is 17 times faster on the smallest example, the dominance hierarchy of wolves, while on the largest example, the photographs, it is a remarkable 41 times faster. The smallest difference is for the American football data set, for which the new algorithm is 3.4 times faster than Zermelo's algorithm. For MAP estimates the numbers are similar: the new algorithm is a factor of 22, 14, and 3.3 times faster respectively on these three data sets.

As with the synthetic data, the speed difference on the tests with ties is less dramatic though still substantial, with the new algorithm being about 4 to 11 times faster. Convergence was also somewhat slower overall for both algorithms in the case with ties, although this may have more to do with the fact that these data sets are sparser (which tends to slow convergence) than with the presence of ties. Notice that convergence of our algorithm is very fast for the synthetic data with ties, which is relatively dense.

These effects can make a substantial difference to running times in practice. For the dominance hierarchy of wolves, for instance, a single run of Zermelo's algorithm (implemented in the Python programming language on an up-to-date but otherwise unremarkable personal computer *circa* 2022) converges to the maximum-likelihood solution in a running time of about 1 minute. The algorithm of this paper, by contrast, takes 3 seconds. For the more demanding photograph data set, Zermelo's algorithm takes over 8 minutes; the method of this paper takes just 11 seconds. For larger applications still, such as to web data or online social networks, the difference could become very significant.

All the results of this section are numerical. Ideally we would like to be able to prove formally that the algorithms presented in this paper converge faster than Zermelo's algorithm. At present we do not have such a proof but we can show certain results. As demonstrated in Appendix B, we can prove that within the one-parameter family of algorithms defined in Section 4, all those for $\alpha > 1$ converge slower than Zermelo's algorithm (the case $\alpha = 1$), which means these are not normally of interest. We can also prove that convergence becomes monotonically faster with decreasing α down to some point $\alpha < 1$, meaning that there provably exist algorithms that are faster than Zermelo's algorithm. In general, however, the proof does not extend to $\alpha = 0$ (the algorithm of this paper), so for the moment the finding that convergence is fastest for $\alpha = 0$ is a numerical one only.

8. Conclusions

We have presented an alternative to the classic algorithm of Zermelo for computing rankings from pairwise comparisons using fits to the Bradley-Terry model, with or without ties allowed. Like Zermelo's algorithm, the method presented is a simple iterative scheme. We have proved that the iteration always converges to the global maximum of the likelihood and given numerical evidence that it does so faster—typically many times faster—than Zermelo's algorithm. Given that it is also simple to implement we know of no reason not to favor the algorithm presented here over Zermelo's algorithm.

Acknowledgments

Appendix A: Proof of convergence for $\alpha > 1$

As discussed in Section 4, Zermelo's algorithm and the algorithm of this paper are both special cases of a larger one-parameter family of algorithms given by the iteration of

$$\pi_i' = \frac{\sum_j w_{ij}(\alpha\pi_i + \pi_j)/(\pi_i + \pi_j)}{\sum_j(\alpha w_{ij} + w_{ji})/(\pi_i + \pi_j)} \tag{41}$$

for any $\alpha \geq 0$. For $0 \leq \alpha \leq 1$ the convergence of this iteration to the likelihood maximum can be proved straightforwardly as described in Section 4. For $\alpha > 1$ the same method of proof does not work because $1 - \alpha$ becomes negative and the inequality in (23) no longer follows from (16). It is still possible to prove convergence but the method of proof is somewhat different, as we now describe.

From (15) we have for any $x, y, c > 0$

$$\log(x + c) \geq \log(y + c) - \frac{y + c}{x + c} + 1 = \log(y + c) + \frac{x - y}{x + c}$$

$$= \log(y + c) + \frac{x - y}{x} - \frac{(x - y)/x}{(y + c)/c} + \frac{c(x - y)^2}{x(x + c)(y + c)}$$

$$\geq \log(y + c) + \frac{x - y}{x} - \frac{(x - y)/x}{(y + c)/c}, \tag{42}$$

which is equivalent to

$$\log x - \log y - \frac{x - y}{x} \geq \log\frac{x}{x + c} - \log\frac{y}{y + c} - \frac{(x - y)/x}{(y + c)/c}, \tag{43}$$

with the exact equality applying if and only if $x = y$. Noting that the left-hand side of this inequality is always positive by (15), for any $\alpha > 1$ we then have

$$\log x - \log y - \frac{x - y}{x} \geq \frac{\alpha - 1}{\alpha}\left[\log\frac{x}{x + c} - \log\frac{y}{y + c} - \frac{(x - y)/x}{(y + c)/c}\right], \tag{44}$$

which can be rearranged to read

$$\alpha \log x + (1 - \alpha)\log\frac{x}{x + c} \geq \alpha\left[\log y + \frac{x - y}{x}\right] + (1 - \alpha)\left[\log\frac{y}{y + c} + \frac{(x - y)/x}{(y + c)/c}\right]. \tag{45}$$

Now setting $x = \pi_i'$, $y = \pi_i$, and $c = \pi_j$, multiplying by the positive quantities w_{ij}, and summing, we have

$$\alpha \sum_j w_{ij}\log\pi_i' + (1 - \alpha)\sum_j w_{ij}\log\frac{\pi_i'}{\pi_i' + \pi_j} \geq \alpha\sum_j w_{ij}\left[\log\pi_i + \frac{\pi_i' - \pi_i}{\pi_i'}\right]$$

$$+ (1 - \alpha)\sum_j w_{ij}\left[\log\frac{\pi_i}{\pi_i + \pi_j} + \frac{(\pi_i' - \pi_i)/\pi_i'}{(\pi_i + \pi_j)/\pi_j}\right], \tag{46}$$

where the exact equality applies if and only if $\pi_i' = \pi_i$. In combination with (17), this is now sufficient to establish the inequality in (23) once again, and hence convergence of the algorithm for $\alpha > 1$ is assured.

18

Appendix B: Rate of convergence

The numerical results of Section 7 show markedly faster convergence for the algorithms of this paper than for the standard Zermelo algorithm. As discussed at the end of Section 7, we do not at present have a proof that convergence is faster, but it is possible to prove that some algorithms within the family defined in Section 4 converge faster than Zermelo's algorithm.

As observed in Fig. 1, the iterative algorithms of this paper show exponential convergence, which is expected—in general all iterations of the form $x' = f(x)$ converge exponentially, if they converge at all, except in certain special cases that do not apply here. For the family of algorithms in Section 4 the rate of convergence for any given value of the parameter α can be quantified by the factor $\lambda_i(\alpha)$ by which the distance between the current estimate of π_i and the final maximum-likelihood estimate (MLE) $\hat{\pi}_i$ decreases when π_i is updated, as π_i approaches $\hat{\pi}_i$. Thus

$$\lambda_i(\alpha) = \lim_{\pi \to \hat{\pi}} \frac{\pi'_i - \hat{\pi}_i}{\pi_i - \hat{\pi}_i} = \left(\frac{\partial \pi'_i}{\partial \pi_i} \right)_{\hat{\pi}}, \tag{47}$$

where the subscript $\hat{\pi}$ indicates that the derivative is evaluated at the MLE. For instance, for Zermelo's algorithm (the case $\alpha = 1$), applying Eq. (3) we have

$$\lambda_i(1) = \frac{\sum_j w_{ij} \sum_j (w_{ij} + w_{ji})/(\hat{\pi}_i + \hat{\pi}_j)^2}{\left[\sum_i (w_{ij} + w_{ji})/(\hat{\pi}_i + \hat{\pi}_j) \right]^2} = \frac{1}{\sum_j w_{ij}} \sum_j (w_{ij} + w_{ji}) \left(\frac{\hat{\pi}_i}{\hat{\pi}_i + \hat{\pi}_j} \right)^2, \tag{48}$$

where we have employed (3) again to simplify the expression and made use of the fact that $\pi'_i = \pi_i = \hat{\pi}_i$ at the MLE. Assuming once again that the network of interactions represented by w_{ij} is strongly connected, the w_{ij} are strictly positive for all i, j. As shown by Ford (1957), this implies that the $\hat{\pi}_i$ are strictly positive and finite, which means in turn that the value of $\lambda_i(1)$ is strictly positive. For other α, however, the value of $\lambda_i(\alpha)$ can be negative (meaning that convergence to the MLE is oscillatory). This observation will be important in a moment.

The factor by which the RMS error of Fig. 1 decreases over a complete round of updates depends asymptotically on the slowest decaying π_i and is given by

$$\lambda_{\max}(\alpha) = \max_i |\lambda_i(\alpha)|, \tag{49}$$

where we take the absolute value to allow for the possibility of negative λ_i. An algorithm with given α asymptotically converges faster than Zermelo's algorithm if $\lambda_{\max}(\alpha) < \lambda_{\max}(1)$. Here we demonstrate that this is the case for at least some values of α.

We consider how the value $\lambda_i(\alpha)$ changes with α and compute the derivative

$$\frac{\partial \lambda_i}{\partial \alpha} = \frac{\partial}{\partial \alpha} \left(\frac{\partial \pi'_i}{\partial \pi_i} \right)_{\hat{\pi}} = \left(\frac{\partial^2 \pi'_i}{\partial \pi_i \partial \alpha} \right)_{\hat{\pi}}. \tag{50}$$

From Eq. (20) we have

$$\frac{\partial \pi_i'}{\partial \alpha} = \left[\sum_j \frac{\alpha w_{ij} + w_{ji}}{\pi_i + \pi_j} \sum_j w_{ij} \frac{\pi_i}{\pi_i + \pi_j} - \sum_j w_{ij} \frac{\alpha \pi_i + \pi_j}{\pi_i + \pi_j} \sum_j \frac{w_{ij}}{\pi_i + \pi_j}\right] \bigg/ \left[\sum_j \frac{\alpha w_{ij} + w_{ji}}{\pi_i + \pi_j}\right]^2$$

$$= \frac{\sum_j w_{ij}/(\pi_i + \pi_j)}{\sum_j (\alpha w_{ij} + w_{ji})/(\pi_i + \pi_j)}(\pi_i - \pi_i'), \tag{51}$$

where we have used (20) again in the second line. Differentiating with respect to π_i, setting $\pi_i = \hat{\pi}_i$ for all i, and noting again that $\pi_i' = \pi_i$ at the MLE, we find that

$$\frac{\partial \lambda_i}{\partial \alpha} = \left(\frac{\partial^2 \pi_i'}{\partial \pi_i \partial \alpha}\right)_{\hat{\pi}} = \frac{\sum_j w_{ij}/(\hat{\pi}_i + \hat{\pi}_j)}{\sum_j (\alpha w_{ij} + w_{ji})/(\hat{\pi}_i + \hat{\pi}_j)}\big[1 - \lambda_i(\alpha)\big], \tag{52}$$

where we have used Eq. (47).

The fact that the iteration of Eq. (20) converges to the MLE for all $\alpha \geq 0$ implies that $\lambda_i(\alpha)$ must be strictly less than 1 for all i and hence (52) is strictly positive, since w_{ij} and $\hat{\pi}_i$ are strictly positive. At the same time it is also finite, and hence $\lambda_i(\alpha)$ is increasing in α and continuous for all $\alpha \geq 0$.

This now establishes some useful results. First, it implies that $\lambda_i(\alpha) > \lambda_i(1)$ for all i when $\alpha > 1$ (and also that $\lambda_i(\alpha)$ is positive in this regime). Thus, if the largest value of $\lambda_i(1)$ occurs for $i = \mu$, then

$$\lambda_{\max}(\alpha) \geq \lambda_\mu(\alpha) > \lambda_\mu(1) = \lambda_{\max}(1). \tag{53}$$

Hence all algorithms with $\alpha > 1$ converge slower than Zermelo's algorithm. For this reason these algorithms are not normally of practical interest.

Second, we also have $\lambda_i(\alpha) < \lambda_i(1)$ for all i when $0 \leq \alpha < 1$. Unfortunately, this is not sufficient to establish that $\lambda_{\max}(\alpha) < \lambda_{\max}(1)$ in this regime (and hence that these algorithms converge faster than Zermelo's algorithm) because, as mentioned above, it is not guaranteed that $\lambda_i(\alpha)$ is positive. The value of $\lambda_i(\alpha)$ for $\alpha < 1$ can—and in practice often does—become negative. This means that $|\lambda_i(\alpha)|$ could be larger than $\lambda_i(1)$ and indeed it is straightforward to find cases where this occurs.

On the other hand, we can prove that there exist *some* algorithms that are faster than Zermelo's. Given that $\lambda_i(\alpha)$ is continuous and increasing in α, its value must diminish smoothly and monotonically from $\alpha = 1$ all the way down to $\alpha = 0$. Thus, given that $\lambda_i(1)$ is strictly positive, one of two things must happen: either $\lambda_i(\alpha)$ never reaches the line $\lambda_i(\alpha) = -\lambda_i(1)$, in which case $|\lambda_i(0)| < \lambda_i(1)$, or it *does* reach this line, in which case there exists some $c_i < 1$ such that $\lambda_i(c_i) = -\lambda_i(1)$. In this case, by continuity, $|\lambda_i(\alpha)| < \lambda_i(1)$ in the non-vanishing interval $c_i < \alpha < 1$.

Now we repeat the same argument for all i and define $c = \max_i c_i$, or $c = 0$ if $|\lambda_i(0)| < \lambda_i(1)$ for all i, and then for all i we have $|\lambda_i(\alpha)| < \lambda_i(1)$ in the non-vanishing interval $c < \alpha < 1$. Now choose any α in this interval and suppose the largest value of $|\lambda_i(\alpha)|$ occurs for $i = \nu$. Then at this α we have

$$\lambda_{\max}(\alpha) = |\lambda_\nu(\alpha)| < \lambda_\nu(1) \leq \lambda_{\max}(1). \tag{54}$$

Hence all algorithms with $c < \alpha < 1$ converge faster than Zermelo's algorithm. Algorithms with $0 \leq \alpha \leq c$ may also converge faster than Zermelo's algorithm—and the numerical evidence suggests that they do—but this cannot be proved using the present approach.

Appendix C: Proof of convergence for the model with ties

For the case where ties are allowed, the proof that iteration of Eqs. (38) and (40) converges to the maximum of the log-likelihood (32) follows similar lines to that for the case without ties. Davidson (1970) proved that the likelihood has only a single stationary point with respect to its parameters, corresponding to the global likelihood maximum, provided the π_i are normalized and the network of interactions is strongly connected (with a tie counting as an edge in both directions between the relevant pair of players). Since any fixed point of Eqs. (38) and (40) corresponds to a stationary point of the likelihood, this implies that if our iteration converges to a fixed point at all then that point is the global maximum. To prove that we converge to a fixed point it suffices to show that the log-likelihood always increases upon application of either Eq. (38) or Eq. (40), unless a fixed point has been reached.

The terms in the log-likelihood of Eq. (32) that depend on π_i can be written in the form

$$
f(\pi_i) = \sum_j a_{ij} \log \frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - \sum_j a_{ji} \log\big(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}\big), \tag{55}
$$

where $a_{ij} = w_{ij} + \frac{1}{2}t_{ij}$ as previously. Applying the inequalities (14) and (15), we have for any π_i and π_i'

$$
\begin{aligned}
\log \frac{\pi_i'}{\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j}} &\geq \log \frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - \frac{\pi_i/(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j})}{\pi_i'/(\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j})} + 1 \\
&= \log \frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} + \left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i'}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \\
&\quad + \left(\frac{\pi_i' - \pi_i}{\pi_i'}\right) \frac{\pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}
\end{aligned} \tag{56}
$$

and

$$
\begin{aligned}
\log\big(\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j}\big) &\leq \log\big(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}\big) + \frac{\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - 1 \\
&= \log\big(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}\big) + \left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \\
&\quad + \frac{\pi_i' - \pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}.
\end{aligned} \tag{57}
$$

Evaluating Eq. (55) at the point π_i' defined by Eq. (38) and applying these two inequalities, we have

$$
\begin{aligned}
f(\pi_i') &= \sum_j a_{ij} \log \frac{\pi_i'}{\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j}} - \sum_j a_{ji} \log\left(\pi_i' + \pi_j + 2\nu\sqrt{\pi_i'\pi_j}\right) \\
&\geq \sum_j a_{ij} \left[\log \frac{\pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} + \left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i'}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \right. \\
&\qquad\qquad \left. + \left(\frac{\pi_i' - \pi_i}{\pi_i'}\right) \frac{\pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right] \\
&\quad - \sum_j a_{ji} \left[\log\left(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}\right) + \left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \right. \\
&\qquad\qquad \left. + \frac{\pi_i' - \pi_i}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right] \\
&= f(\pi_i) + \sum_{ij} a_{ij} \left[\left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i'}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - \left(\frac{\pi_i' - \pi_i}{\pi_i'}\right) \frac{\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right] \\
&\quad - \sum_j a_{ji} \left[\left(\frac{\sqrt{\pi_i'} - \sqrt{\pi_i}}{\sqrt{\pi_i}}\right) \frac{2\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - \left(\frac{\pi_i' - \pi_i}{\pi_i}\right) \frac{\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}}\right] \\
&= f(\pi_i) + \frac{(\sqrt{\pi_i'} - \sqrt{\pi_i})^2}{\pi_i'} \sum_j a_{ij} \frac{\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \\
&\quad + \frac{(\sqrt{\pi_i'} - \sqrt{\pi_i})^2}{\pi_i} \sum_j a_{ji} \frac{\nu\sqrt{\pi_i\pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} \\
&\geq f(\pi_i), \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (58)
\end{aligned}
$$

where we have used Eq. (38) and the exact equality applies if and only if $\pi_i' = \pi_i$. Hence $f(\pi_i)$ always increases upon application of Eq. (38) unless $\pi_i' = \pi_i$, and so therefore does the log-likelihood as well.

The same is also true of the update (40) for the parameter ν. The terms in the log-likelihood that depend on ν can be written

$$
g(\nu) = \tfrac{1}{2} \sum_{ij} t_{ij} \log \frac{\nu}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}} - \sum_{ij} w_{ij} \log(\pi_i + \pi_j + 2\nu\sqrt{\pi_i\pi_j}). \qquad (59)
$$

For any ν, ν' the inequalities (14) and (15) imply that

$$\log \frac{\nu'}{\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j}} \geq \log \frac{\nu}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}} - \frac{\nu/(\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j})}{\nu'/(\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j})} + 1$$

$$= \log \frac{\nu}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}} + \left(\frac{\nu' - \nu}{\nu'}\right)\frac{\pi_i + \pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}}, \quad (60)$$

$$\log(\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j}) \leq \log(\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}) + \frac{\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}} - 1$$

$$= \log(\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}) + (\nu' - \nu)\frac{2\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}}. \quad (61)$$

Evaluating (59) at the point ν' given by Eq. (40) and applying these two inequalities we get

$$g(\nu') = \tfrac{1}{2}\sum_{ij} t_{ij} \log \frac{\nu'}{\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j}} - \sum_{ij} w_{ij}\log(\pi_i + \pi_j + 2\nu'\sqrt{\pi_i \pi_j})$$

$$\geq \tfrac{1}{2}\sum_{ij} t_{ij}\left[\log \frac{\nu}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}} + \left(\frac{\nu' - \nu}{\nu'}\right)\frac{\pi_i + \pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}}\right]$$

$$- \sum_{ij} w_{ij}\left[\log(\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}) + (\nu' - \nu)\frac{2\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}}\right]$$

$$= g(\nu) + (\nu' - \nu)\left[\frac{1}{2\nu'}\sum_{ij} t_{ij}\frac{\pi_i + \pi_j}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}} - \sum_{ij} w_{ij}\frac{2\sqrt{\pi_i \pi_j}}{\pi_i + \pi_j + 2\nu\sqrt{\pi_i \pi_j}}\right],$$

$$= g(\nu), \quad (62)$$

where the term in square brackets in the penultimate line vanishes because of Eq. (40) and the exact equality applies if and only if $\nu' = \nu$. Thus $g(\nu)$ always increases upon application of (40) unless $\nu' = \nu$, and so therefore does the log-likelihood.

The remainder of the proof follows the same lines of argument as in Section 3 and hence convergence of Eqs. (38) and (40) to the unique likelihood maximum is established.

References

A. Agarwal, P. Patil, and S. Agarwal. Accelerated spectral ranking. *Proceedings of Machine Learning Research*, 80:70–79, 2018.

B. Ball and M. E. J. Newman. Friendship networks and social status. *Network Science*, 1: 16–30, 2013.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

F. Caron and A. Doucet. Efficient Bayesian inference for generalized Bradley-Terry models. *Journal of Computational and Graphical Statistics*, 21:174–196, 2012.

M. Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, 27:412–433, 2012.

H. A. David. *The Method of Paired Comparisons*. Griffin, London, 2 edition, 1988.

R. R. Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65:317–328, 1970.

R. R. Davidson and P. H. Farquhar. A bibliography on the method of paired comparisons. *Biometrics*, 32:241–252, 1976.

R. R. Davidson and D. L. Solomon. A Bayesian approach to paired comparison experimentation. *Biometrika*, 60:477–487, 1973.

O. Dykstra, Jr. A note on the rank analysis of incomplete block designs. *Biometrics*, 12: 301–306, 1956.

L. R. Ford, Jr. Solution of a ranking problem from binary comparisons. *American Mathematical Monthly*, 64(8):28–33, 1957.

M. T. Hallinan and W. N. Kubitschek. The effect of individual and structural characteristics on intransitivity in social networks. *Social Psychology Quarterly*, 51:81–92, 1988.

D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, 32:384–406, 2004.

L. Maystre and M. Grossglauser. Fast and accurate inference of Plackett-Luce models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Proceedings of the 2015 Conference on Advances in Neural Information Processing Systems*, pages 172–180, Cambridge, MA, 2015. MIT Press.

S. Negahban, S. Oh, and D. Shah. Rank centrality: Ranking from pair-wise comparisons. *Operations Research*, 65:266–287, 2017.

N. Pavlichenko and D. Ustalov. IMDB-WIKI-SbS: An evaluation dataset for crowdsourced pairwise comparisons. Preprint arxiv:2110.14990, 2021.

P. V. Rao and L. L. Kupper. Ties in paired-comparison experiments: A generalization of the Bradley-Terry model. *Journal of the American Statistical Association*, 62:194–204, 1967.

J. R. Udry, P. S. Bearman, and K. M. Harris. National Longitudinal Study of Adolescent Health, 1997. This research uses data from Add Health, a program project directed by Kathleen Mullan Harris and designed by J. Richard Udry, Peter S. Bearman, and Kathleen Mullan Harris at the University of North Carolina at Chapel Hill, and funded by grant P01–HD31921 from the Eunice Kennedy Shriver National Institute of Child Health and Human Development, with cooperative funding from 23 other federal agencies and foundations. Special acknowledgment is due Ronald R. Rindfuss and Barbara Entwisle for assistance in the original design. Information on how to obtain the Add Health data

J. A. R. A. M. van Hooff and J. A. B. Wensing. Dominance and its behavioral measures in a captive wolf pack. In H. Frank, editor, *Man and Wolf*, pages 219–252. Junk Publishers, Dordrecht, 1987.

C. Vilette, T. Bonnell, P. Henzi, and L. Barrett. Comparing dominance hierarchy methods using a data-splitting approach with real-world data. *Behavioral Ecology*, 31:1379–1390, 2020.

M. Vojnovic, S.-Y. Yun, and K. Zhou. Accelerated MM algorithms for ranking scores inference from comparison data. Preprint arxiv:1901.00150, 2019.

J. T. Whelan. Prior distributions for the Bradley-Terry model of paired comparisons. Preprint arxiv:1712.05311, 2017.

R. Yurko, S. Ventura, and M. Horowitz. nflWAR: A reproducible method for offensive player evaluation in football. *Journal of Quantitative Analysis in Sports*, 15:163–183, 2019.

E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29:436–460, 1929.