Robust Active Ranking from Sparse Noisy Comparisons

Lucas Maystre  
EPFL  
lucas.maystre@epfl.ch

Matthias Grossglauser  
EPFL  
matthias.grossglauser@epfl.ch

February 20, 2015

Abstract

From sporting events to sociological surveys, ranking from pairwise comparisons is a tool of choice for many applications. When certain pairs of items are difficult to compare, outcomes can be noisy, and it is necessary to develop robust strategies. In this work, we show how a simple active sampling scheme that uses a standard black box sorting algorithm enables the efficient recovery of the ranking, achieving low error with sparse samples. Both in theory and practice, this active strategy performs systematically better than selecting comparisons at random. As a detour, we show a link between Rank Centrality, a recently proposed algorithm for rank aggregation, and the ML estimator for the Bradley–Terry model. This enables us to develop a new, provably convergent iterative algorithm for computing the ML estimate.

1 Introduction

We consider the problem of recovering a ranking over a set of \( n \) items from a collection of \( m \) pairwise noisy comparisons. Applications are broad and diverse, ranging from ranking chess players based on game outcomes [Elo, 1978] to learning user preferences from individual choices in a recommender system [Wang et al., 2014]. Furthermore, pairwise comparisons have been advocated by psychologists and sociologists as a way of eliciting information for almost a century [Thurstone, 1927; Salganik and Levy, 2014]: Arguably, comparisons are one of the simplest and most natural ways for humans to give feedback. As an example, GIFGIF\footnote{See: \url{http://gifgif.media.mit.edu}} is a large-scale, crowdsourced attempt at understanding how emotions are represented.
in animated GIF images. Visitors of the site are presented with a prompt that displays two images, together with a question such as, “Which better expresses *happiness*?” Aggregating millions of answers enables us to rank the images in a compelling way.

Clearly, any useful method should be able to cope with inconsistent or noisy data. A popular probabilistic model of comparison outcomes is the Bradley–Terry (BT) model [Bradley and Terry, 1952], which we shortly describe in Section 1.1. The BT model posits a notion of distance, or similarity, between items; comparisons are more noisy for similar items than for distant items. As a consequence, the amount of information we can learn from a comparison depends on previous comparisons. For example, if we already know that two items are very distant, comparing them might be almost pointless. This suggests that active (or adaptive) strategies, which select items for comparison based on previous comparisons, might perform much better than agnostic strategies.

In this paper, we confirm this intuition by exploring strategies for active ranking from noisy comparison data. Specifically, we show that sorting algorithms (such as Quicksort), which recover the exact ranking in the noiseless case, provide a basis for efficient active approximate ranking in the noisy case. Running a sorting algorithm over a set of items induces a set of comparisons over these items. We propose a ranking method which runs a sorting algorithm repeatedly over the $n$ items until the budget $m$ is exhausted; the output of the sorting algorithm is ignored, but we retain the set of item pairs that were compared. We then estimate the final ranking from this set of comparisons and their outcomes.

This result requires a detour: We first develop the maximum-likelihood estimator for the BT model based on a set of noisy comparisons among items. The starting point for this is recent work on Rank Centrality (RC), an algorithm for aggregating pairwise comparisons [Negahban et al., 2012]. The analysis of this algorithm provided first-of-its-kind bounds for the finite sample error under the BT model. Since then, there have been several new results that analyze the statistical convergence of ranking estimators under this model [Hajek et al., 2014] and various generalizations [Rajkumar and Agarwal, 2014].

In Section 2, our first contribution is to show how RC can be interpreted as an approximation of the maximum likelihood (ML) estimator. This enables us (a) to show that the ML estimate enjoys essentially the same error guarantees as the RC estimate, and (b) to give a new, provably convergent iterative algorithm that computes the ML estimate. In Section 3, we return to the problem of ranking induced by the BT model. Previous results in the literature show that if compared pairs are chosen in advance, $\Omega(n^2)$ comparisons are necessary to even *approximately* recover the ranking. In light of this, we investigate what happens when we use a simple active strategy: sorting the items. This leads to our second contribution, which is to show (under a certain distribution over BT model parameters) that (c) we can
approximately recover the ranking over the items using $O(n \log n)$ comparisons, and that (d) the “worst mistake” in the estimated ranking is bounded by $O(\log^2 n)$ with high probability. Finally, in Section 4 we test our theoretical findings and show experimentally that the speed of convergence is increased when we obtain comparison outcomes by sorting the items.

**Notations.** We define some of the notations used throughout the paper. The $n$ items are denoted without loss of generality by their true rank in $[n] = \{1, \ldots, n\}$, i.e., $i < j$ means that $i$ is ranked higher than $j$. When $i$ is preferred to $j$ as a result of a pairwise comparison, we denote the observation by $i \prec j$. Let $E$ be the set containing all pairs of items $(i, j), i < j$ that have been compared at least once. A given pair might be compared more than once; let $k_{ij} = k_{ji}$ be the number of times items $i$ and $j$ were compared, out of which $i$ was preferred $A_{ij} = A_{ji}$ times and $j$ was preferred $A_{ij}$ times. When $(i, j) \in E$ we define $a_{ij} = A_{ij}/k_{ij}$ as the ratio of wins of $j$ over $i$. When $(i, j) \notin E$ we let $a_{ij} = a_{ji} = 0$. Finally, an event $B$ is said to occur with high probability if $P(B) \geq 1 - c/n^\alpha$ for $c, \alpha$ fixed.

1.1 Bradley–Terry Model

Informally, the Bradley–Terry (BT) model [Bradley and Terry, 1952] places every item on the real line. The probability of observing a comparison outcome that is inconsistent with the ranking decreases with the distance between the items. This captures the intuitive notion that some pairs are easy to compare, whereas some are more difficult.

More formally, the model defines the probability of observing a particular comparison outcome. It can be parameterized in two different ways, and we will make use of both. In the first parameterization, each item $i$ is associated to a parameter $w_i \in \mathbb{R} > 0$, which we call weight. The probability that $i$ is preferred over $j$ is then

$$p(i < j) = \frac{w_i}{w_i + w_j}.$$  

We call $b = \max_{i,j} w_i/w_j$ the dynamic range of the model. It is clear that the parameters $w = [w_1, \ldots, w_n]^\top$ are defined up to a multiplicative factor, so without loss of generality we assume $w_i \in [1/b, 1]$. Sometimes, it will be convenient to view the parameters as a probability distribution over the items, in which case they need to sum to one. We will then use the notation $\pi$, with $\pi_i = w_i/\sum_j w_j$. The distinction becomes important when we consider sequences with a growing number of items.

An alternative parameterization the model is given by $\theta = [\theta_1, \ldots, \theta_n]^\top$, with $\theta_i = \log(w_i)$. Then

$$p(i < j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}} = \frac{1}{1 + e^{-(\theta_i - \theta_j)}}.$$
In this formulation the parameters are defined up to an additive constant. This second parameterization enables us to make the link with our informal description: intuitively, “close” items have more uncertain comparison outcomes. Finally, let us note that as the likelihood is concave, maximum likelihood estimation is tractable, and efficient algorithms exist [Zermelo, 1929][Hunter, 2004].

1.2 Related Work

[Negahban et al., [2012] 2014] propose Rank Centrality, a spectral ranking algorithm for aggregating pairwise comparisons. They analyze the case where \( k \) comparisons are obtained under the BT model for a fixed set of pairs \( E \). They give bounds on the mean-square error (MSE) of the resulting parameter estimate, as a function of the structural properties of the comparisons graph \( G([n], E) \). The \( G(n, p) \) random graph turns out to have optimal structural properties with high probability, which justifies selecting comparison pairs at random. In this case, their main result implies that \( \omega(n \log n) \) comparisons are enough to ensure that the MSE is \( o(1) \). In Section 2 we provide further insights into Rank Centrality by relating it to the ML estimator. We show, among other contributions, how RC can be used to compute the ML estimate, and provide a principled way of extending RC to the case where pairs might have a different number of comparisons.

[Hajek et al., [2014]] provide a result similar to that of Rank Centrality for the ML estimator. They also present a lower bound on the minimax error rate. It follows that both RC and ML are minimax-optimal up to logarithmic factors. In Section 2 we recover their result for the ML estimator by analyzing RC and give an independent proof of the error bound on the ML estimate. Furthermore, in Section 3 we observe that a minimax analysis is not insightful when the task is to recover the ranking instead of the model parameters. This is the reason why we develop a novel analysis of ranking estimates, in a way that enables us to provide useful guarantees.

The results mentioned all bound the MSE of the parameter estimate \( w \) or \( \theta \). If there is interest in recovering the ranking induced by a BT model, a low average error on the parameter estimate is good only in so far as parameter values are not too close to each other. When comparisons are chosen at random, [Rajkumar and Agarwal, [2014]] look at the statistical convergence of various ranking algorithms, including RC and ML. Their sample complexity results are impractical, with quartic dependencies in \( n \). However, they show that the ML estimator (under the BT model), unlike RC, can recover rankings from comparisons generated according to a model that only satisfies a weaker, more general low-noise condition. The robustness of the ML estimator—to variations in the model—provides another justification to use it for ranking aggregation.

Finally, several works consider active approaches for ranking when the observed
comparisons are noisy. Using a different noise model, Braverman and Mossel [2008] show that $O(n \log n)$ comparisons are sufficient to find a ranking that is close to the original one. In their model, the noise is independent of the comparison pair. In a way, our developments in Section 3 provide similar guarantees for the BT model. Jamieson and Nowak [2011], Ailon [2012], Wang et al. [2014] also consider active ranking with pairwise data, but they depart from the BT model used in our setting.

2 Maximum Likelihood vs. Rank Centrality

In this section, we find a link between Rank Centrality [Negahban et al., 2012, 2014] and the ML estimator for the BT model. This allows us (a) to show that the MSE on the ML estimate is $o(1)$ as long as we have $\omega(n \log n)$ comparisons over randomly chosen pairs, (b) to present a novel, provably convergent iterative algorithm that computes the ML estimate using RC, and (c) to extend RC for the case of uneven number of comparisons across pairs.

We will use $\cdot^*$ to denote true model parameters, $\bar{\cdot}$ for parameters estimated by or related to RC, and $\hat{\cdot}$ for parameters estimated by or related to the ML estimator.

2.1 Rank Centrality Approximates the ML Estimate

We start by recalling the Rank Centrality algorithm. Let $G([n], E)$ be the comparisons graph and assume that every pair in $E$ has been compared exactly $k$ times. Informally, the RC estimate is defined as the stationary distribution of a random walk on the graph that is biased towards items that are preferred more often. Denote by $d_{\max}$ and $d_{\min}$ the maximum and the minimum degree of a node in $G$, respectively, and let $\epsilon = 1/d_{\max}$. Consider the Markov chain on $G$ with the following transition matrix, which we assume to be irreducible and aperiodic.

$$
\bar{P}_{ij} = \begin{cases} 
\epsilon a_{ij} & \text{if } i \neq j, \\
1 - \epsilon \sum_{l \neq i} a_{il} & \text{if } i = j.
\end{cases}
$$

The factor $\epsilon$ ensures that the matrix is stochastic. The RC estimate $\bar{\pi}$ is defined as the (unique) stationary distribution of the random walk on $\bar{P}$, i.e., satisfying $\bar{\pi} \bar{P} = \bar{\pi}$.

Let $B$ be the adjacency matrix of $G$, and $D$ the diagonal matrix of its nodes’ degrees. A Markov chain with transition matrix $D^{-1}B$ represents an unbiased random walk on $G$. Let $\gamma$ be the spectral gap of this transition matrix; intuitively, the larger the spectral gap is, the faster the convergence towards the stationary distribution is. The main result of the analysis of RC is as follows.
Theorem 1 (Negahban et al., 2012). For $k \geq 4C^2(1 + (b^5\kappa^2/d_{\text{max}}\gamma^2) \log n)$, the error on the RC estimate $\bar{\pi}$ satisfies w.h.p.

$$\frac{\|\bar{\pi} - \pi^*\|}{\|\pi^*\|} \leq C \frac{b^{5/2}\kappa}{\gamma} \sqrt{\frac{\log n}{kd_{\text{max}}}},$$

where $C$ is a constant, $b = \max_{i,j} w_i^*/w_j^*$ and $\kappa = d_{\text{max}}/d_{\text{min}}$.

How to interpret the left-hand side of (1) is not necessarily clear at first sight, especially as parameters $\pi^*_i$ and $\bar{\pi}_i$ contract as $n$ grows to ensure that they continue to form a probability distribution. For this reason, we state the following corollary, which expresses the error in a more interpretable way.

Corollary 1. For $k \geq 4C^2(1 + (b^5\kappa^2/d_{\text{max}}\gamma^2) \log n)$, the error on the RC estimate $\bar{w}$ satisfies w.h.p.

$$\|\bar{w} - w^*\| < C \frac{b^3\kappa}{\gamma} \sqrt{\frac{n \log n}{kd_{\text{max}}}},$$

where $C$ is a constant, $b = \max_{i,j} w_i^*/w_j^*$ and $\kappa = d_{\text{max}}/d_{\text{min}}$.

The proof can be found in Appendix A. Corollary 1 bounds the error of the parameter estimate when the dynamic range stays constant. The error depends on the structure of the comparisons graph through $d_{\text{min}}, d_{\text{max}}$ and $\gamma$. In the best case, $\kappa$ and $\gamma$ are bounded by a constant, in which case $nkd_{\text{max}} = \omega(n \log n)$ comparisons are enough to make sure that the MSE is $o(1)$.

We move on to show how RC relates to the ML estimator. On one hand, the Rank Centrality estimate, as the stationary distribution of a Markov chain, satisfies the global balance equations. Let $N_i$ be the set of neighbors of $i$ in $G$, i.e., the set of items $i$ has been compared with. For all $i$,

$$\sum_{j \in N_i} \bar{\pi}_i \bar{P}_{ij} = \sum_{j \in N_i} \bar{\pi}_j \bar{P}_{ji} \iff \sum_{j \in N_i} (\bar{\pi}_i a_{ij} - \bar{\pi}_j a_{ji}) = 0.$$ 

(2)

On the other hand, the maximum likelihood satisfies the following claim.

Claim. The maximum likelihood estimate $\hat{\pi}$ satisfies, for all $i$,

$$f_i(\hat{\pi}) = \sum_{j \in N_i} \frac{1}{\hat{\pi}_i + \hat{\pi}_j} (\hat{\pi}_i a_{ij} - \hat{\pi}_j a_{ji}) = 0.$$ 

(3)
Proof. The log-likelihood of the observations given weights \( \pi \) can be written as
\[
\log L = \sum_{i \in [n]} \sum_{j \in N_i} a_{ji} (\log(\pi_i) - \log(\pi_i + \pi_j))
\]
\[
\Leftrightarrow\frac{\partial \log L}{\partial \pi_i} = \sum_{j \in N_i} \frac{1}{\pi_i(\pi_i + \pi_j)} (\pi_i a_{ij} - \pi_j a_{ji}).
\]

The ML estimate \( \hat{\pi} \) by definition satisfies \( \nabla \hat{\pi} L = 0 \). The factor \( 1/\hat{\pi}_i \) can be dismissed, and the claim follows.

Consider the task of finding an appropriate linearization of the function \( f \) in (3). We will now present two approaches that lead to the same result. First, the idea of linearizing \( f \) directly could be abandoned because a good point for \( f \) to be developed around cannot be given. In this case, by writing \( f_i(\pi) = \sum_{j \in N_i} f_{ij}(\pi_i, \pi_j) \) with
\[
f_{ij}(\pi_i, \pi_j) = \frac{1}{\pi_i + \pi_j}(\pi_i a_{ij} - \pi_j a_{ji}),
\]
we can linearize each function \( f_{ij} \) separately. A good point around which to develop is \( c_{ij} \cdot [a_{ji}, a_{ij}]^T \), as the ratio \( \pi_i/\pi_j \) is expected to be \( \approx a_{ji}/a_{ij} \) (\( c_{ij} \) is a scaling constant.) Summing up the first-order Taylor approximations of \( \{f_{ij}\}_{j \in N_i} \), we get
\[
f_i(\pi) = \sum_{j \in N_i} \frac{1}{c_{ij}}(\pi_i a_{ij} - \pi_j a_{ji}) + \text{H. O. T.}
\]
Taking \( c_{ij} = 1 \) yields Rank Centrality (2). Alternatively, if we were to linearize around the maximum likelihood estimate, we would find the following first-order Taylor approximation.
\[
f_i(\pi) = \sum_{j \in N_i} \frac{1}{\hat{\pi}_i + \hat{\pi}_j} \left( \frac{\hat{\pi}_j}{\hat{\pi}_i + \hat{\pi}_j} \pi_i - \frac{\hat{\pi}_i}{\hat{\pi}_i + \hat{\pi}_j} \pi_j \right) + \text{H. O. T.}
\]
As \( a_{ij} \) is expected to be close to \( \hat{\pi}_j/(\hat{\pi}_i + \hat{\pi}_j) \), we can plug the data in. As for the coefficient \( 1/(\hat{\pi}_i + \hat{\pi}_j) \), in the absence of additional knowledge, we can just ignore it (or set it to \( \frac{1}{2n} \)). Again we fall back on Rank Centrality (2).

Furthermore, (3) enables us to view the ML estimate as the stationary distribution of a Markov chain with transition matrix \( \hat{P} \), where
\[
\hat{P}_{ij} = \begin{cases} 
\epsilon \frac{a_{ij}}{\hat{\pi}_i + \hat{\pi}_j} & \text{if } i \neq j, \\
1 - \epsilon \sum_{l \neq i} \frac{a_{il}}{\hat{\pi}_i + \hat{\pi}_l} & \text{if } i = j.
\end{cases}
\]
The factor $\epsilon = \hat{\pi}_{\text{min}} / d_{\text{max}}$ ensures that $\hat{P}$ is stochastic. This matrix enables us to analyze the ML estimator by using the methods developed for RC.

**Theorem 2.** For $k \geq 4C^2(1 + (b^4 \kappa^2 / d_{\text{max}} \gamma^2) \log n)$, the error on the ML estimate $\hat{w}$ satisfies w.h.p.

$$\|\hat{w} - w^*\|_2 < C b^4 \sqrt{n \log n / kd_{\text{max}}},$$

where $C$ is a constant, $b = \max_{i,j} w^*_i / w^*_j$ and $\kappa = d_{\text{max}} / d_{\text{min}}$.

**Proof.** Given the Markov chain $\hat{P}$, the proof essentially follows that of Theorem 1 of Negahban et al. [2014]. Let $P^*$ be the ideal Markov chain, when $a_{ij} = \pi^*_j / (\pi^*_i + \pi^*_j)$, i.e., the ratios are noiseless. The key insight is to note that the stationary distribution of $P^*$ is $\pi^*$, the true model parameters. By bounding $\|\hat{P} - P^*\|_2$ and $1 - \lambda_{\text{max}}(P^*)$, we can bound the error on the stationary distribution of $P^*$.

For the former, a straightforward application of the proof in the RC case suffices. For the latter, in the application of the comparison theorem, the lower bound on $\min_{i,j} \pi^*_i P^*_{ij}$ changes by a factor of $1/(2b^4)$. This is due to the additional factor $\hat{\pi}_{\text{min}} / (\hat{\pi}_i + \hat{\pi}_j)$ in the off-diagonal entries of $P^*$.

This independently recovers the result recently shown by Hajek et al. [2014], by using an alternative method. For completeness, we provide a corollary that gives an instance of the bound for random $G(n, d/n)$ graphs.

**Corollary 2.** For $d \geq C' \log n$, $k \geq C'$ and $kd \geq C'b^6 \log n$, the error on the ML estimate $\hat{w}$ satisfies w.h.p.

$$\|\hat{w} - w^*\|_2 < C b^4 \sqrt{n \log n / kd},$$

where $C, C'$ are constants and $b = \max_{i,j} w^*_i / w^*_j$.

Corollary 2 shows that $\omega(n \log n)$ comparisons on randomly chosen pairs are sufficient also for the ML estimator to ensure that the MSE vanishes as $n$ increases. Combining these results with the lower bounds of Hajek et al. [2014] shows that ML and RC are minimax-optimal up to a logarithmic factor.

### 2.2 The Case of General $k_{ij}$

In this section, we make a small aside to show how our newly developed understanding of RC enables us to state a principled way to extend the algorithm to a non-uniform number of comparisons. Let $k_{\text{max}} = \max_{i,j} k_{ij}$, and $\epsilon = (k_{\text{max}} d_{\text{max}})^{-1}$. 
Algorithm 1 Adjusted Rank Centrality

Require: $[n], E, [a_{ij}]$

1: $\pi \leftarrow [1/n, \ldots , 1/n]^T$
2: repeat
3: $Q \leftarrow P(\pi)$ as per (5)
4: $\pi \leftarrow \pi Q$
5: until convergence

The Markov chain
\[
\tilde{P}_{ij} = \begin{cases} 
\epsilon \pi a_{ij} & \text{if } i \neq j, \\
1 - \epsilon \sum_{l \neq i} A_{il} \pi_l & \text{if } i = j
\end{cases}
\]
leads to an estimate $\tilde{\pi}$ that, via the global balance equations, satisfies
\[
\sum_{j \in N_i} (\tilde{\pi}_i A_{ij} - \tilde{\pi}_j A_{ji}) = 0.
\]
This is to be compared to the maximum likelihood estimate that satisfies
\[
\sum_{j \in N_i} \frac{1}{\tilde{\pi}_i + \tilde{\pi}_j} (\tilde{\pi}_i A_{ij} - \tilde{\pi}_j A_{ji}) = 0.
\]
Unfortunately, the analysis of the error does not benefit from the higher number of comparisons available for certain pairs. Bounds in this general case can be obtained by setting $k \doteq \min_{i,j} k_{ij}$ in Theorem 1.

2.3 Adjusted Rank Centrality

Equation (3) and the Markov chain $\tilde{P}$ in (4) suggest an iterative procedure to compute the maximum likelihood estimate using adapted Rank Centrality Markov chains as primitives. Algorithm 1, which we call Adjusted Rank Centrality (ARC), describes such a procedure. It makes use of the following Markov chain, parameterized by $\pi$.
\[
P(\pi)_{ij} = \begin{cases} 
\epsilon \pi \frac{a_{ij}}{\pi_i + \pi_j} & \text{if } i \neq j, \\
1 - \epsilon \pi \sum_{l \neq i} \frac{a_{il}}{\pi_i + \pi_l} & \text{if } i = j
\end{cases}
\]
The factor $\epsilon = \pi_{\min}/d_{\max}$ ensures that the matrix is stochastic.
Theorem 3. Algorithm 1 converges to the maximum likelihood estimate, provided that the Markov chain \( \bar{P} \) is irreducible and aperiodic.

The full proof of Theorem 3 is given in Appendix A. The proof simply shows that \( T(\pi) = \pi P(\pi) \) is a contraction mapping; by (3) the unique fixed point is the ML estimate \( \hat{\pi} \). Note that \( \bar{P} \) is aperiodic as soon as it has a self-transition, and that if \( \bar{P} \) is not irreducible, the ML estimate is ill-defined. Lastly, ARC bears similarities to Hunter’s minorization–maximization (MM) algorithm [Hunter, 2004]. Like MM, it solves the ML optimization problem by iterative linear relaxations. However, MM linearizes part of the likelihood function directly, whereas ARC linearizes the gradient.

3 Ranking by Sorting

In Section 2 we see that RC and the ML estimator are both minimax-optimal up to logarithmic factors, given comparisons for pairs chosen uniformly at random. This might lead us to think that we should not seek further and simply try to gather comparisons on random pairs of items. However, when the task is to recover the ranking induced by a BT model, some considerations change.

First, let us review the assumption of fixed dynamic range \( b \) for growing \( n \). For finding a good BT parameter estimate, a constant \( b \) is good, as it restricts the possibilities to a constant range. This is made clear in Theorem 2, where the error has a dependency in \( b^4 \). The task of learning the induced ranking, however, seems easier for large (or growing) values of \( b \). If two items are in general more “distant” from each other, the outcome of the comparison is less noisy, resulting in a better aggregate order (even though the actual parameters \( w \) or \( \theta \) might be estimated less precisely.) Whether to consider \( b \) as growing or fixed is ultimately a modeling choice that has to be tested on real cases and based on quantitative results.

Second, a minimax analysis is not insightful for learning a ranking. It is easy to construct an instance of BT model where a fraction of the items are arbitrarily close to each other, thus making it arbitrarily hard to find their relative order. For this reason, we consider in this section the Bayes risk. We assume that the parameters \( \theta^* \) are a realization of a Poisson point process with rate \( \lambda \). Equivalently, for \( i = 1, \ldots, n - 1 \),

\[
i.i.d \quad X_i = \theta_i^* - \theta_{i+1}^* \sim \text{Exp}(\lambda).
\]

Hence, \( E(b) = \exp((n - 1)/\lambda) \). Let \( \sigma \) be the ranking induced by a BT estimate \( \theta \), i.e., \( \sigma(i) = \text{rank of item } i \). To measure the quality of this ranking, we look at the average displacement

\[
\Delta(\sigma) = \frac{1}{n} \sum_{i=1}^{n} |\sigma(i) - i|.
\]
This metric is the one considered by Braverman and Mossel [2008]. It is also known as the (average) Spearman’s footrule distance.

In the noiseless case, \( \Omega(n^2) \) random comparisons are necessary to recover the ranking, even if we allow for a constant \( \Delta \) bounded away from zero [Alon et al., 1994]. However, various sorting algorithms find the true ranking with \( O(n \log n) \) comparisons, attaining the information-theoretic lower bound. This raises the question: Can we characterize the quality of a ranking produced by a sorting procedure operating with noisy comparisons? In Section 3.1 we give theoretical guarantees for one such sorting algorithm, Quicksort. In Section 3.2 we compare the theory to simulation results.

### 3.1 Analysis of Quicksort

We consider the case of the randomized Quicksort algorithm [Sedgewick and Wayne, 2011]. Given a set of \( n \) items, Quicksort works by selecting a pivot element \( p \) uniformly at random and dividing the items into two subsets: elements lesser and greater than \( p \). This operation is called partition. The algorithm then recurses on these two subsets. First, we show that Quicksort always terminates and is, in a sense, noise-agnostic.

**Claim.** Quicksort returns a ranking after \( O(n \log n) \) comparisons with high probability, regardless of the outcome of the comparisons.

**Proof.** The claim follows from standard results of the analysis of the algorithm, see for example Sedgewick and Wayne [2011]. Comparisons used by Quicksort can be reinterpreted as those used in the construction of a binary search tree, where elements are inserted in a random order. That the comparisons form a tree shows that no non-transitive comparison is observed, and the ranking can simply be read by a pre-order traversal. \( \square \)

Notice also that if we were to take the comparisons used by a single run of Quicksort and plug them into a ML estimator for BT, the ranking induced by the resulting estimate would necessarily match the ranking output by Quicksort. We will now state two theorems that bound the average and the maximum displacement of a ranking produced by Quicksort.

**Theorem 4.** Assume comparisons are made under a BT model with \( \theta \) as per (6).

Let \( \sigma \) be the ranking output by randomized Quicksort. Then, w.h.p.,

\[
\Delta(\sigma) = O(\lambda^3)
\]

We only give a sketch of the proof here, the full proof is available in Appendix B. In general, the analysis of a sorting algorithm under noisy comparisons is difficult,
because comparisons are strongly interdependent and a single mistake can affect many items. In our case, we are helped by the fact that, in the BT model, the probability of making a mistake decreases exponentially fast with the distance between items. Within a single partition operation, consider the rank difference (relative to \( p \)) of items that end up in the wrong subset. It is highly improbable that an incorrect outcome of the comparison is observed for an item with a high rank difference. This enables us to bound the expectation and variance of the total displacement caused by a single partition operation by \( O(\lambda^3) \) and \( O(\lambda^5) \), respectively. As the total number of partition operations performed by Quicksort is upper bounded by \( n \), the average displacement has mean \( O(\lambda^3) \) and variance \( O(\lambda^5/n) \).

**Theorem 5.** Assume comparisons are made under a BT model with \( \theta \) as per (6). Let \( \sigma \) be the ranking output by randomized Quicksort. Then, w.h.p.,

\[
\max_i |\sigma(i) - i| = O(\lambda \log^2 n)
\]

Again, we sketch the proof, with the full proof available in Appendix B. Consider a partition operation. Using a Chernoff bound argument, with high probability no item with rank distance \( O(\lambda \log n) \) from the pivot will end up in the wrong subset. Combined with the fact that an item goes with high probability through \( O(\log n) \) partition operations, the result follows.

These bounds are rather good news: They mean that a very simple and efficient active strategy, sorting the items, enables us to approximately recover the ranking.

### 3.2 Empirical Validation of Bounds

To empirically test our bounds, we perform some simulations. We generate values for \( \theta \) according to (6), apply Quicksort, and record the ranking. Figure 1 shows the average and maximum displacement for varying values of \( n \) and \( \lambda \). Our bound for \( \Delta(\sigma) \) appears to be tight in \( n \), but seems to be loose in \( \lambda \), as the simulations seem to indicate a linear dependency in this parameter. Our bound for the maximum displacement appears to be tight in \( \lambda \). The dependency in \( n \) seems to be \( O(\log n) \), however, we observe that the variance does not decrease. Therefore, our with-high-probability bound is presumably optimal. The contraction observed on the right plot for small values of \( n \) is due to border effects.

### 4 Experimental Evaluation

In this section, we further expand on the connection between sorting and rank estimation from noisy comparisons. Note that all (non-trivial) sorting algorithms
Figure 1: Average and maximum displacement of ranking with Quicksort, under a BT model following (6). On the left, we fix $n = 10^4$ and show the displacement as a function of $\lambda$. On the right, we fix $\lambda = 20$ and show it for exponentially increasing $n$. Values are averaged over 100 realization and are reported with their standard deviation.

are active, in the sense that each comparison pair is a function of the outcomes of earlier comparisons. For example, in Quicksort, two non-pivot items $i$ and $j$ are compared only if they fall on the same side of all previous pivots $p$. A natural question to ask is whether the set of comparisons induced by such a sorting procedure can lead to a better ranking estimate than a set of random comparison pairs. In this section, we answer this question in the affirmative.

Section 3 shows that under some assumptions, selecting comparison pairs via sorting enables us to recover the ranking up to a constant displacement. In practice, the comparison budget for ranking estimation from noisy data is typically larger than that for a single sort. This suggests the following heuristic strategy: For a budget of $m$ comparisons, run the sorting procedure repeatedly until the budget is depleted (the last call may have to be truncated.) We retain only the set of $m$ comparison pairs and their outcomes, but discard the rankings produced by the sorting procedure. The final ranking estimate is then given by the ML estimate $\hat{\pi}$ over the set of $m$ comparison outcomes.

In this section, we compare the quality of this estimate to one based on comparisons selected uniformly at random. We first present the results on synthetic data, and then use two real-world datasets: one containing comparisons between animated GIFs, and one consisting of sushi preferences.
Figure 2: Three BT models are generated following (6), with $\lambda \in \{2, 10, 50\}$ and $n = 500$, and comparisons are collected following a passive and an active approach. We report the ratio (passive vs. active) of the average displacement of the estimated ranking (mean and standard deviation over 20 realizations.)

4.1 Synthetic Data

In this experiment, we set $n = 500$ and generate models following (6) for values of $\lambda \in \{2, 10, 50\}$. For increasing values of $m$, we collect comparisons outcomes using (a) repeated calls to Quicksort, and (b) random pairs of items. In both cases, the ranking is induced by the ML estimate fitted on the observed comparisons.

In Figure 2, we show the relative improvement of selecting comparisons via sorting over random selection. We can see that the average displacement is always improved by some factor for our range of parameters. For example, with $\lambda = 10$ and for $m = 10^4$, the average displacement when comparisons are selected actively is half that obtained with the same number of random comparisons. As can be expected, the less noisy the comparisons are (small values of $\lambda$), the better the improvement obtained by the active approach is. Furthermore, it seems that even for large $m$, the active approach keeps its relative advantage.

4.2 GIFGIF Dataset

Next, we look at a dataset from GIFGIF, a survey to which we referred in Section 1. GIFGIF aims at explaining the emotions communicated by a collection of over 6000 animated GIF images. Users of the website are shown a prompt with two images and a question, “Which better expresses $x$?” where $x$ is one of 17 emotions. The users can click on either image, or use a third option, neither. To date, over 2.7 million comparison outcomes have been collected. For the purpose of our experiment,
we restrict ourselves to a single emotion, happiness, and we ignore outcomes that resulted in neither. We considered $m = 106,887$ comparison outcomes over $n = 6,121$ items. To the best of our knowledge, this is the largest dataset of human-generated pairwise comparisons analyzed in the context of rank aggregation.

As the data, despite a large number of comparisons, remains sparse (less than 20 comparisons per item on average), we proceed as follows. We fit a BT model by using all the available comparisons and use it as ground truth. We then generate comparison outcomes directly from the BT model, for both active and passive approaches. In this sense, the experiment enables us to compare active and passive sampling by using a model with realistic parameters that do not necessarily follow (6) anymore.

Figure 3 (left) compares the average displacement $\Delta$ of passive and active approaches. The active approach performs systematically better, with an improvement of about 7% after observing 160,000 outcomes. The improvement is noticeable but modest. We notice that item parameters are close to each other on average; fitting a Poisson point process yields $\hat{\lambda} \approx 376$. This is because there is a considerable fraction of items that have their parameters very close to one another. This might mean that a significant fraction of the images do not express any opinion with respect to happiness. Figuring out the exact order of these images is therefore difficult and probably of marginal value.

4.3 Sushi Dataset

For our last experiment, we look at a dataset of Sushi preferences [Kamishima and Akaho, 2009]. In this dataset, 5000 respondents give a strict ordering over 10
different types of sushis. These 10 sushis are chosen among a larger set of \( n = 100 \) items. To suit our purposes, we decompose each partial ranking into pairwise comparisons, resulting in \( m = 225,000 \) comparison outcomes.

This time, the data is dense enough to enable us to use the comparison outcomes directly. When an outcome for pair \((i, j)\) is requested, we sample uniformly at random over the outcomes observed for this pair. In the rare case where no outcome is available, we return \( i \prec j \) with probability \( 1/2 \). This enables us to compare active and passive approaches in a more realistic setting, where the assumptions of the BT model do not necessarily hold anymore.

The ground truth is set to the ranking induced by a BT model fitted with all the available data. Results are shown in Figure 3 (right.) We see that the active approach is able to bring the average displacement down to a required level much more quickly than the passive approach. For example, an average displacement of \( \Delta = 10 \) is reached after 5,015 random comparisons, but only 2,492 if we collect comparisons with Quicksort—reducing the number of comparisons needed by a factor 2.

5 Conclusion

In this work, we have shown that recovering a ranking from pairwise comparisons can benefit from an active approach. Selecting the next pair of items to compare based on previous comparison outcomes yields systematic gains over selecting random pairs. Perhaps surprisingly, a model-agnostic black box, Quicksort, is able to approximately recover the ranking even in the presence of noise. On our way, we broadened the understanding of the Rank Centrality algorithm. This enabled us to develop a novel iterative algorithm that computes the maximum likelihood estimate in the Bradley–Terry model, and to give formal guarantees on the MSE of this estimate.

Future work on active strategies for ranking could investigate the very sparse case, for example when the budget of comparisons \( m \) is linear in the number of items \( n \). In this case, there are too few comparisons for even a single sort to be carried out entirely.

Acknowledgments

We are very grateful to Travis Rich and Kevin Hu for providing us with the GIFGIF dataset. Furthermore, we wish to thank Holly Cogliati-Bauereis and Ksenia Konyushkova for their careful proofreading and comments on the text.
A Analysis of Rank Centrality

First we give the proof Corollary 1, which bounds the error of the RC estimate in a more interpretable way. Then, we prove the convergence of the Adjusted Rank Centrality (ARC) algorithm.

Proof of Corollary 1. We will show that
\[ \| w - w^* \|_2 \leq \sqrt{nb} \frac{\| \pi - \pi^* \|_2}{\| \pi^* \|_2}. \] (7)

First, we bound \( \| \pi^* \|_2 \). Let \( v^{(i)} \) s.t. \( v^{(i)}_i = b/(n - 1 + b) \) and \( v^{(i)}_j = 1/(n - 1 + b) \) for all \( j \neq i \). We have, for \( n \geq b - 1 \),
\[ \| v^{(i)} \|_2 = \sqrt{\frac{n - 1 + b}{(n - 1 + b)^2}} = \sqrt{\frac{n + (b - 1)^2}{n(n + 2b - 2) + (b - 1)^2}} \] (8)
\[ \leq \sqrt{\frac{1}{n} + \frac{b - 1}{n}} = \sqrt{\frac{b}{n}}. \] (9)

Note that because \( b \) is fixed, any feasible \( \pi \) is a convex combination of the \( \{ v^{(i)} \}_{i \in [n]} \), hence by the triangle inequality \( \| \pi^* \|_2 \leq \sqrt{b/n} \). Furthermore, we have
\[ \| \pi - \pi^* \|_2 = \frac{1}{\sum_i w_i^*} \| w - w^* \|_2 \geq \frac{1}{n} \| w - w^* \|_2. \] (10)

The corollary ensues.

The proof of convergence for the Adjusted Rank Centrality algorithm (Theorem 3) essentially analyzes the operator defined by (5).

Proof of Theorem 3. The proof is inspired from [Tresch 2008] and [von Petersdorff 2014]. Let \( T : \text{int}(\Delta^{n-1}) \to \text{int}(\Delta^{n-1}) \) be a mapping of the interior of the simplex onto itself, defined by
\[ T(\pi) = \pi P(\pi). \] (11)

We will use the notation
\[ T^k(\pi) = \underbrace{T \circ T \circ \ldots \circ T}_{k \text{ times}}(\pi), \] (12)
with \( T^0(\pi) = \pi \). The Jacobian matrix of \( T \) is defined as
\[ T'_{ij} = \frac{\partial T_i}{\partial \pi_j} = \begin{cases} \pi_{\min} \frac{\pi_i}{d_{\max} (\pi_i + \pi_j)^2} & \text{if } (i, j) \in E, \\ 1 - \frac{\pi_{\min}}{d_{\max}} \sum_{i \neq j} \frac{\pi_i}{(\pi_i + \pi_j)^2} & \text{if } i = j, \\ 0 & \text{otherwise}. \end{cases} \] (13)
Note that $T$ and $T'$ are stochastic, and have the same non-zero entries as $\bar{P}$. By stochasticity, $\|T\|_1 = \|T'\|_1 = 1$. Now consider the smallest $m$ such that $Q = \bar{P}^m > 0$, i.e., such that $Q_{ij} > 0$ for all $i, j \in [n]$. That such a finite $m$ exists is guaranteed by the fact that $\bar{P}$ is irreducible and aperiodic. By the above, it follows that

\begin{itemize}
  \item $S = T^m$ is such that $S_{ij} > 0$ for all $i, j \in [n]$, and
  \item $S' = (T^m)' = \prod_{i=0}^{n-1} T'(T^i(\pi))$ is such that $S'_{ij} > 0$ for all $i, j \in [n]$ and $\|S'\|_1 = 1$.
\end{itemize}

Let $\epsilon = \min_{i,j} S'_{ij} > 0$, and let $J_n$ be the $n \times n$ all-ones matrix. We can write $S'$ as

$$S'(\pi) = \epsilon J_n + R(\pi),$$

where $\|R(\pi)\|_1 = 1 - n\epsilon = c$. Note that by construction $0 < c < 1$. Now pick any $x, y \in \text{int}(\Delta^{n-1})$, and let $\tilde{S} = S(x + u(x - y))$. Then $\tilde{S}'(u) = S'(x + u(y - x))(y - x)$, and

$$S(y) - S(x) = \tilde{S}(1) - \tilde{S}(0) = \int_0^1 \tilde{S}'(u)du = \int_0^1 S'(x + u(y - x))(y - x)du$$

As $S'$ is continuous, we have

$$\|S(y) - S(x)\|_1 \leq \int_0^1 \|S'(x + u(y - x))(y - x)\|_1 du$$

where $\|R(\pi)\|_1 = 1 - n\epsilon = c$. Now pick any $x, y \in \text{int}(\Delta^{n-1})$, and let $\tilde{S} = S(x + u(x - y))$. Then $\tilde{S}'(u) = S'(x + u(y - x))(y - x)$, and

$$S(y) - S(x) = \tilde{S}(1) - \tilde{S}(0) = \int_0^1 \tilde{S}'(u)du = \int_0^1 S'(x + u(y - x))(y - x)du$$

As $S'$ is continuous, we have

$$\|S(y) - S(x)\|_1 \leq \int_0^1 \|S'(x + u(y - x))(y - x)\|_1 du$$

where $\|R(\pi)\|_1 = 1 - n\epsilon = c$. Now pick any $x, y \in \text{int}(\Delta^{n-1})$. Then $\tilde{S}'(u) = S'(x + u(y - x))(y - x)$, and

$$S(y) - S(x) = \tilde{S}(1) - \tilde{S}(0) = \int_0^1 \tilde{S}'(u)du = \int_0^1 S'(x + u(y - x))(y - x)du$$

As $S'$ is continuous, we have

$$\|S(y) - S(x)\|_1 \leq \int_0^1 \|S'(x + u(y - x))(y - x)\|_1 du$$

where $\|R(\pi)\|_1 = 1 - n\epsilon = c$. Now pick any $x, y \in \text{int}(\Delta^{n-1})$. Then $\tilde{S}'(u) = S'(x + u(y - x))(y - x)$, and

$$S(y) - S(x) = \tilde{S}(1) - \tilde{S}(0) = \int_0^1 \tilde{S}'(u)du = \int_0^1 S'(x + u(y - x))(y - x)du$$

This shows that $S$ is a contraction in $\text{int}(\Delta^{n-1})$. Therefore, the Banach fixed-point theorem applies, and the unique fixed-point $\hat{\pi}$ can be found by iterative applications of $S$. Furthermore, $\hat{\pi}$ is also a fixed point of $T$ since

$$T(\hat{\pi}) = T(\lim_{k \to \infty} S^k(\hat{\pi})) = \lim_{k \to \infty} T(S^k(\hat{\pi})) = \lim_{k \to \infty} S^k(T(\hat{\pi})) = \hat{\pi},$$

and this is the only fixed point because any fixed point of $T$ is a fixed point of $S$. Finally, let $x \in \text{int}(\Delta^{n-1})$. The vectors $x, T(x), T^2(x), \ldots$
occur in one of the sequences \( (S^k(T^r(x)))_{k \in \mathbb{Z}_{\geq 0}}, r \in \{0, \ldots, m-1\} \). All sequences converge to \( \hat{\pi} \), and therefore

\[
\lim_{k \to \infty} T^k(x) = \hat{\pi}.
\] (22)

\[\square\]

B Analysis of Quicksort

Algorithm 2 formalizes the Quicksort algorithm (lines 4–13 are referred to as the partition operation.) We consider the case where the outcomes of comparisons follow a Bradley–Terry (BT) model. Comparison outcomes are probabilistic, and might be inconsistent with the true ranking of the items.

We recall that w.l.o.g. the \( n \) items are assumed to be labeled in decreasing rank order. Let \( \theta_n < \ldots < \theta_1 \) be the parameters of the \( n \) items, distributed according to a Poisson point process of rate \( \lambda \). Equivalently, for \( i \in \{1, \ldots, n-1\} \) the random variables \( X_i = \theta_{i+1} - \theta_i \) are i.i.d \( \sim \text{Exp}(\lambda) \). We denote by \( d(i, j) = |i - j| \) the ranking distance between \( i \) and \( j \), and by \( D(i, j) = |\theta_i - \theta_j| \) the distance between their parameters. If \( i < j \), then the probability of observing the outcome \( j < i \), inconsistent with the true ranking, is bounded by

\[
P(j < i) = \frac{1}{1 + e^{D(i,j)}} < e^{-D(i,j)}.
\] (23)

Theorem 4 bounds the average displacement \( \Delta(\sigma) \), and Theorem 5 bounds the maximum displacement \( \max_i |\sigma(i) - i| \).

B.1 Proof of Theorem 4

First we will formalize the fact that in expectation, the probability of observing a wrong outcome in the comparison between \( i \) and \( j \) is exponentially decreasing, not only \( D(i, j) \), but also in the ranking distance \( d(i, j) \).

**Lemma 1.** Let \( i < j \) be two items such that \( d(i, j) = k \). Then, the probability of observing the wrong outcome \( j < i \), in expectation over \( \theta \), is bounded by

\[
P(j < i) \leq \left( \frac{\lambda}{\lambda+1} \right)^k.
\]
Algorithm 2 Quicksort

Require: set $S$ of size $n$

1: if $n < 2$ then
2: return list($S$)
3: end if
4: $L$ ← $\emptyset$
5: $R$ ← $\emptyset$
6: $p$ ← random element in $S$
7: for $i \in S \setminus p$ do
8: if comparing $i$ and $p$ yields $i \prec p$ then
9: $L$ ← $L$ $\cup \{i\}$
10: else
11: $R$ ← $R$ $\cup \{i\}$
12: end if
13: end for
14: return Quicksort($L$) $\cdot p$ $\cdot$ Quicksort($R$)

Proof. The distance $D(i, j) = \theta_i - \theta_j = \sum_{i=i}^{j-1} X_i$ is $\sim$ Erlang($k, \lambda$). Therefore, using Eq. 23,

$$P(j \prec i) < E(e^{-D(i,j)}) = \int_0^\infty \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!} e^{-t} dt$$

$$= \frac{\lambda^k}{(\lambda + 1)^k} \int_0^\infty \frac{(\lambda + 1)^k t^{k-1} e^{-(\lambda+1)t}}{(k-1)!} dt$$

$$= \left(\frac{\lambda}{\lambda + 1}\right)^k$$

Consider a partition operation at an arbitrary level of the recursion, where items in $S \setminus \{p\}$ are compared against $p$. Let $\bar{k}$ be such that no item $i$ compared to $p$ with $d(p, i) > \bar{k}$ is misclassified. In other words, $\bar{k}$ is the maximum rank distance of a misclassified item. There are at most $2\bar{k} + 1$ items that are displaced (counting $p$), and each of them is displaced by at most $\bar{k}$ if $L$ and $R$ are subsequently correctly sorted. In view of this, we define the random variable

$$\alpha = (2\bar{k} + 1)\bar{k},$$

which represents an upper bound to the local contribution to the total displacement caused by a single partition operation. We can bound its expected value and variance by a constant, as is made precise by the next lemma.
Lemma 2. In a partition operation, let \( \bar{k} \) be such that no item \( i \) compared to \( p \) with \( d(p, i) > \bar{k} \) is misclassified. Let \( \alpha = (2\bar{k} + 1)\bar{k} \). Then,

\[
E(\alpha) < \lambda(4\lambda^2 + 7\lambda + 3),
\]

\[
\text{Var}(\alpha) < C\lambda(\lambda^4 + 1)
\]

for some constant \( C \).

Proof. We will use Lemma 1 to bound the probability that an item with a given rank distance is misclassified. First we observe that

\[
P(\bar{k} = k) < P(\text{item at distance } k \text{ is misclassified}). \tag{28}
\]

Therefore,

\[
E((2\bar{k} + 1)\bar{k}) < \sum_{k=0}^{\infty} (2k + 1)k \cdot P(\text{item at distance } k \text{ is misclassified}) \tag{29}
\]

\[
< \sum_{k=0}^{\infty} k(2k + 1) \left( \frac{\lambda}{\lambda + 1} \right)^k = \lambda(4\lambda^2 + 7\lambda + 3). \tag{30}
\]

We can use a similar argument for the variance.

\[
\text{Var}((2\bar{k} + 1)\bar{k}) \leq E((2\bar{k} + 1)^2\bar{k}^2) \tag{31}
\]

\[
< \sum_{k=0}^{\infty} k^2(2k + 1)^2 \left( \frac{\lambda}{\lambda + 1} \right)^k \tag{32}
\]

\[
= \lambda(96\lambda^4 + 264\lambda^3 + 250\lambda^2 + 91\lambda + 9) \tag{33}
\]

We can now proceed to prove the main theorem that bounds the total expected displacement of the output of the Quicksort algorithm.

Proof of Theorem. As each item can be a pivot at most once, there are \( k \leq n \) such partition operations. Denote the upper bound on the displacement caused by each partition operation by \( \{\alpha_1, \ldots, \alpha_k\} \). Then,

\[
E(\Delta(\sigma)) \leq E\left( \frac{1}{n} \sum_{t=1}^{k} \alpha_t \right) = \frac{k}{n} E(\alpha) \leq E(\alpha) < C_1\lambda^3 + C_2, \tag{34}
\]

\[
\text{Var}(\Delta(\sigma)) \leq \frac{1}{n^2} \sum_{t=1}^{k} \text{Var}(\alpha_t) \leq \frac{1}{n} \text{Var}(\alpha) < \frac{C_3\lambda^5 + C_4}{n}, \tag{35}
\]

21
for some constants $C_1$, $C_2$, $C_3$ and $C_4$. By Chebyshev’s inequality,

$$
P(\Delta(\sigma) \leq C_1 \lambda^3 + C_2 + \sqrt{C_3 \lambda^5} + C_4) \geq 1 - \frac{1}{n}.
$$

(36)

Finally, let us remark that it is well known that randomized Quicksort needs \( \approx 2n \log n \) comparisons on average [Sedgewick and Wayne 2011].

**B.2 Proof of Theorem 5**

The first lemma states that w.h.p., the maximum rank difference of an item getting displaced in any partition operation is $O(\lambda \log n)$. For convenience, we assume $\lambda \geq 1$. The same result holds with different constants for smaller values of $\lambda$.

**Lemma 3.** The maximum displacement in any single partition operation is $\leq 3e\lambda \log n$, w.h.p.

**Proof.** Consider a particular partition operation, with pivot $p$. First, we will give a bound on $D(p, i)$ given that $d(p, i) = k$. We recall that $D(p, i) = \sum_{i=1}^{k} X_i$, where $X_i$ is i.i.d. $\sim \text{Exp}(\lambda)$. Using a Chernoff bound yields

$$
P(D(p, i) \leq (1 - \delta)k/\lambda) \leq \left(\frac{1 - \delta}{\lambda} \right)^k. \tag{37}
$$

Let $k = \lceil 3e\lambda \log n \rceil$. By using $\delta = 1 - 1/e$, we get

$$
P(D(p, i) \leq 3 \log n) \leq \frac{1}{n^{2\lambda}} \leq \frac{1}{n^2}, \tag{38}
$$

for $\lambda \geq 1$. Hence with probability $\geq 1 - \frac{2}{n^2}$, elements $x_L$ and $x_R$ at ranking distance of $\lceil 3e\lambda \log n \rceil$ on the left and on the right of $p$ are such that $D(p, x_L) \geq 3 \log n$ and $D(p, x_R) \geq 3 \log n$, respectively. Then,

$$
P(\text{at least one item } i \text{ s.t. } d(p, i) > 3e \log n \text{ is misclassified}) \leq ne^{3\log n} = \frac{1}{n^2}. \tag{39}
$$

In total, there are at most $n$ such partition operations, and therefore the claim follows with probability $\geq 1 - \frac{2}{n}$. \qed

The next lemma states that with high probability, the Quicksort call tree has depth $O(\log n)$. This is a classical result in the analysis of randomized Quicksort.

**Lemma 4.** The maximum call depth of Quicksort is $\leq 24 \log n$ w.h.p.
Proof. At each recursion step, we select a pivot at random in the set of items $S$. The probability that the rank of this item lies in the middle 50% of $S$ is $1/2$, and if it does, the resulting sets $L$ and $R$ see their size decrease by at least $\frac{4}{3}$. Therefore, in at most $\log_{4/3}(n) \leq 4 \log n$ such partitions we get to a subset of size one and match the stopping condition. Even though we do not pick the pivot in the middle 50% every time, it is unlikely that more than $c \cdot 4 \log n$ recursions are needed (for some small $c$) to pick the pivot in the middle range at least $4 \log n$ times. We can make this formal with a Chernoff bound. Let $Z_l$ be i.i.d. $\sim$ Bern$(1/2)$. Then,

$$
P \left( \sum_{l=1}^{k} Z_l \leq (1 - \delta)k/2 \right) \leq e^{-\delta^2 k}. \quad (40)
$$

Setting $\delta = 2/3$ and $k = 24 \log n$, it follows that

$$
P \left( \sum_{l=1}^{24 \log n} Z_l \leq 4 \log n \right) \leq \frac{1}{n^2}. \quad (41)
$$

This means that the depth of a leaf in the recursion tree is at most $24 \log n$ with probability at least $1 - \frac{1}{n^2}$, and therefore that the maximum depth is at most $24 \log n$ with probability at least $1 - \frac{1}{n^2}$. \hfill \Box 

Proof of Theorem 5. Lemma 4 states that an item goes through at most $24 \log n$ partition operations with probability $\geq 1 - \frac{1}{n^2}$. Lemma 3 states that the displacement due to a single partition operation never exceeds $3e \lambda \log n$ with probability $\geq 1 - \frac{3}{n}$. It follows that the total displacement of any item never exceeds $72e \lambda \log^2 n$ with probability $\geq 1 - \frac{4}{n}$. \hfill \Box 

References

N. Ailon. An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity. *Journal of Machine Learning Research*, 13:137–164, 2012.

N. Alon, B. Bollobás, G. Brightwell, and S. Janson. Linear Extensions of a Random Partial Order. *The Annals of Applied Probability*, 4(1):108–123, 1994.

R. A. Bradley and M. E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

M. Braverman and E. Mossel. Noisy sorting without resampling. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA ’08)*, San Francisco, CA, 2008.
A. Elo. *The Rating Of Chess Players, Past & Present*. Arco, 1978.

B. Hajek, S. Oh, and J. Xu. Minimax-optimal Inference from Partial Rankings. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Montreal, QC, Canada, 2014.

D. R. Hunter. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

K. Jamieson and R. Nowak. Active Ranking using Pairwise Comparisons. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, Granada, Spain, 2011.

T. Kamishima and S. Akaho. Efficient Clustering for Orders. In *Mining Complex Data*, pages 261–279. Springer, 2009.

S. Negahban, S. Oh, and D. Shah. Iterative Ranking from Pair-wise Comparisons. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Lake Tahoe, CA, 2012.

S. Negahban, S. Oh, and D. Shah. Rank Centrality: Ranking from Pair-wise Comparisons. preprint, arXiv:1209.1688v2 [cs.LG], Jan. 2014.

A. Rajkumar and S. Agarwal. A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2014)*, Beijing, China, 2014.

M. Salganik and K. Levy. Wiki surveys: Open and quantifiable social data collection. preprint, arXiv:1209.1688v2 [cs.LG], Jan. 2014.

R. Sedgewick and K. Wayne. *Algorithms*. Addison-Wesley, 4th edition, 2011.

L. Thurstone. The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, 21(4):384–400, 1927.

A. Tresch. Convergence Proof for discrete ergodic Markov Chains. Course notes, http://www.treschgroup.de/MachineLearning.html, Jan. 2008.

T. von Petersdorff. Fixed Point Iteration and Contraction Mapping Theorem. Course notes, http://terpconnect.umd.edu/~petersd/6667, 2014.
J. Wang, N. Srebro, and J. Evans. Active Collaborative Permutation Learning. In Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’14), New York, NY, 2014.

E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. Mathematische Zeitschrift, 29(1):436–460, 1929.