Accelerated Spectral Ranking

Arpit Agarwal [1] **Prathamesh Patil** [1] **Shivani Agarwal** [1]

Abstract

The problem of rank aggregation from pairwise and multiway comparisons has a wide range of implications, ranging from recommendation systems to sports rankings to social choice. Some of the most popular algorithms for this problem come from the class of spectral ranking algorithms; these include the rank centrality algorithm for pairwise comparisons, which returns consistent estimates under the Bradley-Terry-Luce (BTL) model for pairwise comparisons (Negahban et al., 2017), and its generalization, the Luce spectral ranking algorithm, which returns consistent estimates under the more general multinomial logit (MNL) model for multiway comparisons (Maystre & Grossglauser, 2015). In this paper, we design a provably faster spectral ranking algorithm, which we call accelerated spectral ranking (ASR), that is also consistent under the MNL/BTL models.

Our accelerated algorithm is achieved by designing a random walk that has a faster mixing time than the random walks associated with previous algorithms. In addition to a faster algorithm, our results yield improved sample complexity bounds for recovery of the MNL/BTL parameters: to the best of our knowledge, we give the first general sample complexity bounds for recovering the parameters of the MNL model from multiway comparisons under any (connected) comparison graph (and improve significantly over previous bounds for the BTL model for pairwise comparisons). We also give a message-passing interpretation of our algorithm, which suggests a decentralized distributed implementation. Our experiments on several real world and synthetic datasets confirm that our new ASR algorithm is indeed orders of magnitude faster than existing algorithms.

[1]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA. Correspondence to: Arpit Agarwal <aarpit@seas.upenn.edu>, Shivani Agarwal <ashivani@seas.upenn.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

1. Introduction

The problem of rank aggregation from pairwise or multiway comparisons is a fundamental one in machine learning with applications in recommendation systems, sports, social choice etc. In this problem, given pairwise or multiway comparisons among n items, the goal is to learn a score for each item. These scores can further be used to produce a ranking over these items. For example, in recommendation systems, the goal might be to learn a ranking over items by observing the choices that users make when presented with different subsets of these items; in sports, the goal might be to rank teams/individuals at the end of a tournament based on pairwise or multiway games between these individuals/teams; in social choice, the goal might be to aggregate the choices of individuals when presented with different alternatives such as candidates in an election.

In the case of pairwise comparisons, a popular model is the Bradley-Terry-Luce (BTL) model (Bradley & Terry, 1952; Luce, 1959) which posits that given a set of n items, there is a positive weight w_i associated with each item i, and the probability that i is preferred over j in a pairwise comparison between i and j is $\frac{w_i}{w_i+w_j}$. The BTL model is a special case of the multinomial logit (MNL)/Plackett-Luce model (Plackett, 1975; McFadden, 1974) which is defined for more general multiway comparisons. Under the MNL model, the probability that an item i is preferred amongst all items in a set S is $\frac{w_i}{\sum_{j \in S} w_j}$.

Rank aggregation under pairwise comparisons has been an active area of research, and several algorithms have been proposed that are consistent under the BTL model (Negahban et al., 2017; Rajkumar & Agarwal, 2014; Hunter, 2004; Chen & Suh, 2015; Jang et al., 2016; Guiver & Snelson, 2009; Soufiani et al., 2013). The case of multiway comparisons has also received some attention recently (Maystre & Grossglauser, 2015; Jang et al., 2017; Chen et al., 2017). Two popular algorithms are the rank centrality (RC) algorithm (Negahban et al., 2017) for the case of pairwise comparisons, and its generalization to the case of multiway comparisons, called the Luce spectral ranking (LSR) algorithm (Maystre & Grossglauser, 2015). The key idea behind these algorithms is to construct a *random walk* (equivalently a Markov chain) over the *comparison graph* on n items, where there is an edge between two items if they are com-

pared in a pairwise or multiway comparison. This random walk is constructed such that its stationary distribution corresponds to the weights of the MNL/BTL model.

Given the widespread application of these algorithms, understanding their computational aspects is of paramount importance. For random walk based algorithms this amounts to analyzing the mixing/convergence time of their random walks to stationarity. In the case of rank centrality and Luce spectral ranking, ensuring that the stationary distribution of the random walk corresponds to the weights of the underlying model forces their construction to have self loops with large mass. These self loops can lead to a large mixing time of $\Omega\big(\xi^{-1} d_{\max}\big)$, where $d_{\max}$ is the maximum number of unique comparisons that any item participates in; and ξ is the spectral gap of the graph Laplacian. In practical settings $d_{\max}$ can be very large, for example when the graph follows a power-law distribution, and can even be $\Omega(n)$ if one item is compared to a large fraction of the items.

In this paper we show that it is possible to construct a faster mixing random walk whose mixing time is $O\big(\xi^{-1}\big)$. We are able to construct this random walk by relaxing the condition that its stationary distribution should exactly correspond to the weights of the MNL model, and instead imposing a weaker condition that the weights can be recovered through a linear transform of the stationary distribution. We call the resulting algorithm accelerated spectral ranking (ASR).

In addition to computational advantages, the faster mixing property of our random walk also comes with statistical advantages, as it is well understood that faster mixing Markov chains lend themselves to tighter perturbation error bounds (Mitrophanov, 2005). We are able to establish a sample complexity bound of $O\big(\xi^{-2}\, n \,\mathrm{poly}(\log n)\big)$, in terms of the *total variation* distance, for recovering the true weights under the MNL (and BTL) model for almost any comparison graph of practical interest. To our knowledge, these are the first sample complexity bounds for the general case of multiway comparisons under the MNL model. Negahban et al. (2017) show similar results in terms of L_2 error for the special case of BTL model. However, their bounds have an additional dependence on $d_{\max}$, due to the large mixing time of their random walk.

We also show that our algorithm can be viewed as a message passing algorithm. This connection provides a very attractive property to our algorithm – it can be implemented in a distributed manner with decentralized communication and comparison data being stored in different machines.

We finally conduct several experiments on synthetic and real world datasets to compare the convergence time of our algorithm with the previous algorithms. These experiments confirm the behavior predicted by our theoretical analysis of mixing times– the convergence of our algorithm is in fact orders of magnitude faster than existing algorithms.

1.1. Our Contributions

We summarize our contributions as follows:

1. **Faster Algorithm:** We present an algorithm for aggregating pairwise comparisons under the BTL model, and more general multiway comparisons under the MNL model, that is provably faster than the previous algorithms of Negahban et al. (2017); Maystre & Grossglauser (2015). We also give experimental evidence supporting this fact.

2. **New and Improved Error Bounds:** We present the first error bounds for parameter recovery by spectral ranking algorithms under the general MNL model for any general (connected) comparison graph. These bounds improve upon the existing bounds of Negahban et al. (2017) for the special case of the BTL model.

3. **Message Passing Interpretation:** We provide an interpretation of our algorithm as a message passing/belief propagation algorithm. This connection can be used to design a decentralized distributed algorithm, which can work with distributed data storage.

1.2. Organization

In Section 2 we describe the problem formally. In Section 3 we present our algorithm for rank aggregation under the MNL/BTL model. In Section 4 we analyze the mixing time of our random walk, showing that our random walk converges much faster than existing approaches. In Section 5 we give bounds on sample complexity for recovery of MNL parameters with respect to the total variation distance. In Section 6 we give a message passing view of our algorithm. In Section 7 we provide experimental results on synthetic and real world datasets.

2. Problem Setting and Preliminaries

We consider a setting where there are n items, and one observes outcomes of noisy pairwise or multiway comparisons between these items. We will assume that the outcome of these comparisons is generated according to the multinomial logit (MNL) model, which posits that each item $i \in [n]$ is associated with a (unknown) weight/score $w_i > 0$, and the probability that item i wins a comparison is proportional to its weight w_i. More formally, when there is a (multiway) comparison between items of a set $S \subseteq [n]$, for $i \in S$, we have

$$p_{i|S} := \Pr(i \text{ is the most preferred item in } S) = \frac{w_i}{\sum_{j \in S} w_j}\,.$$

This model is also referred to as the Plackett-Luce model, and it reduces to the Bradley-Terry-Luce (BTL) model in the special case of pairwise comparisons, i.e. $|S| = 2$. Let

$\mathbf{w} \in \mathbb{R}^n_+$ be the vector of weights, i.e. $\mathbf{w} = (w_1, \cdots, w_n)^\top$. Note that this model is invariant to any scaling of $\mathbf{w}$, so for uniqueness we will assume that $\sum_{i=1}^n w_i = 1$, i.e. $\mathbf{w} \in \Delta_n$ where Δ_n is the n-dimensional probability simplex.

The comparison data is of the following form: there are d different *comparison sets* $S_1, \cdots, S_d \subseteq [n]$, with $|S_a| = m$ for all $a \in [d]$ and some constant $m < n$. For each set S_a, for $a \in [d]$, one observes the outcomes of L independent m-way comparisons between items in S_a, drawn according to the MNL model. The assumptions that each comparison set is of the same size m, and each set is compared an equal L number of times, are only for simplicity of exposition, and we give a generalization in the supplementary material. We will denote by y_a^l the winner of the l-th comparison amongst items of S_a, for $l \in [L]$ and $a \in [d]$.

Given comparison data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^d$, where $\mathbf{y}_a = (y_a^1, \cdots, y_a^L)$, the problem is to find a weight vector $\hat{\mathbf{w}} \in \Delta_n$, which is close to the true weight vector $\mathbf{w}$ under some notion of error/distance. More formally, the problem is to find $\hat{\mathbf{w}} \in \Delta_n$, such that $\|\hat{\mathbf{w}} - \mathbf{w}\|$ can be bounded in terms of the parameters n, L, and m, for some norm $\|\cdot\|$. We will give results in terms of the total variation distance, which for two vectors $\mathbf{u}, \hat{\mathbf{u}} \in \Delta_n$ is defined as

$$\|\mathbf{u} - \hat{\mathbf{u}}\|_{\text{TV}} = \frac{1}{2}\|\mathbf{u} - \hat{\mathbf{u}}\|_1 = \frac{1}{2}\sum_{i \in [n]} |u_i - \hat{u}_i| \,.$$

In the following sections, we will present an algorithm for recovering an estimate $\hat{\mathbf{w}}$ of $\mathbf{w}$, and give bounds on the error $\|\hat{\mathbf{w}} - \mathbf{w}\|_{\text{TV}}$ in terms of the problem parameters under natural assumptions on the comparison data.

3. Accelerated Spectral Ranking Algorithm

In this section, we will describe our algorithm, which we term as accelerated spectral ranking (ASR). Our algorithm is based on the idea of constructing a *random walk*[1] on the *comparison graph* with n vertices, which has an edge between nodes i and j if items i and j are compared in any m-way comparison. The key idea is to construct the random walk such that the probability of transition from node i to node j is proportional to w_j. If w_j is larger than w_i, then with other quantities being equal, one would expect the random walk to spend more time in node j than node i in its steady state distribution. Hence, if we can calculate the stationary distribution of this random walk, it might give us a way to estimate the weight vector $\mathbf{w}$. Moreover, for computational efficiency, we would also want this random walk to have a fast *mixing time*, i.e. it should rapidly converge to its stationary distribution.

The rank centrality (RC) algorithm (Negahban et al., 2017) for the BTL model, and its generalization the Luce spectral ranking (LSR) algorithm (Maystre & Grossglauser, 2015) for the MNL model, are based on a similar idea of constructing a random walk over the comparison graph. These algorithms construct a random walk whose stationary distribution, in expectation, is exactly $\mathbf{w}$. However, this construction forces their Markov chain to have self loops with large mass, slowing down the convergence rate.

In this section we will show that it is possible to design a *significantly* faster mixing random walk that belongs to a different class of random walks over the comparison graph. More precisely, the random walk that we construct is such that it is possible to recover the weight vector $\mathbf{w}$ from its stationary distribution using a fixed linear transformation, while for RC and LSR, the stationary distribution is exactly $\mathbf{w}$. Our theoretical analysis in Section 5 as well as experiments on synthetic and real world datasets in Section 7 will show that this difference can lead to vastly improved results.

Given comparison data $\mathbf{Y}$, let us denote by $G_c([n], E)$ the undirected graph on n vertices, with an edge $(i, j) \in E$ for any i, j that are a part of an m-way comparison. More formally, $(i, j) \in E$ if there exists an index $a \in [d]$ such that $i, j \in S_a$. We will call G_c the *comparison* graph, and throughout this paper, we will assume that $\mathbf{Y}$ is such that G_c is connected. We will denote by d_i the number of unique m-way comparisons of which $i \in [n]$ was a part, i.e. $d_i = \sum_{a \in [d]} \mathbf{1}[i \in S_a]$. Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix, with D_{ii} being equal to d_i, $\forall i \in [n]$. Also, let $d_{\max} := \max_i d_i$ and $d_{\min} := \min_i d_i$.

Suppose for each $a \in [d]$ and $j \in S_a$, one had access to the true probability $p_{j|S_a}$ of j being the most preferred item in S_a. Then one could define a random walk on G_c with transition probability from node $i \in [n]$ to $j \in [n]$ given by

$$P_{ij} := \frac{1}{d_i} \sum_{a \in [d]: i, j \in S_a} p_{j|S_a} = \frac{1}{d_i} \sum_{a \in [d]: i, j \in S_a} \frac{w_j}{\sum_{j' \in S_a} w_{j'}} \,. \tag{1}$$

Let $\mathbf{P} := [P_{ij}]$. One can verify that $\mathbf{P}$ corresponds to a valid transition probability matrix as it is non-negative and row stochastic. Furthermore, $\mathbf{P}$ defines a reversible Markov chain as it satisfies the detailed balance equations

$$w_i \, d_i \, P_{ij} = w_j \, d_j \, P_{ji} \,,$$

for all $i, j \in [n]$. If the graph G_c is connected then $\boldsymbol{\pi} = \mathbf{D}\,\mathbf{w}/\|\mathbf{D}\,\mathbf{w}\|_1$ is the unique stationary distribution of $\mathbf{P}$, and one can recover the true weight vector $\mathbf{w}$ from this stationary distribution using a linear transform $\mathbf{D}^{-1}$.

In practice one does not have access to $\mathbf{P}$, so we propose an *empirical* estimate of $\mathbf{P}$ that can be computed from the given comparison data. Formally, define $\hat{p}_{i|S_a}$ to be the fraction of times that i won a m-way comparison amongst items in the

[1] Throughout this paper we will use the terminology Markov chain and random walk interchangeably.

Algorithm 1 ASR

Input Markov chain $\hat{\mathbf{P}}$ according to Eq. (2)
Initialize $\hat{\boldsymbol{\pi}} = (\frac{1}{n}, \cdots, \frac{1}{n})^\top \in \Delta_n$
while estimates do not converge **do**
  $\hat{\boldsymbol{\pi}} \leftarrow \hat{\mathbf{P}}^\top \hat{\boldsymbol{\pi}}$
end while
Output $\hat{\mathbf{w}} = \frac{\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}}{\|\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}\|_1}$

set S_a, i.e. $\hat{p}_{i|S_a} := \frac{1}{L}\sum_{l=1}^{L} \mathbf{1}[y_a^l = i]$. Let us then define a random walk where the probability of transition from node $i \in [n]$ to node $j \in [n]$ is given by

$$\hat{P}_{ij} := \frac{1}{d_i} \sum_{a \in [d]: i,j \in S_a} \hat{p}_{j|S_a} \,. \tag{2}$$

Let $\hat{\mathbf{P}} := [\hat{P}_{ij}]$. One can again verify that $\hat{\mathbf{P}}$ corresponds to a valid transition probability matrix. We can think of $\hat{\mathbf{P}}$ as a perturbation of $\mathbf{P}$, with the error due to perturbation decreasing with more and more comparisons. There is a rich literature (Cho & Meyer, 2001; Mitrophanov, 2005) on analyzing sensitivity of the stationary distribution of a Markov chain under small perturbations. Hence, given a large number of comparisons, one can expect the stationary distribution of $\hat{\mathbf{P}}$ to be close to that of $\mathbf{P}$. Since we take a linear transform of these stationary distributions, one also needs to show that closeness is preserved under this linear transform. We defer this analysis to Section 5.

The pseudo-code for our algorithm is given in Algorithm 1. The algorithm computes the stationary distribution $\hat{\boldsymbol{\pi}}$ of the Markov chain $\hat{\mathbf{P}}$ using the power method.[2] It then outputs the (normalized) vector $\hat{\mathbf{w}}$ that is obtained after applying the linear transform $\mathbf{D}^{-1}$ to $\hat{\boldsymbol{\pi}}$, i.e. $\hat{\mathbf{w}} = \frac{\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}}{\|\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}\|_1}$. In the next section we will compare the convergence time of our algorithm with previous algorithms (Negahban et al., 2017; Maystre & Grossglauser, 2015).

4. Comparison of Mixing Time with Rank Centrality (RC) and Luce Spectral Ranking (LSR)

The random walk $\mathbf{P}^{\text{RC}}$ constructed by the RC (Negahban et al., 2017) algorithm for the BTL model is given by

$$P_{ij}^{\text{RC}} := \begin{cases} \frac{1}{d_{\max}} \sum_{a \in [d]: i,j \in S_a} p_{j|S_a} & \text{if } i \neq j \\ 1 - \frac{1}{d_{\max}} \sum_{j' \neq i} P_{ij'}^{\text{RC}} & \text{if } i = j \end{cases}, \tag{3}$$

[2]The stationary distribution of the Markov chain may also be computed using other linear algebraic techniques, but these techniques typically have a running time of $O(n^3)$ which is impractical for most modern applications.

and the random walk $\mathbf{P}^{\text{LSR}}$ constructed by LSR (Maystre & Grossglauser, 2015) for the MNL model is given by

$$P_{ij}^{\text{LSR}} := \begin{cases} \epsilon \sum_{a \in [d]: i,j \in S_a} p_{j|S_a} & \text{if } i \neq j \\ 1 - \epsilon \sum_{j' \neq i} P_{ij'}^{\text{LSR}} & \text{if } i = j \end{cases}, \tag{4}$$

where $\epsilon > 0$ is chosen such that the diagonal entries are non-negative. In general ϵ would be $O(\frac{1}{d_{\max}})$. The random walks $\hat{\mathbf{P}}^{\text{RC}}$ and $\hat{\mathbf{P}}^{\text{LSR}}$ constructed from the comparison data are defined analogously using empirical probabilities $\hat{p}_{j|S_a}$ instead of $p_{j|S_a}$.

We first begin by showing that for any given comparison data $\mathbf{Y}$, both RC/LSR and our algorithm will return the same estimate upon convergence.

Proposition 1. *Given items $[n]$ and comparison data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^d$, let $\hat{\boldsymbol{\pi}}$ be the stationary distribution of the Markov chain $\hat{\mathbf{P}}$ constructed by* ASR*, and let $\hat{\mathbf{w}}^{LSR}$ be the stationary distribution of the Markov chain $\hat{\mathbf{P}}^{LSR}$. Then $\hat{\mathbf{w}}^{LSR} = \frac{\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}}{\|\mathbf{D}^{-1}\hat{\boldsymbol{\pi}}\|_1}$. The same result is also true for $\hat{\mathbf{w}}^{RC}$ for the case of pairwise comparisons.*

We give a proof of this result in the supplementary material. Although the above lemma shows that in a convergent state both these algorithms will return the same estimates, it does not say anything about the time it takes to reach this convergent state. This is where the *key difference* lies.

Observe that each row $i \in [n]$ of our matrix $\mathbf{P}$ is divided by d_i, whereas each row of $\mathbf{P}^{\text{RC}}$ is divided by $d_{\max}$ except the diagonal entries. Now if $d_{\max}$ is very large, a row $i \in [n]$ of $\mathbf{P}^{\text{RC}}$ that corresponds to an item i with small d_i would have very small non-diagonal entries. This can make the diagonal entry P_{ii}^{RC} very large, which amounts to having a heavy self loop at node i. This heavy self loop can significantly reduce the time it takes for the random walk to reach its stationary distribution, since a lot of transitions starting from i will return back to i. The same analysis holds true for LSR under multiway comparisons.

To formalize this intuition, we need to analyze the spectral gap of a random walk $\mathbf{X}$, which we denote by $\mu(\mathbf{X})$, which plays an important role in determining its mixing time. The spectral gap of a reversible random walk (or Markov chain) $\mathbf{X}$ is defined as $\mu(\mathbf{X}) := 1 - \lambda_2(\mathbf{X})$, where $\lambda_2(\mathbf{X})$ is the second largest eigenvalue of $\mathbf{X}$ in terms of absolute value. The following lemma (see Levin et al. (2008) for more details) gives both upper and lower bounds on the mixing time (w.r.t. the total variation distance) of a random walk in terms of the spectral gap.

Lemma 1. *(Levin et al., 2008) Let $\mathbf{X}$ be the transition probability matrix of a reversible, irreducible Markov chain with state space $[n]$, $\boldsymbol{\pi}$ be the stationary distribution of $\mathbf{X}$, and $\pi_{\min} := \min_{i \in [n]} \pi_i$, and let*

$$d(r) = \sup_{\mathbf{p} \in \Delta_n} \|\mathbf{p}\mathbf{X}^r - \boldsymbol{\pi}\|_{TV} \,.$$

For any $\gamma > 0$, let $t(\gamma) = \min\{r \in \mathbb{N} : d(r) \leq \gamma\}$; then

$$\log(\frac{1}{2\gamma})\Big(\frac{1}{\mu(\mathbf{X})} - 1\Big) \leq t(\gamma) \leq \log(\frac{1}{\gamma\pi_{\min}})\frac{1}{\mu(\mathbf{X})}\,.$$

The above lemma states that the mixing time of a Markov chain $\mathbf{X}$ is inversely proportional to its spectral gap $\mu(\mathbf{X})$. Now, we will compare the spectral gap of our Markov chain $\mathbf{P}$ with the spectral gap of $\mathbf{P}^{\text{RC}}$ (and $\mathbf{P}^{\text{LSR}}$).

Proposition 2. *Let the probability transition matrix $\mathbf{P}$ for our random walk be as defined in Eq. (1). Let $\mathbf{P}^{RC}$ and $\mathbf{P}^{LSR}$ be as defined in Eq. (3) and Eq. (4), respectively. Then*

$$\frac{d_{\min}}{d_{\max}}\mu(\mathbf{P}) \leq \mu(\mathbf{P}^{RC}) \leq \mu(\mathbf{P})\,, \tag{5}$$

and

$$\epsilon d_{\min}\mu(\mathbf{P}) \leq \mu(\mathbf{P}^{LSR}) \leq \mu(\mathbf{P})\,, \tag{6}$$

where $\epsilon = O(\frac{1}{d_{\max}})$.

A formal proof of this lemma is given in the supplementary material, and uses comparison theorems for reversible Markov chains due to Diaconis & Saloff-Coste (1993). This lemma shows that the spectral gap of $\mathbf{P}$ is always lower bounded by that of $\mathbf{P}^{\text{RC}}$ (and $\mathbf{P}^{\text{LSR}}$), but can be much larger than it. In the latter case one would observe, using Lemma 1, that our algorithm will converge faster than the RC algorithm (and LSR). In fact there are instances where $O(d_{\max}/d_{\min}) = \Omega(n)$ and the leftmost inequalities in both Eq. (5) and Eq. (6) hold with equality. In these instances the convergence of our algorithm will be $\Omega(n)$ times faster. We give examples of two such instances.

Example 1. *Let $n = 3$, $m = 2$, $w_1 = 1/2, w_2 = 1/4$ and $w_3 = 1/4$. In the comparison data 1 is compared to both 2 and 3; but items 2 and 3 are not compared to each other. This implies that $d_1 = 2$, and $d_i = 1$ for $i \neq 1$. One can calculate the matrices $\mathbf{P}$ and $\mathbf{P}^{RC}$, and their respective eigenvalues, and observe that $\mu(\mathbf{P}) = 2\mu(\mathbf{P}^{RC})$.*

Example 2. *Let $m = 2$, $\mathbf{w} = (1/n, \cdots, 1/n)^\top$, and the comparison data be such that item 1 is compared to every other item, and no other items are compared to each other. This implies that $d_1 = n-1$, and $d_i = 1$ for $i \neq 1$. One can calculate the matrix $\mathbf{P}$ and $\mathbf{P}^{RC}$ again, and their respective eigenvalues, and observe that $\mu(\mathbf{P}) = (n-1)\cdot\mu(\mathbf{P}^{RC})$.*

Note that in the above lemma, we only show the relation between the spectral gaps of the matrices $\mathbf{P}$ and $\mathbf{P}^{\text{RC}}$, and not for any particular realization $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}^{\text{RC}}$. If the Markov chains $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}^{\text{RC}}$ are reversible, then identical results hold. However, similar results are very hard to prove for non-reversible Markov chains (Dyer et al., 2006). Nevertheless, for large L, one can expect the realized matrices $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}^{\text{RC}}$ to be close to their expected matrices $\mathbf{P}$ and $\mathbf{P}^{\text{RC}}$, respectively. Hence, using eigenvalue perturbation bounds (Horn & Johnson, 1990), one can show that the spectrum of $\hat{\mathbf{P}}$ and $\hat{\mathbf{P}}^{\text{RC}}$ is close to the spectrum of $\mathbf{P}$ and $\mathbf{P}^{\text{RC}}$, respectively. The same analysis holds true for LSR under multiway comparisons. In Section 7 we perform experiments on synthetic and real world datasets which empirically show that the mixing times of the realized Markov chains behave as predicted.

It has been observed that faster mixing rates of Markov chains gives us the ability to prove sharper perturbation bounds for these Markov chains (Mitrophanov, 2005). In the following section we will use these perturbation bounds to prove sharper sample complexity bounds for our algorithm.

5. Sample Complexity Bounds

In this section we will present sample complexity bounds for the estimates returned by ASR in terms of total variation distance. The following theorem gives an error bound in terms of the total variation distance for estimates $\hat{\mathbf{w}}$ of the MNL weights returned by our algorithm

Theorem 1. *Given items $[n]$ and comparison data $\mathbf{Y} = \{(S_a, \mathbf{y}_a)\}_{a=1}^d$, let each set S_a of cardinality m be compared L times, with outcomes $\mathbf{y}_a = (y_a^1, \cdots, y_a^L)$ produced as per a MNL model with parameters $\mathbf{w} = (w_1, \ldots, w_n)$, such that $\|\mathbf{w}\|_1 = 1$. If the random walk $\hat{\mathbf{P}}$ (Eq. (2)) on the comparison graph $G_c([n], E)$ induced by the comparison data $\mathbf{Y}$ is strongly connected, then the* ASR *algorithm (Algorithm 1) converges to a unique distribution $\hat{\mathbf{w}}$, which with probability $\geq 1 - 3n^{-(C^2-50)/25}$ satisfies the following error bound*[3]

$$\|\mathbf{w} - \hat{\mathbf{w}}\|_{TV} \leq \frac{C\,\kappa\,d_{\text{avg}}}{\mu(\mathbf{P})\,d_{\min}}\sqrt{\frac{\max\{m, \log(n)\}}{L}}\,,$$

where $\kappa = \log\left(\frac{d_{\text{avg}}}{d_{\min} w_{\min}}\right)$, $w_{\min} = \min_{i\in[n]} w_i$, $d_{\text{avg}} = \sum_{i\in[n]} w_i d_i$, $d_{\min} = \min_{i\in[n]} d_i$, $\mu(\mathbf{P})$ is the spectral gap of the random walk $\mathbf{P}$ (Eq. (1)), and C is any constant.

Recall from Section 3 that the Markov chain $\hat{\mathbf{P}}$ can be viewed as a perturbation of $\mathbf{P}$. To show that the stationary distributions of $\hat{\mathbf{P}}$ and $\mathbf{P}$ are close, we use the results of Mitrophanov (2005) on the stability of Markov chains under perturbations. We also show closeness is preserved under the linear transformation $\mathbf{D}^{-1}$, giving the final bound stated in the aforementioned theorem. We present a formal proof in the supplementary material.

In the error bound of Theorem 1, one can further bound the spectral gap $\mu(\mathbf{P})$ of $\mathbf{P}$ in terms of the spectral gap of the *random walk normalized Laplacian* of G_c, which is a

[3] The dependence on κ is due to the dependence on $\frac{1}{\pi_{\min}}$ in the mixing time upper bounds in Lemma 1. There are other bounds for κ in terms of the condition number for Markov chains, for example see (Mitrophanov, 2005), and any improvement on these bounds will lead to an improvement in our sample complexity. In the worst case, κ has a trivial upper bound of $O(\log n)$.

fundamental quantity associated with G_c. The Laplacian represents a random walk on G_c that transitions from a node i to one of its neighbors uniformly at random. Formally, the Laplacian $\mathbf{L} := \mathbf{C}^{-1}\mathbf{A}$, where $\mathbf{C}$ is a diagonal matrix with $C_{ii} = \left|\bigcup_{a\in[d]:i\in S_a} S_a\right|$, i.e. the number of unique items i was compared with, and $\mathbf{A}$ is the adjacency matrix, such that for $i, j \in [n]$, $A_{ij} = 1$ if $(i, j) \in E$, and $A_{ij} = 0$ otherwise. Let $\xi := \mu(\mathbf{L})$ be the spectral gap of $\mathbf{L}$. Then we can lower bound $\mu(\mathbf{P})$ as follows (proof in the supplement)

$$\mu(\mathbf{P}) \geq \frac{\xi}{m\, b^2}\,,$$

where b is the ratio of the maximum to the minimum weight, i.e. $b = \max_{i,j\in[n]} w_i/w_j$. This gives us the following.

Corollary 1. *In the setting of Theorem 1, the* ASR *algorithm converges to a unique distribution* $\hat{\mathbf{w}}$*, which with probability* $\geq 1 - 3n^{-(C^2-50)/25}$ *satisfies the following error bound:*

$$\|\mathbf{w} - \hat{\mathbf{w}}\|_{TV} \leq \frac{C\, m\, b^2\, \kappa\, d_{\text{avg}}}{\xi\, d_{\min}} \sqrt{\frac{\max\{m, \log(n)\}}{L}}\,,$$

where $b = \max_{i,j\in[n]} \frac{w_i}{w_j}$.

In the discussion that follows, we will assume $b = O(1)$, and hence, $\mu(\mathbf{P}) = \Omega(\xi/m)$. The quantity d_{avg} has an interesting interpretation: it is the weighted average of the number of sets in which each item was shown. It has a trivial upper bound of $d_{\max}$, however, a careful analysis will reveal a better bound of $O(|E|/n)$ where E is the set of edges in the comparison graph G_c. Using this observation we can give the following corollary of the above theorem.

Corollary 2. *If the conditions of Theorem 1 are satisfied, and if the number of edges in the comparison graph* G_c *are* $O(n\,\text{poly}(\log n))$*, i.e.* $|E| = O(n\,\text{poly}(\log n))$*, then in order to ensure a total variation error of* $o(1)$*, the required number of comparisons per set is upper bounded as*

$$L = O\big(\mu(\mathbf{P})^{-2}\,\text{poly}(\log n)\big) = O\big(\xi^{-2}\, m^3\,\text{poly}(\log n)\big)\,.$$

Hence, the sample complexity, i.e. total number of m*-way comparisons needed to estimate* $\mathbf{w}$ *with error* $o(1)$*, is given by* $|E| \times L = O\big(\xi^{-2}\, m^3\, n\,\text{poly}(\log n)\big)$.

We again provide a proof of this corollary in the appendix. Note that the case when the total number of edges in the comparison graph is $O(n\,\text{poly}(\log n))$ captures the most interesting case in ranking and sorting. Also, in most practical settings the size m of comparison sets will be $O(\log n)$. In this case, the above corollary implies a sample complexity bound of $O\big(\xi^{-2}\, n\,\text{poly}(\log n)\big)$, which is sometimes referred to as *quasi-linear* complexity. The following simple example illustrates this sample complexity bound.

Example 3. *Consider a star comparison graph, discussed in Example 2, where there is one item* $i \in [n]$ *that is compared to all other* $n - 1$ *items, and no other items are compared to each other. Let* $\mathbf{w} = (\frac{1}{n}, \cdots, \frac{1}{n})^\top$*. One can calculate the spectral gap* $\mu(\mathbf{P})$ *to be* 0.5 *exactly. In this case, the sample complexity bound given by our result is* $O(n\,\text{poly}(\log n))$.

Discussion/Comparison. For the special case of pairwise comparisons under the BTL model ($m = 2$), Negahban et al. (2017) give a sample complexity bound of $O\big(\frac{d_{\max}}{d_{\min}}\xi^{-2}\, n\,\text{poly}(\log n)\big)$ for recovering the estimates $\hat{\mathbf{w}}$ with low (normalized) L_2 error. Using Proposition 1 one can see that this bound also applies to the estimates returned by our algorithm, and our bound in terms of L_1 applies to rank centrality as well. However, the bounds due to Negahban et al. (2017) have a dependence on the ratio $\frac{d_{\max}}{d_{\min}}$ due to the large spectral gap of their Markov chain as compared to ξ, the spectral gap of the Laplacian. In Section 7 we show that for many real world datasets $\frac{d_{\max}}{d_{\min}}$ can be much larger than $\log n$, and hence, their bounds are no longer quasi-linear. A large class of graphs that occur in many real world scenarios and exhibit this behavior are the power-law graphs. Another real world scenario in which $\frac{d_{\max}}{d_{\min}} = \Omega(n)$ arises is choice modeling (Agrawal et al., 2016), where one explicitly models the *'no choice option'* where the user has an option of not selecting any item from the set of items presented to her. In this case the 'no choice option' will be present in each comparison set, and the comparison graph will behave like a star graph discussed in Example 2. In fact for such graphs, the results of (Negahban et al., 2017) give a trivial bound of $\text{poly}(n)$ in terms of the L_2 error.

For the general case of multiway comparisons we are not aware of any other sample complexity bounds. It is also important to note that the dependence on the number of comparison sets comes only through the spectral gap ξ of the natural random walk on the comparison graph. For example, if the graph is a cycle ($d = n$), then the spectral gap is $O(1/n^2)$, whereas if the graph is a clique ($d = O(n^2)$) the spectral gap is $O(1)$.

6. Message Passing Interpretation of ASR

In this section, we show our spectral ranking algorithm can be interpreted as a message passing/belief propagation algorithm. This connection can be used to design a decentralized distributed version of our algorithm.

Let us introduce the *factor graph*, which is an important data structure used in message passing algorithms. The factor graph is a bipartite graph $G_f([n] \cup [d], E_f)$ which has two type of nodes– *item nodes* which correspond to the n items, and *set nodes* which correspond to the d sets. More formally, there is an item node i for each item $i \in [n]$, and there is a set node a for each set S_a, $\forall a \in [d]$. There is an edge $(i, a) \in E_f$ between node i and a if and only if $i \in S_a$. There is a weight $\hat{p}_{i|S_a}$ on the edge (i, a) which corresponds

to the fraction of times i won in the set S_a.

Algorithm 2 Message Passing

Input Graph $G_f = ([n] \cup [d], E_f)$, edge $(i, a) \in E$ has weight $\hat{p}_{i|S_a}$
Initialize Set $m^{(0)}_{a\to i} \leftarrow m/n, \forall a \in [d], \forall i \in S_a$
for $t = 1, 2, \cdots$ until convergence **do**
  for all $i \in [n]$ **do** $m^{(t)}_{i\to a} = \frac{1}{d_i} \sum_{a':i\in S_{a'}} \hat{p}_{i|S_{a'}} \cdot m^{(t-1)}_{a'\to i}$
  for all $a \in [d]$ **do** $m^{(t)}_{a\to i} = \sum_{i'\in S_a} m^{(t)}_{i'\to a}$
end for
Set $\hat{w}_i \leftarrow m^{(t-1)}_{i\to a}, \forall i \in [n]$
Output $\hat{\mathbf{w}}/\|\hat{\mathbf{w}}\|_1$

We shall now describe the algorithm. In each iteration of this algorithm, the item nodes send a message to their neighboring set nodes, and the set nodes respond to these messages. A message from an item node i to a set node a represents an estimate of the weight w_i of item i, and a message from a set node a to an item i represents an estimate of the sum of weights of items contained in set S_a.

In each iteration, the item nodes update their estimates based on the messages they receive in the previous iteration, and send these estimates to their neighboring set nodes. The set nodes then update their estimate by summing up the messages they receive from their neighboring item nodes, and then send these estimates to their neighboring item nodes. This process continues until the messages converge.

Formally, let $m^{(t-1)}_{i\to a}$ be the message from item node i to set node a in iteration $t-1$, and $m^{(t-1)}_{a\to i}$ be the corresponding message from the set node a to item node i. Then the messages in the next iteration are updated as follows:

$$m^{(t)}_{i\to a} = \frac{1}{d_i} \sum_{a'\in[d]:i\in S_{a'}} \hat{p}_{i|S_{a'}} \cdot m^{(t-1)}_{a'\to i} \,,$$

$$m^{(t)}_{a\to i} = \sum_{i'\in S_a} m^{(t)}_{i'\to a} \,.$$

Now, suppose that the empirical edge weights $\hat{p}_{i|S_a}$ are equal to the true weights $p_{i|S_a} = \frac{w_i}{\sum_{j\in S_a} w_j}, \forall i \in [n], a \in [d]$. Also, suppose on some iteration $t \geq 1$, the item messages $m^{(t)}_{i\to a}$ become equal to the item weights $w_i, \forall i \in [n]$. Then it is easy to observe that the next iteration of messages $m^{(t+1)}_{i\to a}$ are also equal to w_i. Therefore, the true weights $\mathbf{w}$, in some sense, are a fixed point of the above set of equations. The following lemma shows that the ASR algorithm is equivalent to this message passing algorithm.

Lemma 2. *For any realization of comparison data* $\mathbf{Y}$*, there is a one-to-one correspondence d each iteration of the message passing algorithm (2) and the corresponding power iteration of the ASR algorithm (1), and both algorithms return the same estimates* $\hat{\mathbf{w}}$ *for any* $\mathbf{Y}$.

We give a proof of the above lemma in the supplementary material. The above lemma gives an interesting connection between spectral ranking under the MNL model and message passing/belief propagation. Such connections have been observed for other problem such as the problem of aggregating crowdsourced binary tasks (Khetan & Oh, 2016). A consequence of this connection is that it facilitates a fully decentralized distributed implementation of the ASR algorithm. This can be very useful for modern applications, where machines can communicate local parameter updates to each other, without explicitly communicating the data.

7. Experiments

In this section we perform experiments on both synthetic and real data to compare our algorithm to the existing LSR (Maystre & Grossglauser, 2015) and RC (Negahban et al., 2017) algorithms for recovering the weight vector $\mathbf{w}$ under the MNL and BTL model, respectively. The implementation[4] of our algorithm is based on applying the power method on $\hat{\mathbf{P}}$ (Eq. (2)). The power method was chosen due to its simplicity, efficiency, and scalability to large problem sizes. Similarly, the implementations of LSR and RC are based on applying the power method on $\hat{\mathbf{P}}^{\text{LSR}}$ (Eq. (4)), and $\hat{\mathbf{P}}^{\text{RC}}$ (Eq. (3)), respectively. In the definition of $\hat{\mathbf{P}}^{\text{LSR}}$, the parameter ϵ was chosen to be the maximum possible value that ensures $\hat{\mathbf{P}}^{\text{LSR}}$ is a Markov chain.

7.1. Synthetic Data

We conducted experiments on synthetic data generated according to the MNL model, with weight vectors $\mathbf{w}$ generated randomly (details below). We compared our algorithm with the LSR algorithm for comparison sets of size $m = 5$, and with the RC algorithm for sets of size $m = 2$. We used two different graph topologies for generating the comparison graph G_c, or equivalently the comparison sets:

1. **Random Topology:** This graph topology corresponds to random graphs where $n \log_2(n)$ comparison sets are chosen uniformly at random from all the $\binom{n}{m}$ unique sets of cardinality m. This topology is very close to the Erdős-Rényi topology which has been well-studied in the literature. In fact the degree distributions of nodes in this random topology are very close to the degree distributions in the Erdős-Rényi topology (Mezard & Montanari, 2009). The only reason we study the former is computational, as iterating over all $\binom{n}{m}$ hyper-edges is computationally challenging.

2. **Star Topology:** In this graph topology, there is a single item that belongs to all sets; the remaining $(m-1)$ items in each set are contained only in that set. We study this topology because it corresponds to the choice sets used in Example 2, where there was a factor of $\Omega(n)$ gap in the

[4] code available: https://github.com/agarpit/asr

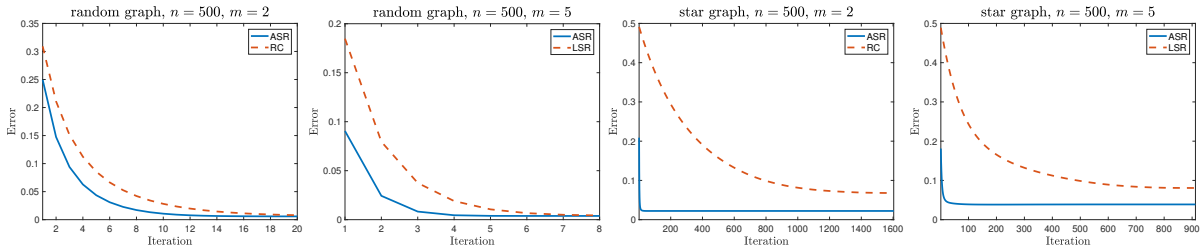

Figure 1. Results on synthetic data: L_1 error vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for $m = 2$) and the LSR algorithm (for $m = 5$), on data generated from the MNL/BTL model with the random and star graph topologies.

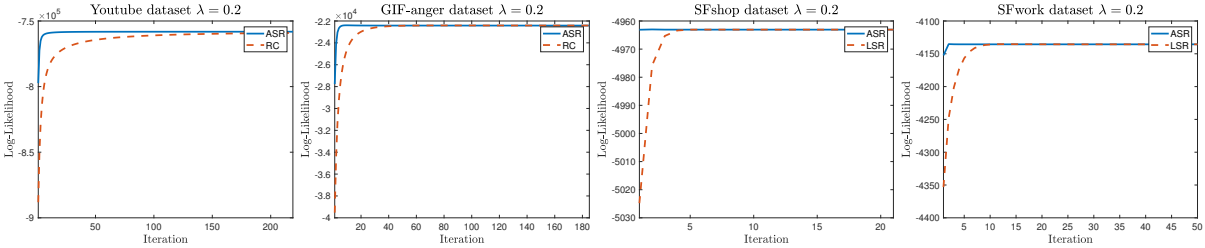

Figure 2. Results on real data: Log-likelihood vs. number of iterations for our algorithm, ASR, compared with the RC algorithm (for pairwise comparison data) and the LSR algorithm (for multi-way comparison data), all with regularization parameter set to 0.2.

spectral gap between our algorithm and the other algorithms.

In our experiments we selected $n = 500$[5], and the weight w_i of each item $i \in [n]$ was drawn uniformly at random from the range $(0, 1)$; the weights were then normalized so they sum to 1. A comparison graph G_c was generated according to each of the graph topologies above. The parameter L was set to $300 \log_2 n$. The winner for each comparison set was drawn according to the MNL model with weights $\mathbf{w}$. The convergence criterion for all algorithms was the same: we run the algorithm until the L_1 distance between the new estimates and the old estimates is ≤ 0.0001. Each experiment was repeated 100 times and the average values over all trials are reported. For $n = 500$, $m \in \{2, 5\}$, and both graph topologies described above, we compared the convergence as a function of the number of iterations[6] for each algorithm. We plotted the L_1 error of the estimates produced by these algorithms after each iteration. The plots are given in Figure 1. These plots verify the mixing time analysis of Section 4, and show that our algorithm converges much faster than RC and LSR, and orders of magnitude faster in the case of the star graph.

7.2. Real World Datasets

We conducted experiments on the YouTube dataset (Shetty, 2012), GIF-anger dataset (Rich et al.), and the SFwork and SFshop (Koppelman & Bhat, 2006) datasets. Table 1 gives some statistics about these datasets. We also plot the degree distributions of these datasets in the supplementary material.

[5]Results for other values of n are given in the supplement.

[6]We also plotted the convergence as a function of the running time; the results were similar as the running time of each iteration is similar for all these algorithm.

Table 1. Statistics for real world datasets

Dataset	n	m	d	$d_{\max}/d_{\min}$
Youtube	21207	2	394007	600
GIF-anger	6119	2	64830	106
SFwork	6	3-6	12	4.3
SFshop	8	4-8	10	1.9

For these datasets, a ground-truth $\mathbf{w}$ is either unknown or undefined; and hence, we compare our algorithm and the RC/LSR algorithm with respect to the log-likelihood of the estimates as a function of number of iterations. Due to the number of comparisons per set (or pair) being very small, in order to ensure irreducibility of random walks, we use a regularized version of all algorithms (see supplementary material, and also Section 3.3 in Negahban et al. (2017), for more details). Here, we give results when the regularization parameter λ is set to 0.2, and defer the results for other parameter values to the supplementary material. The results are given in Figure 2. We observe that our algorithm converges rapidly to the peak log-likelihood value while RC and LSR are always slower in converging to this value.

8. Conclusion and Future Work

We presented a spectral algorithm for the problem of rank aggregation from pairwise and multiway comparisons. Our algorithm is considerably faster than previous algorithms; in addition, our analysis yields improved sample complexity results for estimation under the BTL and MNL model. We also give a message passing/belief propagation interpretation for our algorithm. It would be interesting to see if one can use our algorithm to give better guarantees for recovery of top-k items under MNL.

Acknowledgements

We would like to thank Rohan Ghuge and Ashish Khetan for helpful discussions. We would also like to thank the anonymous reviewers for helpful comments. This material is based upon work supported by the US National Science Foundation under Grant No. 1717290. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Agarwal, A., Agarwal, S., Assadi, S., and Khanna, S. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *COLT*, 2017.

Agrawal, S., Avadhanula, V., Goyal, V., and Zeevi, A. A near-optimal exploration-exploitation approach for assortment selection. In *EC*, 2016.

Arrow, K. J. Social choice and individual values, 1963.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chen, X., Li, Y., and Mao, J. A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *SODA*, 2017.

Chen, Y. and Suh, C. Spectral MLE: Top-k rank aggregation from pairwise comparisons. In *ICML*, 2015.

Cho, G. and Meyer, C. Comparison of perturbation bounds for the stationary distribution of a Markov chain. *Linear Algebra and its Applications*, 335(1-3):137–150, 2001.

Condorcet, M. d. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, 1785.

de Borda, J. C. Mémoire sur les élections au scrutin. 1781.

Devroye, L. The equivalence of weak, strong and complete convergence in l1 for kernel density estimates. *The Annals of Statistics*, pp. 896–904, 1983.

Diaconis, P. and Saloff-Coste, L. Comparison theorems for reversible Markov chains. *The Annals of Applied Probability*, pp. 696–730, 1993.

Diaconis, P. and Stroock, D. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, pp. 36–61, 1991.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank aggregation methods for the web. In *WWW*, 2001.

Dyer, M., Goldberg, L. A., Jerrum, M., Martin, R., et al. Markov chain comparison. *Probability Surveys*, 3:89–111, 2006.

Guiver, J. and Snelson, E. Bayesian inference for Plackett-Luce ranking models. In *ICML*, 2009.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 1990.

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

Jang, M., Kim, S., Suh, C., and Oh, S. Top-k Ranking from Pairwise Comparisons: When Spectral Ranking is Optimal. *arXiv preprint arXiv:1603.04153*, 2016.

Jang, M., Kim, S., Suh, C., and Oh, S. Optimal sample complexity of m-wise data for top-k ranking. In *NIPS*, 2017.

Khetan, A. and Oh, S. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *NIPS*, 2016.

Koppelman, F. S. and Bhat, C. A self instructing course in mode choice modeling: multinomial and nested logit models. *US Department of Transportation, Federal Transit Administration*, 2006.

Levin, D. A., Peres, Y., and Wilmer, E. L. *Markov Chains and Mixing Times*. American Mathematical Society, Providence, RI, USA, 2008.

Luce, D. R. Individual choice behavior. 1959.

Maystre, L. and Grossglauser, M. Fast and accurate inference of plackett-luce models. In *NIPS*, 2015.

McFadden, D. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pp. 105–142, 1974.

Mezard, M. and Montanari, A. *Information, Physics, and Computation*. Oxford University Press, Inc., New York, NY, USA, 2009.

Mitrophanov, A. Y. Sensitivity and convergence of uniformly ergodic Markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005.

Negahban, S., Oh, S., and Shah, D. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65 (1):266–287, 2017.

Plackett, R. L. The analysis of permutations. *Applied Statistics*, pp. 193–202, 1975.

Rajkumar, A. and Agarwal, S. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *ICML*, 2014.

Rich, T., Hu, K., and Tome, B. GIFGIF dataset. Data Available: http://www.gif.gf.

Shetty, S. Quantifying comedy on YouTube: why the number of o's in your LOL matter. Data Available: https://archive.ics.uci.edu/ml/datasets/YouTube+Comedy+Slam+Preference+Data, 2012.

Soufiani, H. A., Chen, W. Z., Parkes, D. C., and Xia, L. Generalized method-of-moments for rank aggregation. In *NIPS*, 2013.