Iterative Ranking from Pair-wise Comparisons

Sahand Negahban
Department of EECS
Massachusetts Institute of Technology
sahandn@mit.edu

Sewoong Oh
Department of IESE
University of Illinois at Urbana Champaign
swoh@illinois.edu

Devavrat Shah
Department of EECS
Massachusetts Institute of Technology
devavrat@mit.edu

Abstract

The question of aggregating pairwise comparisons to obtain a global ranking over a collection of objects has been of interest for a very long time: be it ranking of online gamers (e.g. MSR's TrueSkill system) and chess players, aggregating social opinions, or deciding which product to sell based on transactions. In most settings, in addition to obtaining ranking, finding 'scores' for each object (e.g. player's rating) is of interest to understanding the intensity of the preferences.

In this paper, we propose a novel iterative rank aggregation algorithm for discovering scores for objects from pairwise comparisons. The algorithm has a natural *random walk* interpretation over the graph of objects with edges present between two objects if they are compared; the scores turn out to be the stationary probability of this random walk. The algorithm is model independent. To establish the efficacy of our method, however, we consider the popular Bradley-Terry-Luce (BTL) model in which each object has an associated score which determines the probabilistic outcomes of pairwise comparisons between objects. We bound the finite sample error rates between the scores assumed by the BTL model and those estimated by our algorithm. This, in essence, leads to order-optimal dependence on the number of samples required to learn the scores well by our algorithm. Indeed, the experimental evaluation shows that our (model independent) algorithm performs as well as the Maximum Likelihood Estimator of the BTL model and outperforms a recently proposed algorithm by Ammar and Shah [1].

1 Introduction

Rank aggregation is an important task in a wide range of learning and social contexts arising in recommendation systems, information retrieval, and sports and competitions. Given n items, we wish to infer relevancy scores or an ordering on the items based on partial orderings provided through many (possibly contradictory) samples. Frequently, the available data that is presented to us is in the form of a comparison: player A defeats player B; book A is purchased when books A and B are displayed (a bigger collection of books implies multiple pairwise comparisons); movie A is liked more compared to movie B. From such partial preferences in the form of comparisons, we frequently wish to deduce not only the order of the underlying objects, but also the scores associated with the objects themselves so as to deduce the intensity of the resulting preference order. For example, the Microsoft TrueSkill engine assigns scores to online gamers based on the outcomes of (pairwise) games between players. Indeed, it assumes that each player has inherent "skill" and the

outcomes of the games are used to learn these skill parameters which in turn lead to scores associate with each player. In most such settings, similar model-based approaches are employed.

In this paper, we have set out with the following goal: develop an algorithm for the above stated problem which (a) is computationally simple, (b) works with available (comparison) data only and does not try to fit any model per se, (c) *makes sense* in general, and (d) if the data indeed obeys a reasonable model, then the algorithm should do as well as the best model aware algorithm. The main result of this paper is an affirmative answer to all these questions.

Related work. Most rating based systems rely on users to provide explicit numeric scores for their interests. While these assumptions have led to a flurry of theoretical research for item recommendations based on matrix completion [2, 3, 4], it is widely believed that numeric scores provided by individual users are generally inconsistent. Furthermore, in a number of learning contexts as illustrated above, it is simply impractical to ask a user to provide explicit scores.

These observations have led to the need to develop methods that can aggregate such forms of ordering information into relevance ratings. In general, however, designing consistent aggregation methods can be challenging due in part to possible contradictions between individual preferences. For example, if we consider items A, B, and C, one user might prefer A to B, while another prefers B to C, and a third user prefers C to A. Such problems have been well studied as in the work by Condorcet [5]. In the celebrated work by Arrow [6], existence of a rank aggregation algorithm with reasonable sets of properties (or axioms) was shown to be impossible.

In this paper, we are interested in a more restrictive setting: we have outcomes of pairwise comparisons between pairs of items, rather than a complete ordering as considered in [6]. Based on those pairwise comparisons, we want to obtain a ranking of items along with a score for each item indicating the intensity of the preference. One reasonable way to think about our setting is to imagine that there is a distribution over orderings or rankings or permutations of items and every time a pair of items is compared, the outcome is generated as per this underlying distribution. With this, our question becomes even harder than the setting considered by Arrow [6] as, in that work, effectively the entire distribution over permutations was already known! Indeed, such hurdles have not stopped the scientific community as well as practical designers from designing such systems. Chess rating systems and the more recent MSR TrueSkill system are prime examples. Our work falls precisely into this realm: design algorithms that work well in practice, makes sense in general, and perhaps more importantly, have attractive theoretical properties under common comparative judgment models.

With this philosophy in mind, in recent work, Ammar and Shah [1] have presented an algorithm that tries to achieve the goal with which we have set out. However, their algorithm requires information about comparisons between all pairs, and for each pair it requires the exact pairwise comparison 'marginal' with respect to the underlying distribution over permutations. Indeed, in reality, not all pairs of items can typically be compared, and the number of times each pair is compared is also very small. Therefore, while an important step is taken in [1], it stops short of achieving the desired goal.

In somewhat related work by Braverman and Mossel [7], the authors present an algorithm that produces an ordering based on $O(n \log n)$ pair-wise comparisons on adaptively selected pairs. They assume that there is an underlying true ranking and one observes noisy comparison results. Each time a pair is queried, we are given the true ordering of the pair with probability $1/2 + \gamma$ for some $\gamma > 0$ which does not depend on the items being compared. One limitation of this model is that it does not capture the fact that in many applications, like chess matches, the outcome of a comparison very much depends on the opponents that are competing.

Such considerations have naturally led to the study of noise models induced by parametric distributions over permutations. An important and landmark model in this class is called the Bradley-Terry-Luce (BTL) model [8, 9], which is also known as the Multinomial Logit (MNL) model (cf. [10]). It has been the backbone of many practical system designs including pricing in the airline industry [11]. Adler et al. [12] used such models to design adaptive algorithms that select the winner from small number of rounds. Interestingly enough, the (near-)optimal performance of their adaptive algorithm for winner selection is matched by our non-adaptive (model independent) algorithm for assigning scores to obtain global rankings of all players.

Our contributions. In this paper, we provide an iterative algorithm that takes the noisy comparison answers between a subset of all possible pairs of items as input and produces scores for each item

as the output. The proposed algorithm has a nice intuitive explanation. Consider a graph with nodes/vertices corresponding to the items of interest (e.g. players). Construct a random walk on this graph where at each time, the random walk is likely to go from vertex i to vertex j if items i and j were ever compared; and if so, the likelihood of going from i to j depends on how often i lost to j. That is, the random walk is more likely to move to a neighbor who has more "wins". How frequently this walk visits a particular node in the long run, or equivalently the stationary distribution, is the score of the corresponding item. Thus, effectively this algorithm captures preference of the given item versus all of the others, not just immediate neighbors: the global effect induced by transitivity of comparisons is captured through the stationary distribution.

Such an interpretation of the stationary distribution of a Markov chain or a random walk has been an effective measure of relative importance of a node in wide class of graph problems, popularly known as the *network centrality* [13]. Notable examples of such network centralities include the random surfer model on the web graph for the version of the PageRank [14] which computes the relative importance of a web page, and a model of a random crawler in a peer-to-peer file-sharing network to assign trust value to each peer in EigenTrust [15].

The computation of the stationary distribution of the Markov chain boils down to 'power iteration' using transition matrix lending to a nice iterative algorithm. Thus, in effect, we have produced an algorithm that (a) is computationally simple and iterative, (b) is model independent and works with the data only, and (c) intuitively makes sense. To establish rigorous properties of the algorithm, we analyze its performance under the BTL model described in Section 2.1.

Formally, we establish the following result: given n items, when comparison results between randomly chosen $O(n\mathsf{poly}(\log n))$ pairs of them are produced as per an (unknown) underlying BTL model, the stationary distribution produced by our algorithm (asymptotically) matches the true score (induced by the BTL model). It should be noted that $\Omega(n \log n)$ is a necessary number of (random) comparisons for any algorithm to even produce a consistent ranking (due to connectivity threshold of random bipartite graph). In that sense, we will see that up to $\mathsf{poly}(\log n)$ factor, our algorithm is optimal in terms of sample complexity. Indeed, the empirical experimental study shows that the performance of our algorithm is identical to the ML estimation of the BTL model. Furthermore, it handsomely outperforms other popular choices including the algorithm by [1].

Some remarks about our analytic technique. Our analysis boils down to studying the induced stationary distribution of the random walk or Markov chain corresponding to the algorithm. Like most such scenarios, the only hope to obtain meaningful results for such 'random noisy' Markov chain is to relate it to stationary distribution of a *known* Markov chain. Through recent concentration of measure results for random matrices and comparison technique using Dirichlet forms for characterizing the spectrum of reversible/self-adjoint operators, along with the known expansion property of the random graph, we obtain the eventual result. Indeed, it is the consequence of such powerful results that lead to near-optimal analytic results.

The remainder of this paper is organized as follows. In Section 2 we will concretely introduce our model, the problem, and our algorithm. In Section 3 we will discuss our main theoretical results. The proofs will be presented in Section 4.

Notation. We use C, C', etc. to denote generic numerical constants. We use A^T to denote the transpose of a matrix. The Euclidean norm of a vector is denoted by $\|x\| = \sqrt{\sum_i x_i^2}$, and the operator norm of a linear operator is denoted by $\|A\|_2 = \max_x x^T Ax / x^T x$. Also define $[n] = \{1, 2, \ldots, n\}$ to be the set of all integers from 1 to n.

2 Model, Problem Statement, and Algorithm

We now present a concrete exposition of our underlying probabilistic model and our problem. We then present our explicit random walk approach to ranking.

2.1 Bradley-Terry-Luce model for comparative judgment

In this section we discuss our model of comparisons between various items. As alluded to above, for the purpose of establishing analytic properties of the algorithm, we will assume comparisons are

governed by the BTL model of pairwise comparisons. However, the algorithm itself operates with data generated in arbitrary manner.

To begin with, there are n items of interest, represented as $[n] = \{1, \ldots, n\}$. We shall assume that for each item $i \in [n]$ that there is an associated weight score $w_i \in \mathbb{R}_+$ (i.e. it's a strictly positive real number). Hence, we may consider the vector $w \in \mathbb{R}_+^n$ to be the associated weight vector of all items. Given a pair of items i and j we will let Y_{ij}^l be 1 if j is preferred over i and 0 otherwise during the l^{th} competition for $1 \leq l \leq k$, where k is the total number of competitions for the pair. Under the BTL model we assume that

$$\mathbb{P}(Y_{ij}^l = 1) = \frac{w_j}{w_i + w_j}. \tag{1}$$

Furthermore, conditioned on the score vector w we assume the the variables $Y_{i,j}^l$ are independent for all i, j, and l. We further assume that given some item i we will compare item j to i with probability d/n. In our setting d will be poly-logarithmic in n. This model is a natural one to consider because over a population of individuals the comparisons cannot be adaptively selected. A more realistic model might incorporate selecting various items with different distributions: for example, the Netflix dataset demonstrates skews in the sampling distribution for different films [16]. Thus, given this model our goal is to recover the weight vector w given such pairwise comparisons. We now discuss our method for computing the scores w_i.

2.2 Random walk approach to ranking

In our setting, we will assume that a_{ij} represents the fraction of times object j has been preferred to object i, for example the fraction of times chess player j has defeated player i. Given the notation above we have that $a_{ij} = (1/k)\sum_{l=1}^{k} Y_{ij}^l$. Consider a random walk on a weighted directed graph $G = ([n], E, A)$, where a pair $(i, j) \in E$ if and only if the pair has been compared. The weight edges are defined as the outcome of the comparisons $A_{ij} = a_{ij}/(a_{ij} + a_{ji})$ and $A_{ji} = a_{ji}/(a_{ij} + a_{ji})$. We let $A_{ij} = 0$ if the pair has not been compared. Note that by the Strong Law of Large Numbers, as the number $k \to \infty$ the quantity A_{ij} converges to $w_j/(w_i + w_j)$ almost surely.

A random walk can be represented by a time-independent transition matrix P, where $P_{ij} = \mathbb{P}(X_{t+1} = j | X_t = i)$. By definition, the entries of a transition matrix are non-negative and satisfy $\sum_j P_{ij} = 1$. One way to define a valid transition matrix of a random walk on G is to scale all the edge weights by $1/d_{\max}$, where we define $d_{\max}$ as the maximum out-degree of a node. This ensures that each row-sum is at most one. Finally, to ensure that each row-sum is exactly one, we add a self-loop to each node. More concretely,

$$P_{ij} \quad = \quad \begin{cases} \frac{1}{d_{\max}} A_{ij} & \text{if } i \neq j\ , \\ 1 - \frac{1}{d_{\max}} \sum_{k \neq i} A_{ik} & \text{if } i = j\ . \end{cases} \tag{2}$$

The choice to construct our random walk as above is not arbitrary. In an ideal setting with infinite samples ($k \to \infty$) the transition matrix P would define a reversible Markov chain. Recall that a Markov chain is reversible if it satisfies the *detailed balance equation*: there exists $v \in \mathbb{R}_+^n$ such that $v_i P_{ij} = v_j P_{ji}$ for all i, j; and in that case, $\pi \in \mathbb{R}_+^n$ defined as $\pi_i = v_i/(\sum_j v_j)$ is it's unique stationary distribution. In the ideal setting (say $k \to \infty$), we will have $P_{ij} \equiv \tilde{P}_{ij} = (1/d_{\max}) w_j/(w_i + w_j)$. That is, the random walk will move from state i to state j with probability equal to the chance that item j is preferred to item i. In such a setting, it is clear that $v = w$ satisfies the reversibility conditions. Therefore, under these ideal conditions it immediately follows that the vector $w/\sum_i w_i$ acts as a valid stationary distribution for the Markov chain defined by $\tilde{P}$, the ideal matrix. Hence, as long as the graph G is connected and at least one node has a self loop then we are guaranteed that our graph has a unique stationary distribution proportional to w. If the Markov chain is reversible then we may apply the spectral analysis of self-adjoint operators, which is crucial in the analysis when we repeatedly apply the operator $\tilde{P}$.

In our setting, the matrix P is a noisy version (due to finite sample error) of the ideal matrix $\tilde{P}$ discussed above. Therefore, it naturally suggests the following algorithm as a surrogate. We estimate the probability distribution obtained by applying matrix P repeated starting from any initial condition. Precisely, let $p_t(i) = \mathbb{P}(X_t = i)$ denote the distribution of the random walk at time t

with $p_0 = (p_0(i)) \in \mathbb{R}^n_+$ be an arbitrary starting distribution on $[n]$. Then,

$$p_{t+1}^T \quad = \quad p_t^T P \, . \tag{3}$$

Regardless of the starting distribution, when the transition matrix has a unique top eigenvalue, the random walk always converges to a unique distribution: the stationary distribution $\pi = \lim_{t\to\infty} p_t$. In linear algebra terms, this stationary distribution π is the top left eigenvector of P, which makes computing π a simple eigenvector computation. Formally, we state the algorithm, which assigns numerical scores to each node, which we shall call *Rank Centrality*:

Rank Centrality
Input: $G = ([n], E, A)$
Output: rank $\{\pi(i)\}_{i\in[n]}$
1: Compute the transition matrix P according to (2);
2: Compute the stationary distribution π.

The stationary distribution of the random walk is a fixed point of the following equation:

$$\pi(i) = \sum_j \pi(j) \frac{A_{ji}}{\sum_\ell A_{i\ell}} \, .$$

This suggests an alternative intuitive justification: an object receives a high rank if it has been preferred to other high ranking objects or if it has been preferred to many objects.

One key question remains: does P have a well defined stationary distribution? As discussed earlier, when G is connected, the idealized transition matrix $\tilde{P}$ has stationary distribution with desired properties. But due to noise, P may not be reversible and the arguments of ideal $\tilde{P}$ do not apply to our setting. Indeed, it is the finite sample error that governs the noise. Therefore, by analyzing the effect of this noise (and hence the finite samples), it is likely that we can obtain the error bound on the performance of the algorithm. As an important contribution of this work, we will show that even the iterations (cf. (3)) induced by P are *close enough* to those induced by $\tilde{P}$. Subsequently, we can guarantee that the iterative algorithm will converge to a solution that is close to the ideal stationary distribution.

3 Main Results

Our main result, Theorem 1, provides an upper bound on estimating the stationary distribution given the observation model presented above. The results demonstrate that even with random sampling we can estimate the underlying score with high probability with good accuracy. The bounds are presented as the rescaled Euclidean norm between our estimate π and the underlying stationary distribution $\tilde{P}$. This error metric provides us with a means to quantify the relative certainty in guessing if one item is preferred over another. Furthermore, producing such scores are ubiquitous [17] as they may also be used to calculate the desired rankings. After presenting our main theoretical result we will then provide simulations demonstrating the empirical performance of our algorithm in different contexts.

3.1 Error bound in stationary distribution recovery via Rank Centrality

The theorem below presents our main recovery theorem under the sampling assumptions described above. It is worth noting that while the result presented below is for the specific sampling model described above. The results can be extended to general graphs as long as the spectral gap of the corresponding Markov chain is well behaved. We will discuss the point further in the sequel.

Theorem 1. *Assume that, among n items, each pair is chosen with probability d/n and for each chosen pair we collect the outcomes of k comparisons according to the BTL model. Then, there exists positive universal constants C, C', and C'' such that when $d \geq C(\log n)^2$, and $k\,d \geq Cb^5 \log n$, the following bound on the error rate holds with probability at least $1 - C''/n^3$:*

$$\frac{\left\|\pi - \tilde{\pi}\right\|}{\|\tilde{\pi}\|} \quad \leq \quad C' b^3 \sqrt{\frac{\log n}{k\,d}} \, ,$$

where $\tilde{\pi}(i) = w_i / \sum_\ell w_\ell$ and $b \equiv \max_{i,j} w_i/w_j$.

Remarks. Some remarks are in order. First, the above result implies that as long as we choose $d = \Theta(\log^2 n)$ and $k = \omega(1)$ (i.e. large enough, say $k = \Theta(\log n)$), the error goes to 0 (with $k = \Theta(\log n)$, it goes down at rate $1/\sqrt{\log n}$) as n increases. Since we are sampling each of the $\binom{n}{2}$ pairs with probability d/n and then sampling them k times, we obtain $O(n \log^3 n)$ (with $k = \Theta(\log n)$) comparisons in total. Due to classical results on Erdos-Renyi graphs, the induced graph G is connected with high probability only when total number of pairs sampled scales as $\Omega(n \log n)$–we need at least those many comparisons. Thus, our result can be sub-optimal only up to $\log^2 n$ ($\log^{1+\epsilon} n$ if $k = \log^\epsilon n$).

Second, the b parameter should be treated as constant. It is the *dynamic* range in which we are trying to resolve the uncertainty between scores. If b were scaling with n, then it would be really easy to differentiate scores of items that are at the two opposite end of the dynamic range; in which case one could focus on differentiating scores of items that have their parameter values near-by. Therefore, the interesting and challenging regime is where b is constant and not scaling.

Finally, observe the interesting consequence that under the conditions on d, since the induced distribution π is close to $\tilde{\pi}$, it implies connectivity of G. Thus, the analysis of our algorithm provides an alternative proof of connectivity in an Erdos-Renyi graph (of course, by using heavy machinery!).

3.2 Experimental Results

Under the BTL model, define an error metric of a estimated ordering σ as the weighted sum of pairs (i, j) whose ordering is incorrect:

$$D_w(\sigma) \quad = \quad \Big\{ \frac{1}{2n\|w\|^2} \sum_{i<j} (w_i - w_j)^2 \, \mathbb{I}\big((w_i - w_j)(\sigma_i - \sigma_j) > 0\big) \Big\}^{1/2},$$

where $\mathbb{I}(\cdot)$ is an indicator function. This is a more natural error metric compared to the Kemeny distance, which is an unweighted version of the above sum, since $D_w(\cdot)$ is less sensitive to errors between pairs with similar weights. Further, assuming without loss of generality that w is normalized such that $\sum_i w_i = 1$, the next lemma connects the error in $D_w(\cdot)$ to the bound provided in Theorem 1. Hence, the same upper bound holds for D_w error. Due to space constraints, we refer to a longer version of this paper for a proof of this lemma.

Lemma 3.1. *Let σ be an ordering of n items induced by a scoring π. Then, $D_w(\sigma) \leq \|w - \pi\|/\|w\|$.*

For a fixed $n = 400$ and a fixed $b = 10$, Figure. 1 illustrates how the error scales with two problem parameters: varying the number of comparisons per pair with fixed $d = 10 \log n$ (left) and varying the sampling probability with fixed $k = 32$ (right). The ML estimator directly maximizes the likelihood assuming the BTL model [18]. If we reparameterize the problem so that $\theta_i = \log(w_i)$ then we obtain our estimates $\widehat{\theta}$ by solving the convex program

$$\widehat{\theta} \in \arg\min_{\theta} \sum_{(i,j) \in E} \sum_{l=1}^{k} \log(1 + \exp(\theta_j - \theta_i)) - Y_{ij}^l (\theta_j - \theta_i),$$

which is pair-wise logistic regression. This choice is optimal in the asymptotic setting, however for fixed-samples there do not exist theoretical guarantees for recovering the transformed scores θ_i. The method Count Wins scores an item by counting the number of wins divided by the total number of comparisons [1]. Ratio Matrix assigns scores according to the top eigenvector of a matrix, whose (i, j)-th entry is a_{ij}/a_{ji} [19]. As we see in Figure 1, the error achieved by our Random Walk approach is comparable to that of ML estimator, and vanishes at the rate of $1/\sqrt{k}$ as predicted by our main result. Interestingly, for fixed d, both the Count Wins and Ratio Matrix algorithms have strictly positive error even if we take $k \to \infty$. The figure on the right illustrates that the error scales as $1/\sqrt{d}$ as expected from our main result.

4 Proofs

We may now present the proof of Theorem 1. As previously alluded to the statement of Theorem 1 can be made more general. The result that we presented is a specific instance of a more general

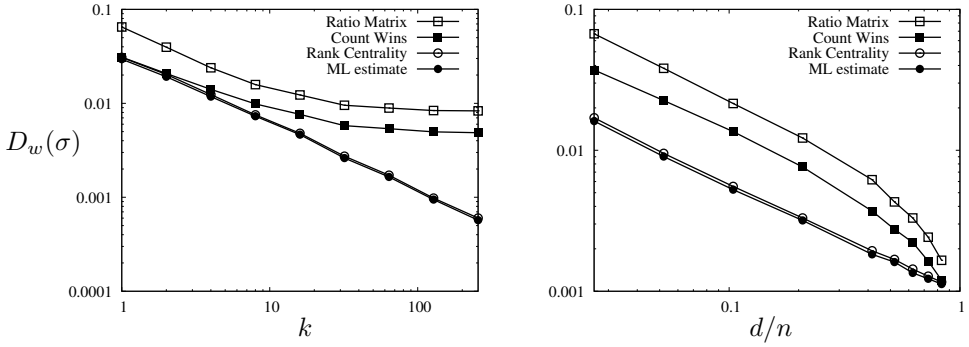

Figure 1: Average error $D_w(\sigma)$ of orderings from four rank aggregation algorithms, averaged over 20 instances. In the figure on the right we assume that d and n are fixed while we increase k. The figure on the right takes $k = 32$ fixed and lets d increase.

lemma that we state below, which shows that our algorithm enjoys convergence properties that result in useful upper bounds. The lemma is made general and uses standard techniques of spectral theory. The main difficulty arises in establishing that the Markov chain P satisfies certain properties that we will discuss below. In order to show that these properties hold we must rely on the specific model that allows us to ultimately establish error bounds that hold with high probability. In what follows we present the lemma and omit the proofs of certain technical details to the longer version of the paper.

4.1 Algorithm convergence

In this section, we characterize the error rate achieved by our ranking algorithm. Given the random Markov chain P, where the randomness comes from the outcome of the comparisons, we will show that it does not deviate too much from its expectation $\tilde{P}$, where we recall is defined as

$$\tilde{P}_{ij} \quad = \quad \begin{cases} \frac{1}{d_{\max}} \frac{w_j}{w_i + w_j} & \text{if } i \neq j \ , \\ 1 - \frac{1}{d_{\max}} \sum_{\ell \neq i} \frac{w_\ell}{w_i + w_\ell} & \text{if } i = j \end{cases}$$

for all $(i, j) \in E$ and $\tilde{P}_{ij} = 0$ otherwise.

Recall from the discussion following equation (2) that the transition matrix P used in our ranking algorithm has been carefully chosen such that the corresponding expected transition matrix $\tilde{P}$ has two important properties. First, the stationary distribution of $\tilde{P}$, which we denote with $\tilde{\pi}$ is proportional to the weight vectors w. Furthermore, when the graph is connected and has self loops (which at least one exists), this Markov chain is irreducible and aperiodic so that the stationary distribution is unique. The next important property of $\tilde{P}$ is that it is reversible– $\tilde{\pi}(i)\tilde{P}_{ij} = \tilde{\pi}(j)\tilde{P}_{ji}$. This observation implies that the operator $\tilde{P}$ is symmetric in an appropriate defined inner product space. The symmetry of the operator $\tilde{P}$ will be crucial in applying ideas from spectral analysis to prove our main results.

Let Δ denote the fluctuation of the transition matrix around its mean, such that $\Delta \equiv P - \tilde{P}$. The following lemma bounds the deviation of the Markov chain after t steps in terms of two important quantities: the spectral radius of the fluctuation $\|\Delta\|_2$ and the spectral gap $1 - \lambda_{\max}(\tilde{P})$, where

$$\lambda_{\max}(\tilde{P}) \quad \equiv \quad \max\{\lambda_2(\tilde{P}), -\lambda_n(\tilde{P})\} \ .$$

Lemma 4.1. *For any Markov chain $P = \tilde{P} + \Delta$ with a reversible Markov chain $\tilde{P}$, let p_t be the distribution of the Markov chain P when started with initial distribution p_0. Then,*

$$\frac{\left\|p_t - \tilde{\pi}\right\|}{\|\tilde{\pi}\|} \quad \leq \quad \rho^t \frac{\|p_0 - \tilde{\pi}\|}{\|\tilde{\pi}\|} \sqrt{\frac{\tilde{\pi}_{\max}}{\tilde{\pi}_{\min}}} + \frac{1}{1-\rho} \|\Delta\|_2 \frac{\tilde{\pi}_{\max}}{\tilde{\pi}_{\min}} \ . \tag{4}$$

where $\tilde{\pi}$ is the stationary distribution of $\tilde{P}$, $\tilde{\pi}_{\min} = \min_i \tilde{\pi}(i)$, $\tilde{\pi}_{\max} = \max_i \tilde{\pi}(i)$, and $\rho = \lambda_{\max}(\tilde{P}) + \|\Delta\|_2 \sqrt{\tilde{\pi}_{\max}/\tilde{\pi}_{\min}}$.

The above result provides a general mechanism for establishing error bounds between an estimated stationary distribution π and the desired stationary distribution $\tilde{\pi}$. It is worth noting that the result only requires control on the quantities $\|\Delta\|_2$ and $1-\rho$. We may now state two technical lemmas that provide control on the quantities $\|\Delta\|_2$ and $1-\rho$, respectively.

Lemma 4.2. *Under the assumptions of Theorem 1, we have that the error matrix $\Delta = P - \tilde{P}$ satisfies*

$$\|\Delta\|_2 \quad \leq \quad C\sqrt{\frac{\log n}{kd}}$$

for some positive universal constant C with probability at least $1 - 3n^{-4}$

The next lemma provides our desired bound on $1-\rho$.

Lemma 4.3. *Under the assumptions of Theorem 1, the spectral radius satisfies*

$$1 - \rho \geq C'/b^2$$

with probability at least $1 - n^{-c}$, for some positive universal constant C' and c. The constant c can be made as large as we want by increasing the constant C in $d \geq C \log n$.

With the above results in hand we may now proceed with the proof of Theorem 1.

When there is a positive spectral gap $\rho < 1$ the first term in (4) vanishes as t grows. The rest of the first term is bounded and independent of t. Formally, we have

$$\tilde{\pi}_{\max}/\tilde{\pi}_{\min} \leq b \ , \quad \|\tilde{\pi}\| \geq 1/\sqrt{n} \ , \quad \text{and } \|p_0 - \tilde{\pi}\| \leq 2 \ ,$$

by the assumption that $\max_{i,j} w_i/w_j \leq b$ and the fact that $\tilde{\pi}(i) = w_i/(\sum_j w_j)$. Hence, the error between the distribution at the t^{th} iteration p^t and the true stationary distribution $\tilde{\pi}$ is dominated by the second term in equation (4). Therefore, in order to finish the proof of Theorem 1 we require bounds on $\|\Delta\|_2$ and $1-\rho$.

We recall that by Lemma 4.2 we have $\|\Delta\|_2 \leq C\sqrt{\log n/(kd)}$ and from Lemma 4.3 that there is a positive spectral gap $1 - \rho \geq C'/b^2$ for some numerical constants C and C'. Given these observations the dominant second term in equation (4) is bounded by

$$\lim_{t\to\infty} \frac{\left\|p_t - \tilde{\pi}\right\|}{\|\tilde{\pi}\|} \quad \leq \quad C\, b^3 \sqrt{\frac{\log n}{kd}} \ .$$

This finishes the proof of Theorem 1.

5 Discussion

In this paper, we developed a novel iterative rank aggregation algorithm for discovering scores of objects given pairwise comparisons. The algorithm has a natural *random walk* interpretation over the graph of objects with edges present between two objects if they are compared; the scores turn out to be the stationary probability of this random walk. In lieu of recent works on network centrality which are graph score functions primarily based on random walks, we call this algorithm *Rank Centrality*. The algorithm is model independent.

We also established the efficacy of the algorithm by analyzing its performance when data is generated as per the popular Bradley-Terry-Luce (BTL) model. We have obtained an analytic bound on the finite sample error rates between the scores assumed by the BTL model and those estimated by our algorithm. As shown, these lead to order-optimal dependence on the number of samples required to learn the scores well by our algorithm. The experimental evaluation show that our (model independent) algorithm performs as well as the Maximum Likelihood Estimator of the BTL model and outperforms other known competitors included the recently proposed algorithm by Ammar and Shah [1]. Given the simplicity of the algorithm, analytic guarantees and wide utility of the problem of rank aggregation, we believe that this algorithm will be of great practical value.

References

[1] A. Ammar and D. Shah. Communication, control, and computing (allerton), 2011, 49th annual allerton conference on. pages 776–783, September 2011.

[2] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[3] R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980 –2998, june 2010.

[4] S. Negahban and M. J. Wainwright. Restricted strong convexity and (weighted) matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 2012. To appear; posted at http://arxiv.org/abs/1009.2118.

[5] M. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. l'Imprimerie Royale, 1785.

[6] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, 1963.

[7] M. Braverman and E. Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 268–276. Society for Industrial and Applied Mathematics, 2008.

[8] 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, 1955.

[9] D. R. Luce. *Individual Choice Behavior*. Wiley, New York, 1959.

[10] D. McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1973.

[11] K. T. Talluri and G. Van Ryzin. *The Theory and Practice of Revenue Management*. springer, 2005.

[12] M. Adler, P. Gemmell, M. Harchol-Balter, R. M. Karp, and C. Kenyon. Selection in the presence of noise: the design of playoff systems. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, SODA '94, pages 564–572. Society for Industrial and Applied Mathematics, 1994.

[13] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[14] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[16] R. Salakhutdinov and N. Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. Technical Report abs/1002.2780v1, Toyota Institute of Technology, 2010.

[17] J. C. Duchi, L. Mackey, and M. I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the ICML Conference*, Haifa, Israel, June 2010.

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

[19] T. L. Saaty. Decision-making with the ahp: Why is the principal eigenvector necessary. *European Journal of Operational Research*, 145:pp. 85–91, 2003.