A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data

Arun Rajkumar
Shivani Agarwal
Indian Institute of Science, Bangalore 560012, INDIA

Abstract
There has been much interest recently in the problem of rank aggregation from pairwise data. A natural question that arises is: under what sorts of statistical assumptions do various rank aggregation algorithms converge to an 'optimal' ranking? In this paper, we consider this question in a natural setting where pairwise comparisons are drawn randomly and independently from some underlying probability distribution. We first show that, under a 'time-reversibility' or Bradley-Terry-Luce (BTL) condition on the distribution, the rank centrality (PageRank) and least squares (HodgeRank) algorithms both converge to an optimal ranking. Next, we show that a matrix version of the Borda count algorithm, and more surprisingly, an algorithm which performs maximum likelihood estimation under a BTL assumption, both converge to an optimal ranking under a 'low-noise' condition that is strictly more general than BTL. Finally, we propose a new SVM-based algorithm for rank aggregation from pairwise data, and show that this converges to an optimal ranking under an even more general condition that we term 'generalized low-noise'. In all cases, we obtain explicit sample complexity bounds.

1. Introduction
Rank aggregation is a classical problem that has been studied in several contexts, starting with social choice theory in 18th century France (Borda, 1781; Condorcet, 1785), and more recently, in computer science, statistics, linear algebra, and optimization, with a variety of different applications, and with different forms of both input rankings and desired aggregated rankings being considered (Dwork et al., 2001; Hochbaum, 2006; Meila et al., 2007; Ailon et al., 2008; Klementiev et al., 2008; Jagabathula & Shah, 2008; Guiver & Snelson, 2009; Ailon, 2010; Qin et al., 2010; Jiang et al., 2011; Gleich & Lim, 2011; Volkovs & Zemel, 2012; Negahban et al., 2012; Soufiani et al., 2012; Osting et al., 2013). A prominent setting that has gained interest in recent years is that of rank aggregation from pairwise data, where there is a set of $n$ items to rank (such as movies or webpages), and one is given outcomes of various pairwise comparisons among these items (such as pairwise movie or webpage preferences of users); the goal is to aggregate these pairwise comparisons into a global ranking over the items. Various algorithms have been studied for this problem, including maximum likelihood under a Bradley-Terry-Luce (BTL) model assumption, rank centrality (PageRank/MC3) (Negahban et al., 2012; Dwork et al., 2001), least squares (HodgeRank) (Jiang et al., 2011), and a pairwise variant of Borda count (Borda, 1781; Jiang et al., 2011) among others.

In this paper, we consider statistical convergence properties of these rank aggregation algorithms under a natural statistical model, under which pairwise comparisons are drawn i.i.d. from some fixed but unknown probability distribution. An 'optimal' ranking is then one which minimizes the probability of disagreement with a random pairwise comparison drawn from this distribution. We consider three conditions of increasing generality on the distribution: a BTL condition, a 'low-noise' (LN) condition similar to a condition considered by (Duchi et al., 2010) in a different setting, and a 'generalized low-noise' (GLN) condition. We show that the rank centrality and least squares algorithms both converge (in probability) to an optimal ranking under the BTL condition, and that the Borda count and BTL-ML algorithms converge to an optimal ranking under the LN condition; we then propose a new SVM based rank aggregation algorithm which we show converges to an optimal ranking under the GLN condition. In all cases, we obtain explicit sample complexity bounds.
Related Work. The work most closely related to ours is that of Negahban et al. (Negahban et al., 2012), who analyzed convergence of the rank centrality algorithm under a statistical model where a fixed set of item pairs is repeatedly compared a fixed number of times, and the outcomes of the comparisons are determined by a BTL model. Our statistical model, where pairs to be compared are drawn randomly, is more natural in many applications (e.g. movie rankings). Our analysis of the rank centrality algorithm builds on that of (Negahban et al., 2012). However, we cannot use the standard matrix concentration tools used in (Negahban et al., 2012) since the comparison matrix in our case does not contain independent entries; instead, we use a McDiarmid-like concentration result of Kutin (Kutin, 2002) to analyze each entry separately. We point out our setting differs from the active learning settings of (Ailon, 2011; Jamieson & Nowak, 2011), where the goal is to recover a true permutation on n items by actively querying specific pairs; in our setting, the pairs are randomly sampled. Similarly, our setting differs from that of (Wauthier et al., 2013), where each pair of items can be compared at most once; in the rank aggregation setting we consider, it is common to have the same pair of items compared several times (with possibly different random outcomes). Our setting also differs from standard learning-to-rank problems involving pairwise preferences, where algorithms such as RankSVM or RankBoost are typically applied (Herbrich et al., 2000; Joachims, 2002; Freund et al., 2003), as there are no feature vectors in our setting; instead we simply have a finite number of objects with identifiers 1, . . . , n. Finally, our setting also differs from the subset ranking settings studied recently in machine learning and information retrieval (Cossock & Zhang, 2008; Duchi et al., 2010; Liu, 2011), where one ranks documents for various queries.

Summary and Organization. Figure 1 summarizes our results. Section 2 gives preliminaries. Section 3 summarizes various useful properties of the comparison matrix in our setting. Sections 4–6 consider conditions of increasing generality on the probability distribution generating pairwise comparisons, and analyze convergence properties of various rank aggregation algorithms under these conditions. Section 7 gives our experimental results. All proofs can be found in the supplementary material.

2. Preliminaries, Notation, and Background

Setup. Let [n] = {1, . . . , n} denote a set of n items to rank. Let X = {(i, j) : i, j ∈ [n], i < j}. The learner is given a training sample S = ((i1, j1, y1), . . . , (im, jm, ym)) ∈ (X × {0, 1})m, where for each k ∈ [m], (ik, jk) ∈ X denotes the k-th pair of items compared, and yk ∈ {0, 1} denotes the outcome of the comparison; we adopt the convention that yk is 1 if item jk is ranked higher than item ik, and 0 otherwise. Given S, the goal of the learner is to produce a ranking or permutation of the n items, σ ∈ Sn.

We assume a probability distribution µ on X from which item pairs are sampled. For each i < j, we denote by µij the probability of the pair (i, j) under µ; with some abuse of notation, we also denote µji = µij ∀ i < j. We also assume a set of conditional label probabilities from which labels are drawn. Specifically, for each i < j, we denote by Pij ∈ [0, 1] the probability that item j will be ranked higher than item i when items i and j are compared; we represent this as a pairwise preference matrix P ∈ [0, 1]m×n with Pij = 1 − Pji for i < j and Pii = 0. The training sample S = ((i1, j1, y1), . . . , (im, jm, ym)) ∈ (X × {0, 1})m is then assumed to be drawn randomly according to S ∼ (µ, P)m, i.e. the item pairs (ik, jk) are drawn randomly and independently according to µ, and conditioned on these, the labels are drawn as yk ∼ Bernoulli(Pik, jk).

Given a distribution (µ, P) as above, define the expected pairwise disagreement error of a permutation σ ∈ Sn as

\[ \text{er}_{µ, P}^{PD}[σ] = \sum_{i \neq j} µ_{ij} P_{ij} \mathbb{1}(σ(i) < σ(j)), \]

where \( \mathbb{1}(·) \) is 1 if its argument is true and 0 otherwise; this is the probability that σ does not agree with a pairwise comparison drawn randomly according to (µ, P). An ‘optimal’ permutation is then any permutation σ∗ satisfying

\[ σ∗ ∈ \text{argmin}_{σ ∈ S_n} \text{er}_{µ, P}^{PD}[σ]. \]

Clearly, an ideal algorithm would recover (with high probability, for a large enough sample) such an optimal permutation. In what follows, we will consider various conditions on (µ, P), including a ‘time-reversibility’ or BTL condition, a ‘low-noise’ condition, and a ‘generalized low-noise’ condition, and will analyze convergence properties of various rank aggregation algorithms under these conditions.

| RANK CENTRALITY (NEGABHAN ET AL., 2012) | BTL | LN | GLN |
| LEAST SQUARES (HODGERANK) (HODGKIN, 2011) | ✓ | × | × |
| BORDA COUNT (BORDA, 1781; JIANG ET AL., 2011) | ✓ | ✓ | ✓ |
| BTL-ML (CLASSICAL) | ✓ | ✓ | ✓ |
| SVM-RANKAGGREGATION (THIS PAPER) | ✓ | ✓ | ✓ |

Figure 1. We consider three increasingly general conditions on the distribution generating pairwise comparisons: BTL, LN, and GLN. The table summarizes our results on convergence of various rank aggregation algorithms to an optimal ranking under these conditions.

\[ \text{er}_{µ, P}^{PD}[σ] = \sum_{i \neq j} µ_{ij} P_{ij} \mathbb{1}(σ(i) < σ(j)), \]

\[ σ∗ ∈ \text{argmin}_{σ ∈ S_n} \text{er}_{µ, P}^{PD}[σ]. \]
All the algorithms we analyze take as input an empirical pairwise comparison matrix \( \hat{P} \in [0, 1]^{n \times n} \), which in our case is constructed from \( S \) as follows:

\[
\hat{P}_{ij} = \begin{cases} 
N_{ij}^{(1)}/N_{ij} & \text{if } i < j \text{ and } N_{ij} > 0 \\
1 - (N_{ji}^{(1)}/N_{ji}) & \text{if } i > j \text{ and } N_{ji} > 0 \\
0 & \text{otherwise};
\end{cases}
\]

where

\[
N_{ij} = \sum_{k=1}^{m} 1(ik = i, jk = j) ;
\]

\[
N_{ij}^{(1)} = \sum_{k=1}^{m} 1(ik = i, jk = j, yk = 1) .
\]

Note that the empirical comparison matrix \( \hat{P} \) differs from the true pairwise preference matrix \( P \) in an important aspect: while \( \hat{P} \) satisfies \( P_{ij} + P_{ji} = 1 \) for all \( i < j \), in the case of \( P \), if a particular pair \( i < j \) is not observed in \( S \), we can have \( \hat{P}_{ij} = \hat{P}_{ji} = 0 \). To reinforce this distinction, we will use the terms pairwise preference matrix for \( P \) and pairwise comparison matrix for \( \hat{P} \) throughout.

The structure of \( \hat{P} \) is critical in our analysis: unlike (Negahban et al., 2012), where the empirical comparison matrix was constructed by fixing a priori some subset of pairs \((i,j)\) to be compared and then repeatedly comparing each such pair a fixed number of times, which led to the entries of the matrix being independent, in our case, the entries of \( \hat{P} \) are not independent (note that some pair of items is sampled many times, other item pairs will be less frequent in \( S \)); therefore we cannot apply the matrix concentration tools used in (Negahban et al., 2012). Instead, we will show the elements of \( \hat{P} \) satisfy a bounded difference property with high probability, allowing us to analyze \( \hat{P} \) using Kutin’s extension of McDiarmid’s inequality (Kutin, 2002). This and other properties of \( \hat{P} \), proved in Section 3, will then be used to analyze various algorithms in Sections 4–6.

**Notation.** We will find it convenient to define

\[
\mu_{\text{min}} = \min_{i < j} \mu_{ij}, 
\]

\[
B(\mu_{\text{min}}) = 3\left(\frac{12}{\mu_{\text{min}}} + 3\right) \ln \left(\frac{12}{\mu_{\text{min}}} + 3\right). 
\]

Our results below will assume \( \mu_{\text{min}} > 0 \). We will use capital boldface letters such as \( \mathbf{P} \), \( \mathbf{Q} \) for matrices and lower case boldface letters such as \( \mathbf{f} \), \( \pi \) for vectors. For \( \mathbf{f} \in \mathbb{R}^n \), we will denote by \( \|\mathbf{f}\|_1 = \sum_{i=1}^{n} |f_i| \), \( \|\mathbf{f}\|_2 = \left(\sum_{i=1}^{n} f_i^2\right)^{1/2} \), and \( \|\mathbf{f}\|_{\infty} = \max_i |f_i| \) the standard \( L_1 \), \( L_2 \) and \( L_{\infty} \) norms. Also, for \( \mathbf{f} \in \mathbb{R}^n \), we will denote by argsort(\( \mathbf{f} \)) the set of permutations that order items \( i \in [n] \) in decreasing order of scores \( f_i \), breaking ties arbitrarily:

\[
\text{argsort}(\mathbf{f}) = \{ \sigma \in S_n : f_{i > j} > f_{j > j} \implies \sigma(i) < \sigma(j) \}. 
\]

**Background Results.** The following definition of strongly difference-bounded random variables and concentration result for such random variables, both due to Kutin (Kutin, 2002), will be used in our analysis of the empirical comparison matrix \( P \) in Section 3.

**Definition 1** (Strong difference-boundedness (Kutin, 2002)). Let \( X = (X_1, \ldots, X_m) \) be a vector of independent random variables with \( X_i \) taking values in some set \( A_i \), and let \( A = A_1 \times \cdots \times A_m \). Let \( \phi : A \to \mathbb{R} \) be any function. Let \( b, c > 0 \) and \( \delta \in (0, 1] \). The random variable \( \phi(X) \) is said to be strongly difference-bounded by \((b, c, \delta)\) if \( \forall B \subset A \) with \( P(X \in B) \leq \delta \) such that for each \( k \in [m] \),

\[
\sup_{x \not\in B, x' \in A_k} \left| \phi(x) - \phi(x_1, \ldots, x'_k, \ldots, x_m) \right| \leq c \]

\[
\sup_{x \in A, x' \in A_k} \left| \phi(x) - \phi(x_1, \ldots, x'_k, \ldots, x_m) \right| \leq b.
\]

**Theorem 2** ((Kutin, 2002)). Let \( X = (X_1, \ldots, X_m) \) be a vector of independent random variables with \( X_i \) taking values in some set \( A_i \), and let \( A = A_1 \times \cdots \times A_m \). Let \( \phi : A \to \mathbb{R} \) be any function such that \( \phi(X) \) is strongly difference-bounded by \((b, c, \delta)\) then

\[
P\left( \left| \phi(X) - \mathbb{E}[\phi(X)] \right| \geq \epsilon \right) \leq \exp\left(-\frac{m\epsilon^2}{8\lambda^2}\right).
\]

**3. Properties of Comparison Matrix \( \hat{P} \)**

The following lemma summarizes various useful properties of the empirical comparison matrix \( \hat{P} \) that are used in our proofs. In particular, a key property is that for large enough \( m \), the elements of \( \hat{P} \) are strongly difference-bounded, allowing us to obtain concentration results for them.

**Lemma 3.** Let \((\mu, P)\) be such that \( \mu_{\text{min}} > 0 \). Let \( S \sim \left(\mu, P\right)^m \), and let \( \hat{P} \) be constructed from \( S \) as in Eq. (3).

1. Let \( i \neq j \). If \( m \geq \frac{4}{\mu_{\text{min}}} \), then \( \hat{P}_{ij} \) is strongly difference-bounded by \( \left(1, \frac{2}{\mu_{\text{min}}} \exp\left(-\frac{m\epsilon^2}{2\lambda^2}\right)\right) \).

2. Let \( i \neq j \). Let \( 0 < \epsilon < 2\sqrt{2} \). If \( m \geq B(\mu_{\text{min}}) \), then

\[
P\left( \left| \hat{P}_{ij} - \mathbb{E}[\hat{P}_{ij}] \right| \geq \epsilon \right) \leq 4 \exp\left(-\frac{m\epsilon^2}{32}\right).
\]

3. Let \( i \neq j \). Let \( \epsilon > 0 \). If \( m \geq \frac{1}{\mu_{\text{min}}} \ln \left(\frac{1}{\epsilon}\right) \), then \( \left| \mathbb{E}[\hat{P}_{ij}] - P_{ij} \right| \leq \epsilon \).

4. Let \( i \neq j \). Let \( 0 < \epsilon < 4\sqrt{2} \). If \( m \geq \max\left( B(\mu_{\text{min}}), \frac{1}{\mu_{\text{min}}} \ln \left(\frac{2}{\epsilon}\right) \right) \), then

\[
P\left( \left| \hat{P}_{ij} - P_{ij} \right| \geq \epsilon \right) \leq 4 \exp\left(-\frac{m\epsilon^2}{128}\right).
\]

5. Let \( P_{ij} \in (0, 1) \forall i \neq j \), and let \( P_{\min} = \min_{i \neq j} P_{ij} \). Let \( \delta \in (0, 1] \). If \( m \geq \frac{1}{\mu_{\text{min}}} \ln \left(\frac{n(n-1)}{6\delta}\right) \), then with probability at least \( 1 - \delta \), \( \hat{P}_{ij} > 0 \forall i \neq j \).

The proof of Part 1 makes use of Hoeffding’s inequality and involves a somewhat detailed, careful case-by-case analysis. Part 2 then follows from Part 1 and Theorem 2. Part 3 follows by observing \( \mathbb{E}[\hat{P}_{ij}] = P_{ij}(1 - (1 - \mu_{ij})^m) \). Part 4 follows from Parts 2 and 3. Part 5 is straightforward. Details can be found in the supplementary material.
4. Time-Reversibility/BTL Condition

We first consider the following ‘time-reversibility’ and BTL conditions on the preference matrix $P$:

**Definition 4 (Time-reversibility condition).** We say the pairwise preference matrix $P \in [0, 1]^{n \times n}$ satisfies the time-reversibility condition if the Markov chain $Q$ given by

$$Q_{ij} = \begin{cases} \frac{1}{n} P_{ij} & \text{if } i \neq j \\ 1 - \frac{1}{n} \sum_{k \neq i} P_{ik} & \text{if } i = j \end{cases}$$

is time-reversible, i.e., if $Q$ is irreducible and aperiodic and the stationary probability vector $\pi$ of $Q$ satisfies $\pi_i Q_{ij} = \pi_j Q_{ji}$ $\forall i, j \in [n]$.

**Definition 5 (Bradley-Terry-Luce (BTL) condition).** We say the pairwise preference matrix $P \in [0, 1]^{n \times n}$ satisfies the Bradley-Terry-Luce (BTL) condition if it corresponds to a BTL model, i.e., if $\exists w \in \mathbb{R}_+^n$ with $w_i > 0$ $\forall i$ such that $P_{ij} = w_j / (w_i + w_j)$ $\forall i \neq j$.

Clearly, if $P$ satisfies the time-reversibility condition with $Q$ and $\pi$ as above, then $\forall i \neq j$, $P_{ji} > P_{ij} \implies \pi_j > \pi_i$, and therefore any permutation that ranks items $i \in [n]$ in decreasing order of scores $\pi_i$ is an optimal permutation w.r.t. the pairwise disagreement error (see Eq. (1)). Similarly, if $P$ corresponds to a BTL model with parameter vector $w$ as above, then any permutation that ranks items according to decreasing order of scores $w_i$ is an optimal permutation. The following lemma shows that the time-reversibility and BTL conditions are in fact equivalent:

**Lemma 6.** A preference matrix $P \in [0, 1]^{n \times n}$ satisfies the time-reversibility condition if and only if it satisfies the BTL condition.

Note that if $P$ satisfies the time-reversibility condition, then by the above result, $P_{ij} \in (0, 1)$ $\forall i \neq j$.

4.1. Convergence of Rank Centrality Algorithm

We start by analyzing convergence behavior of the rank centrality algorithm (Dwork et al., 2001; Negahban et al., 2012) (Algorithm 1)\(^1\) in our setting under the above time-reversibility/BTL condition. In particular, we first show the following result, which establishes convergence of the score vector $\hat{\pi}$ produced by the rank centrality algorithm to $\pi$, the stationary vector of the matrix $Q$ defined in Eq. (6) (in $L_2$ norm):

\(^1\)Note that the rank centrality algorithm as presented here differs slightly from (Negahban et al., 2012) in two aspects: rather than divide elements of $\hat{P}$ by the maximum degree, which in our case depends on the sample $S$, we divide by $n$ in constructing $\hat{Q}$; similarly, since in our case the graph defining the Markov chain $\hat{Q}$ depends on $S$ and may not be strongly connected, we allow for the possibility of producing a default vector $\hat{\pi} = 0$ in this case. Also, while (Negahban et al., 2012) are interested in the score vector $\hat{\pi}$, we are interested in the ordering $\hat{\sigma}$ produced by $\hat{\pi}$.

**Algorithm 1 Rank Centrality (PageRank/MC3) (Negahban et al., 2012; Dwork et al., 2001)**

**Input:** Empirical comparison matrix $\hat{P} \in [0, 1]^{n \times n}$

Construct an empirical Markov chain with transition probability matrix $Q$ as follows:

$$Q_{ij} = \begin{cases} \frac{1}{n} \hat{P}_{ij} & \text{if } i \neq j \\ 1 - \frac{1}{n} \sum_{k \neq i} \hat{P}_{ik} & \text{if } i = j \end{cases}$$

If $Q$ defines an irreducible, aperiodic Markov chain, then compute $\hat{\pi}$, the stationary probability vector of $Q$

**Output:** Permutation $\hat{\sigma} \in \text{argsort}(\hat{\pi})$

**Theorem 7.** Let $(\mu, P)$ be such that $\mu_{\min} > 0$ and $P$ satisfies the BTL condition. Let $Q$ be defined as in Eq. (6), and let $\pi$ be the stationary probability vector of $Q$. Let $P_{\min} = \min_{i \neq j} P_{ij}$, $\pi_{\max} = \max_i \pi_i$, and $\pi_{\min} = \min_i \pi_i$. Let $0 < \epsilon \leq 1$ and $\delta \in (0, 1)$. If

$$m \geq \max \left( \frac{9216 n}{\epsilon^2 \mu_{\min}^2 \pi_{\min}^2} \left( \frac{\pi_{\max}}{\pi_{\min}} \right)^3 \ln \left( \frac{16n^2}{\delta} \right), B(\mu_{\min}) \right),$$

then with probability at least $1 - \delta$ (over the random draw of $S \sim (\mu, P)^m$ from which $\hat{P}$ is constructed), the score vector $\hat{\pi}$ produced by the rank centrality algorithm satisfies

$$||\hat{\pi} - \pi||_2 \leq \epsilon.$$
4.2. Convergence of Least Squares Algorithm

Next, we analyze convergence of the least squares (HodgeRank) algorithm (Jiang et al., 2011) (Algorithm 2) in our framework, again under the above time-reversibility/BTL condition on the preference matrix \( \mathbf{P} \). As discussed in (Jiang et al., 2011), the pairwise comparison matrix \( \hat{\mathbf{P}} \) is converted to a skew-symmetric matrix \( \hat{\mathbf{Y}} \) via a log-odds ratio transform before applying least squares (see Algorithm 2). Similarly, given a pairwise preference matrix \( \mathbf{P} \in [0, 1]^{n \times n} \), we can define a skew-symmetric matrix \( \mathbf{Y} \in \mathbb{R}^{n \times n} \) as

\[
Y_{ij} = \begin{cases} 
\ln \left( \frac{P_{ij}}{P_{ji}} \right) & \text{if } i \neq j \text{ and } P_{ij} \in (0, 1) \\
0 & \text{otherwise.}
\end{cases}
\]  

(7)

Let \( E = \{(i, j) \in \mathcal{X} : P_{ij} \neq 0 \text{ or } P_{ji} \neq 0\} \), and let \( \mathbf{f}^* \in \arg \min_{\mathbf{f} \in \mathbb{R}^n} \sum_{(i,j) \in E} \left( (f_i - f_j - Y_{ij})^2 \right) \). Clearly, if \( P_{ij} = 1 - P_{ji} \forall i \neq j \), we have \( E = \mathcal{X} \). In this case, as discussed in (Jiang et al., 2011), the (minimum norm) solution to the above optimization problem is given by

\[
f_i^* = -\frac{1}{n} \sum_{k=1}^{n} Y_{ik}.
\]  

(8)

The following lemma shows that if \( \mathbf{P} \) satisfies the time-reversibility/BTL condition, then ranking items according to decreasing order of scores \( f_i^* \) as above yields an optimal ranking w.r.t. the pairwise disagreement error:

**Lemma 9.** Let \((\mu, \mathbf{P})\) be such that \( \mathbf{P} \) satisfies the BTL condition. Let \( \mathbf{f}^* \in \mathbb{R}^n \) be defined as in Eq. (8). Then \( \text{argsort}(\mathbf{f}^*) \subseteq \arg \min_\sigma \mathbb{E}_\mathbf{P} \mathbb{E}_\mathbf{P} \mathbf{P} \mathbf{P} [\sigma] \).

The following is our main result regarding convergence of the least squares algorithm. Note that it establishes convergence of the score vector \( \hat{\mathbf{f}} \) produced by the least squares algorithm to \( \mathbf{f}^* \) (in \( L_\infty \) norm) under any \( \mathbf{P} \) satisfying \( P_{ij} \in (0, 1) \forall i \neq j \); however the optimality of permutations obtained from \( \mathbf{f}^* \) w.r.t. pairwise disagreement is guaranteed only when \( \mathbf{P} \) satisfies the time-reversibility/BTL condition.

**Theorem 10.** Let \((\mu, \mathbf{P})\) be such that \( \mu_{\min} > 0 \) and \( P_{ij} \in (0, 1) \forall i \neq j \). Let \( \mathbf{Y} \in \mathbb{R}^{n \times n} \) and \( \mathbf{f}^* \in \mathbb{R}^n \) be defined as in Eqs. (7) and (8). Let \( \mu_{\min} = \min_i P_{ij} \). Let \( 0 < \epsilon \leq 1 \) and \( \delta \in (0, 1) \).

\[
m \geq \max \left( \frac{128}{\mu_{\min}^2} \left( 1 + \frac{2}{\epsilon} \right) \ln \left( \frac{16n^2}{\delta} \right), B(\mu_{\min}) \right),
\]

then with probability at least \( 1 - \delta \) (over the random draw of \( S \sim (\mu, \mathbf{P})^n \) from which \( \hat{\mathbf{P}} \) is constructed), the score vector \( \hat{\mathbf{f}} \) produced by the least squares algorithm satisfies

\[
\| \hat{\mathbf{f}} - \mathbf{f}^* \|_\infty \leq \epsilon.
\]

This immediately yields the following sample complexity bound for the least squares algorithm to exactly recover an optimal permutation under the BTL condition:

**Algorithm 2 Least Squares/HodgeRank (Jiang et al., 2011)**

**Input:** Empirical comparison matrix \( \hat{\mathbf{P}} \in [0, 1]^{n \times n} \)

Construct empirical skew-symmetric matrix \( \hat{\mathbf{Y}} \):

\[
\hat{Y}_{ij} = \begin{cases} 
\ln \left( \frac{\hat{P}_{ij}}{P_{ji}} \right) & \text{if } i \neq j \text{ and } \hat{P}_{ij} \in (0, 1) \\
0 & \text{otherwise.}
\end{cases}
\]  

(9)

Let \( \hat{E} = \{(i, j) \in \mathcal{X} : \hat{P}_{ij} \neq 0 \text{ or } P_{ji} \neq 0\} \)

Compute \( \hat{\mathbf{f}} \in \arg \min_{\mathbf{f} \in \mathbb{R}^n} \sum_{(i,j) \in \hat{E}} \left( (f_j - f_i - \hat{Y}_{ij})^2 \right) \)

**Output:** Permutation \( \hat{\sigma} \in \text{argsort}(\hat{\mathbf{f}}) \).

**Corollary 11.** Let \((\mu, \mathbf{P})\) be such that \( \mu_{\min} > 0 \) and \( \mathbf{P} \) satisfies the BTL condition, and \( \exists(i \neq j) : P_{ij} \neq \frac{1}{2} \). Let \( \mathbf{f}^* \) be as in Eq. (8), and let \( r_{\min} = \min_{i,j : f_i^* \neq f_j^*} |f_i^* - f_j^*| \).

Let \( \delta \in (0, 1] \). If

\[
m \geq \max \left( \frac{128}{\mu_{\min}^2} \left( 1 + \frac{6}{r_{\min}} \right) \ln \left( \frac{16n^2}{\delta} \right), B(\mu_{\min}) \right),
\]

then with probability at least \( 1 - \delta \) (over the random draw of \( S \sim (\mu, \mathbf{P})^n \) from which \( \hat{\mathbf{P}} \) is constructed), the permutation \( \hat{\sigma} \) output by the least squares algorithm satisfies

\[
\hat{\sigma} \in \arg \min_{\sigma \in \mathcal{S}_n} \mathbb{E}_\mathbf{P} \mathbb{E}_\mathbf{P} \mathbf{P} \mathbf{P} [\sigma].
\]

5. Low-Noise (LN) Condition

In this section we consider the following ‘low-noise’ condition on the preference matrix \( \mathbf{P} \), which is similar to the condition studied by (Duchi et al., 2010) in a somewhat different context; as the lemma below shows, the low-noise condition includes the BTL condition as a special case.

**Definition 12 (Low-noise (LN) condition).** We say the pairwise preference matrix \( \mathbf{P} \in [0, 1]^{n \times n} \) satisfies the low-noise (LN) condition if

\[
\forall i \neq j : P_{ij} > P_{ji} \implies \sum_{k=1}^{n} P_{kj} > \sum_{k=1}^{n} P_{ki}.
\]

**Lemma 13.** If \( \mathbf{P} \in [0, 1]^{n \times n} \) satisfies the BTL condition, then it also satisfies the LN condition.

For the rest of this section (Section 5), given a pairwise preference matrix \( \mathbf{P} \in [0, 1]^{n \times n} \), define \( \mathbf{f}^* \in \mathbb{R}^n_+ \) as

\[
f_i^* = \frac{1}{n} \sum_{k=1}^{n} P_{ki}.
\]

(9)

Clearly, if \( \mathbf{P} \) satisfies the LN condition, then any permutation that ranks items \( i \in [n] \) in descending order of scores \( f_i^* \) as defined above is an optimal permutation w.r.t. the pairwise disagreement error (see Eq. (1)).

5.1. Convergence of Borda Count Algorithm

Given a pairwise comparison matrix \( \hat{\mathbf{P}} \), (the matrix version of) the Borda count algorithm (Borda, 1781; Jiang et al., 2011) (Algorithm 3) simply averages for each item \( i \) the
Algorithm 3 Borda Count (Borda, 1781; Jiang et al., 2011)

Input: Empirical comparison matrix $\mathbf{P} \in [0, 1]^{n \times n}$

For $i = 1$ to $n$: $f_i = \frac{1}{n} \sum_{k=1}^{n} \hat{P}_{ki}$

Output: Permutation $\hat{\sigma} \in \text{argsort}(\hat{f})$

fraction of times $\hat{P}_{ki}$ it has beat each other item $k$, and ranks items by this score.\(^2\) Here we show this algorithm converges to an optimal ranking under the LN condition.

The following result establishes convergence of the score vector $\hat{f}$ produced by Borda count to $f^*$ (in $L_\infty$ norm) under general $\mathbf{P}$; however the optimality of permutations obtained from $f^*$ w.r.t. the pairwise disagreement error is guaranteed only for $\mathbf{P}$ satisfying the LN condition.

Theorem 14. Let $(\mu, \mathbf{P})$ be such that $\mu_{\text{min}} > 0$ and let $f^* \in \mathbb{R}^n_+$ be defined as in Eq. (10). Let $0 < \epsilon \leq (4\sqrt{2})$ and $\delta \in (0, 1]$. If

$$m \geq \max \left\{ \frac{128}{\epsilon^2 \mu_{\text{min}}} \ln \left( \frac{4n^2}{\delta} \right), B(\mu_{\text{min}}) \right\},$$

then with probability at least $1 - \delta$ (over the random draw of $S \sim (\mu, \mathbf{P})^m$ from which $\mathbf{P}$ is constructed), the score vector $\hat{f}$ produced by the Borda count algorithm satisfies $\|\hat{f} - f^*\|_\infty \leq \epsilon$.

This immediately yields the following sample complexity bound for the Borda count algorithm to exactly recover an optimal permutation under the LN condition:

Corollary 15. Let $(\mu, \mathbf{P})$ be such that $\mu_{\text{min}} > 0$ and $\mathbf{P}$ satisfies the LN condition, and $\exists (i \neq j) : P_{ij} \neq \frac{1}{2}$. Let $f^*$ be as in Eq. (10), and let $r_{\text{min}} = \min_{i,j : f_i^* \neq f_j^*} |f_i^* - f_j^*|$. Let $\delta \in (0, 1]$. If

$$m \geq \max \left\{ \frac{1152}{r_{\text{min}}^2 \mu_{\text{min}}} \ln \left( \frac{4n^2}{\delta} \right), B(\mu_{\text{min}}) \right\},$$

then with probability at least $1 - \delta$ (over the random draw of $S \sim (\mu, \mathbf{P})^m$ from which $\mathbf{P}$ is constructed), the permutation $\hat{\sigma}$ output by the Borda count algorithm satisfies $\hat{\sigma} \in \text{argsort}_\sigma \text{er}_{\mu, \mathbf{P}}[\sigma]$.

5.2. Convergence of BTL-ML Estimator

Given a pairwise comparison matrix $\hat{\mathbf{P}}$, the BTL-ML estimator (Algorithm 4) finds a maximum likelihood score vector assuming a BTL model. Here we show this algorithm actually converges to an optimal permutation w.r.t. pairwise disagreement under the more general LN condition; in fact we obtain the same sample complexity bound for BTL-ML as for the Borda count algorithm above:

\(^2\)The standard Borda count algorithm ranks items by the number of times they beat other items; this algorithm converges to an optimal ranking under a condition involving both $\mu$ and $\mathbf{P}$. For simplicity, we count here the fraction of times an item beats other items, which allows us to restrict our attention to conditions on $\mathbf{P}$.

Algorithm 4 BTL-ML Estimator

Input: Empirical comparison matrix $\hat{\mathbf{P}} \in [0, 1]^{n \times n}$

Find maximum likelihood estimate of BTL score vector:

$$\hat{\theta} \in \arg\min_{\theta \in \mathbb{R}^n} \sum_{i < j} \left( \ln(1 + \exp(\theta_j - \theta_i)) - \hat{P}_{ij}(\theta_j - \theta_i) \right)$$

For $i = 1$ to $n$: $\hat{\omega}_i = \exp(\theta_i)$

Output: Permutation $\hat{\sigma} \in \text{argsort}(\hat{\omega})$

Theorem 16. Let $(\mu, \mathbf{P})$ be such that $\mu_{\text{min}} > 0$ and $\mathbf{P}$ satisfies the LN condition, and $\exists (i \neq j) : P_{ij} \neq \frac{1}{2}$. Let $f^*$ be as in Eq. (10), and let $r_{\text{min}} = \min_{i,j : f_i^* \neq f_j^*} |f_i^* - f_j^*|$. Let $\delta \in (0, 1]$. If

$$\frac{1152}{r_{\text{min}}^2 \mu_{\text{min}}} - \frac{4n^2}{\delta} < B(\mu_{\text{min}}),$$

then with probability at least $1 - \delta$ (over the random draw of $S \sim (\mu, \mathbf{P})^m$ from which $\hat{\mathbf{P}}$ is constructed), the permutation $\hat{\sigma}$ output by the BTL-ML algorithm satisfies $\hat{\sigma} \in \text{argsort}_\sigma \text{er}_{\mu, \mathbf{P}}[\sigma]$.

6. Generalized Low-Noise (GLN) Condition

In this section we consider a more general condition on the preference matrix $\mathbf{P}$ that we term ‘generalized low-noise’:

Definition 17 (Generalized low-noise (GLN) condition). We say the pairwise preference matrix $\mathbf{P} \in [0, 1]^{n \times n}$ satisfies the generalized low-noise (GLN) condition if $\exists \alpha \in \mathbb{R}^n$ such that

$$\forall i \neq j : P_{ij} > P_{ji} \implies \sum_{k=1}^{n} \alpha_k P_{kj} > \sum_{k=1}^{n} \alpha_k P_{ki}.$$

Clearly, the LN condition of Section 5 is a special case with $\alpha_k = 1 \forall k \in [n]$. Moreover, if $\mathbf{P}$ satisfies the GLN condition for some vector $\alpha$, then any permutation that ranks items in decreasing order of scores $f_i = \sum_{k=1}^{n} \alpha_k P_{ki}$ is an optimal permutation w.r.t. the pairwise disagreement error.

As our experiments will show, none of the four common rank aggregation algorithms considered in Sections 4–5 above are guaranteed to converge to an optimal ranking under a general probability distribution satisfying the GLN condition. Below we propose a new SVM-based rank aggregation algorithm which satisfies this property.

6.1. New Algorithm: SVM-RankAggregation

We will need the following definition:

Definition 18 (P-Induced Dataset). For any matrix $\mathbf{P} \in [0, 1]^{n \times n}$, define the $\mathbf{P}$-induced dataset $\mathcal{S}_\mathbf{P} = \{v_{ij}, z_{ij}\}_{i < j}$ as consisting of the $\binom{n}{2}$ vectors $v_{ij} = (\mathbf{P}_i - \mathbf{P}_j) \in \mathbb{R}^n$ ($i < j$), where $\mathbf{P}_i$ denotes the $i$-th column of $\mathbf{P}$, together with binary labels $z_{ij} = \text{sign}(P_{ji} - P_{ij}) \in \{\pm 1\}$.
6.2. Convergence of SVM-RankAggregation Algorithm

We now show that the SVM-RankAggregation algorithm converges to an optimal ranking under the GLN condition, and give a sample complexity bound for SVM-RankAggregation to exactly recover an optimal ranking:

**Theorem 20.** Let \( (\mu, P) \) be such that \( \mu_{\min} > 0 \) and \( P \) satisfies the GLN condition for some vector \( \alpha \in \mathbb{R}^n \), and \( P_{ij} \neq \frac{1}{2} \forall i, j \). Let \( \gamma = \min_{i,j} |P_{ij} - \frac{1}{2}| \), and let \( r_{\min} = \min_{i,j} [\alpha^\top (P_{ij} - P_{ij}^S)] \). Let \( \delta \in (0, 1) \). If

\[
m \geq \max \left( \frac{2048 n}{\left(r_{\min}^2 \mu_{\min}\right)^2} \log \left( \frac{16n^3}{\delta} \right), \frac{128}{\gamma^2 \mu_{\min}} \log \left( \frac{8n^2}{\delta} \right), B(\mu_{\min}) \right),
\]

then with probability at least \( 1 - \delta \) (over the random draw of \( S' \sim (\mu, P)^m \) from which \( \hat{P} \) is constructed), the permutation \( \hat{\sigma} \) output by SVM-RankAggregation satisfies

\[
\hat{\sigma} \in \arg\min_{\sigma \in S_n} \mathcal{F}_{\mu, P}^D[\sigma].
\]

7. Experiments

In this section we report results of experiments designed to verify our convergence results and investigate the tightness of the corresponding sample complexity bounds.

7.1. Convergence under BTL

Our first experiment was with BTL distributions for \( n = 5, 10, 20 \). For each \( n \), we constructed \( P \) using a random BTL vector \( w \in \mathbb{R}^n_+ \) (each component \( w_i \) chosen uniformly at random from \([0, 1]\)), taking \( \mu \) to be the uniform distribution over the \( \binom{n}{2} \) item pairs, and generated 100 random samples from \((\mu, P)\) for each of several sample sizes \( m \). We then ran the 5 algorithms analyzed in Sections 4-6 on the generated samples, and for each \( n \) and \( m \), computed the fraction of times an optimal permutation was recovered by each algorithm. The results are shown in Figure 2; as can be seen, for sufficiently large sample size, all 5 algorithms recover an optimal permutation with high probability. Similar results were obtained with non-uniform \( \mu \).

![Figure 2](image_url)

**Figure 2.** Fraction of times an optimal ranking was recovered by various algorithms under a BTL distribution (out of 100 random runs), for increasing sample sizes \( m \), and for different numbers of items \( n \) (left to right: \( n = 5, 10, 20 \)).

7.2. Convergence under LN

For our next experiment, we constructed a distribution that satisfies the LN condition but not the BTL condition; the preference matrix \( P \) we used (with \( n = 4 \)) is shown below:

\[
P = \begin{bmatrix}
0 & 0.8 & 0.51 & 0.51 \\
0.2 & 0 & 0.9 & 0.7 \\
0.49 & 0.1 & 0 & 0.65 \\
0.49 & 0.3 & 0.35 & 0
\end{bmatrix}.
\]

In this case we used a random distribution \( \mu \) over the item pairs (specifically, \( \frac{n}{2} \)) numbers \( u_{ij} \) were each chosen uniformly at random from \([0, 1]\), and then normalized to yield \( \mu_{ij} = u_{ij} / \sum_{k i} u_{ki} \). Again, we generated 100 random samples from \((\mu, P)\) for each of several sample sizes \( m \), ran the 5 algorithms on these samples, and in each case computed the fraction of times an optimal permutation was recovered. The results are shown in Figure 3 (left); as can be seen, the rank centrality and least squares algorithms fail to recover an optimal permutation under this distribution.
7.3. Convergence under GLN

For our third experiment, we constructed a distribution that satisfies the GLN condition but not the LN condition; the preference matrix \( P \) we used (with \( n = 5 \)) is shown below:

\[
P = \begin{bmatrix}
0 & 0.51 & 0.46 & 0.4 & 0.4 \\
0.49 & 0 & 0.49 & 0.4 & 0.4 \\
0.54 & 0.51 & 0 & 0.4 & 0.4 \\
0.6 & 0.6 & 0.6 & 0 & 0.4 \\
0.6 & 0.6 & 0.6 & 0.6 & 0 \\
\end{bmatrix}
\]

It can be verified that \( P \) satisfies the GLN condition with \( \alpha = (-0.4530, -2.6021, 1.4660, 3.0796, 3.7197)^T \); however \( P \) does not satisfy the LN condition since \( P_{12} > P_{21} \) but \( \sum_k P_{k1} > \sum_k P_{k2} \). Again we used a random distribution \( \mu \) over the item pairs as above. The results are shown in Figure 3 (right); here only the SVM-RankAggregation algorithm successfully recovers an optimal ranking.

7.4. Tightness of Sample Complexity Bounds

Our final set of experiments was designed to evaluate the tightness of our sample complexity bounds. We first used BTL distributions generated similarly as described in Section 7.1 for various \( n \) between 5 and 20, and evaluated both the actual number of samples required by each algorithm to recover an optimal ranking at least 95% of the time, and the corresponding upper bounds on sample complexity, as a function of \( n \). The results are shown in Figure 4 (left); in most cases, the shapes of the upper bounds are largely similar to those of the actual sample complexity curves.

Figure 4 (left) also suggests the upper bound for the rank centrality algorithm is significantly looser than those for other algorithms. This bound, which builds on techniques of (Negahban et al., 2012), involves an additional \( \frac{r_{\text{max}}}{r_{\text{min}}} \) term not present in the other bounds. To investigate this, we designed BTL distributions for \( n = 5 \) with increasing values of \( \frac{r_{\text{max}}}{r_{\text{min}}} \) (keeping the \( r_{\text{min}} \) term corresponding to the LN bounds in Section 5 fixed), and evaluated the sample complexity and corresponding upper bounds as a function of this ratio. The results are shown in Figure 4 (right). As can be seen, the dependence of the rank centrality upper bound on this term appears to be superfluous, and likely an artefact of the current analysis technique, which is based on that of (Negahban et al., 2012). In future work, we intend to explore alternative techniques for obtaining a tighter bound for the rank centrality algorithm.\(^3\) We also plan to explore whether the additional factor of \( n \) in the rank centrality and SVM-RankAggregation bounds can be removed.

8. Conclusion

The problem of rank aggregation from pairwise comparison data has received much interest recently. We have analyzed various algorithms for this problem, and have shown that under a natural statistical model, where pairwise comparisons are drawn randomly and independently from some underlying probability distribution, the rank centrality and least squares algorithms converge to an optimal ranking under a BTL condition, while the Borda count and BTL-ML algorithms converge to an optimal ranking under a more general LN condition. However, none of these existing algorithms converge under the more general GLN condition; we have proposed a new SVM-based rank aggregation algorithm for which such convergence is guaranteed. Future work includes improving the analysis to obtain tighter bounds, and extending the analysis to other algorithms.

Acknowledgments. Thanks to the anonymous reviewers for helpful comments. AR is supported by a Microsoft Research India PhD Fellowship. SA thanks DST and the Indo-US Science & Technology Forum for their support.

\(^3\)We note that the least squares sample complexity also shows a slight dependence on the \( \frac{r_{\text{max}}}{r_{\text{min}}} \) term; this is due to the fact that this term is connected to \( P_{\text{min}} \), the dependence on which appears to be captured correctly in our bound for least squares.
References


