Ranking from Stochastic Pairwise Preferences: Recovering Condorcet Winners and Tournament Solution Sets at the Top

Arun Rajkumar ARUN_R@CSA.IISC.ERNET.IN
Suprovat Ghoshal SUPROVAT.GHOSHAL@CSA.IISC.ERNET.IN
Indian Institute of Science

Lek-Heng Lim LEKHENG@GALTON.UCHICAGO.EDU
University of Chicago

Shivani Agarwal SHIVANI@CSA.IISC.ERNET.IN
Indian Institute of Science

Abstract

We consider the problem of ranking n items from stochastically sampled pairwise preferences. It was shown recently that when the underlying pairwise preferences are acyclic, several algorithms including the Rank Centrality algorithm, the Matrix Borda algorithm, and the SVM-RankAggregation algorithm succeed in recovering a ranking that minimizes a global pairwise disagreement error (Rajkumar and Agarwal, 2014). In this paper, we consider settings where pairwise preferences can contain cycles. In such settings, one may still like to be able to recover 'good' items at the top of the ranking. For example, if a Condorcet winner exists that beats every other item, it is natural to ask that this be ranked at the top. More generally, several tournament solution concepts such as the top cycle, Copeland set, Markov set and others have been proposed in the social choice literature for choosing a set of winners in the presence of cycles. We show that existing algorithms can fail to perform well in terms of ranking Condorcet winners and various natural tournament solution sets at the top. We then give alternative ranking algorithms that provably rank Condorcet winners, top cycles, and other tournament solution sets of interest at the top. In all cases, we give finite sample complexity bounds for our algorithms to recover such winners. As a by-product of our analysis, we also obtain an improved sample complexity bound for the Rank Centrality algorithm to recover an optimal ranking under a Bradley-Terry-Luce (BTL) condition, which answers an open question of Rajkumar and Agarwal (2014).

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

1. Introduction

There has been much interest in recent years in designing algorithms for aggregating pairwise preferences to rank a set of items (Fürnkranz & Hüllermeier, 2010; Ailon, 2011; Lu & Boutilier, 2011; Jiang et al., 2011; Jamieson & Nowak, 2011; Negahban et al., 2012; Osting et al., 2013; Wauthier et al., 2013; Busa-Fekete et al., 2014a;b; Rajkumar & Agarwal, 2014). Indeed, the need for aggregating pairwise preferences arises in many domains where fully ordered preferences may be hard to obtain, e.g. in customer surveys, recommender systems, sports team rankings etc.

In many such settings, particularly those related to applications such as customer surveys and recommender systems, it is common to assume that the pairwise comparisons follow some (unknown) underlying statistical model. An important question that arises then is, as we observe more and more data from the underlying statistical model, do the algorithms used converge to a 'good' ranking solution?

Recently, Rajkumar & Agarwal (2014) showed that when the underlying pairwise preferences are acyclic, several algorithms including the Rank Centrality (RC), Matrix Borda (MB) and SVM-RankAggregation (SVM-RA) algorithms can converge to an optimal ranking solution in terms of minimizing a pairwise disagreement error. However, in practice, pairwise preferences often contain cycles. In such settings, minimizing the pairwise disagreement error is typically NP-hard, but one may still want to converge to a ranking that places 'good' items at the top.

In this paper, we take 'good' items to refer to Condorcet winners (which beat all other items) when they exist, and more generally, to tournament solution sets such as top cycles and Copeland and Markov sets, which have been used to define 'winners' in the computational social choice literature (Moulin, 1986; De Donder et al., 2000; Laslier, 1997; Brandt et al., 2015). We show that when preferences contain cycles, the RC, MB and SVM-RA algorithms can fail to rank such 'good' items at the top; we then propose three

Table 1. Ranking behavior of various algorithms for ranking from stochastic pairwise preferences under various conditions on the underlying statistical model. Here $\mathcal{P}^{\text{BTL}}$ is the class of preferences following a Bradley–Terry–Luce model, $\mathcal{P}^{\text{LN}}$ the class of preferences following a certain 'low-noise' model, $\mathcal{P}^{\text{DAG}}$ the class of acyclic preferences, and $\mathcal{P}^{\text{CW}}$ the class of preferences that admit a Condorcet winner; it is known that $\mathcal{P}^{\text{BTL}} \subset \mathcal{P}^{\text{LN}} \subset \mathcal{P}^{\text{DAG}} \subset \mathcal{P}^{\text{CW}}$ (see Section 2.3 and Figures 1 and 2). The first three algorithms were analyzed with respect to pairwise disagreement in (Rajkumar & Agarwal, 2014). Results established in this paper are highlighted in gray.

Algorithm	Minimizes pairwise disagreement? $\mathcal{P}^{\text{BTL}}$	$\mathcal{P}^{\text{LN}}$	$\mathcal{P}^{\text{DAG}}$	Ranks Condorcet winner at top (in $\mathcal{P}^{\text{CW}}$)?	Ranks top cycle at top (in $\mathcal{P}$)?	Ranks Copeland set at top (in $\mathcal{P}$)?	Ranks Markov set at top (in $\mathcal{P}$)?
Rank Centrality	✓	×	×	×	×	×	×
Matrix Borda	✓	✓	×	×	×	×	×
SVM-RankAggregation	✓	✓	✓	×	×	×	×
Matrix Copeland	✓	✓	✓	✓	✓	✓	×
Unweighted Markov	×	×	×	✓	✓	×	✓
Parametrized Markov	✓	×	×	✓	✓	×	✓

$$\mathcal{P}_n^{\text{TC}(k)} = \{\mathbf{P} \in \mathcal{P}_n : |\,\text{TC}(\mathbf{P})| = k\}$$
$$\mathcal{P}_n^{\text{CW}} = \{\mathbf{P} \in \mathcal{P}_n : \mathbf{P} \text{ has a Condorcet winner}\} = \mathcal{P}_n^{\text{TC}(1)}$$
$$\mathcal{P}_n^{\text{DAG}} = \{\mathbf{P} \in \mathcal{P}_n : G_{\mathbf{P}} = ([n], E_{\mathbf{P}}) \text{ is a DAG}\}$$
$$\mathcal{P}_n^{\text{LN}} = \{\mathbf{P} \in \mathcal{P}_n : i \succ_{\mathbf{P}} j \implies \textstyle\sum_{k=1}^n p_{ki} > \sum_{k=1}^n p_{kj}\}$$
$$\mathcal{P}_n^{\text{BTL}} = \{\mathbf{P} \in \mathcal{P}_n : \exists \mathbf{w} \in \mathbb{R}_+^n \text{ s.t. } p_{ij} = \tfrac{w_j}{w_i + w_j}\ \forall i \neq j\}$$

Figure 1. Definitions of various conditions on $\mathbf{P}$.

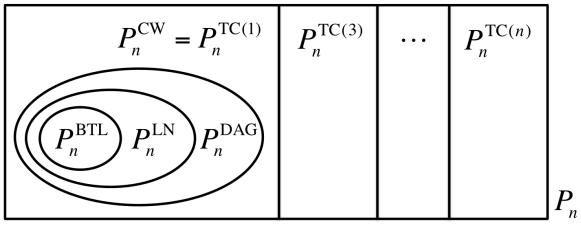

Figure 2. Relationships between various conditions on $\mathbf{P}$.

new algorithms, namely the Matrix Copeland, Unweighted Markov, and Parametrized Markov algorithms, that provably rank such good items at the top (see Table 1 for a summary). In all cases, we provide explicit sample complexity bounds for these algorithms to recover (with high probability) a ranking that places the desired elements at the top. As a by-product of our analysis of the Parametrized Markov algorithm, we also obtain a tighter sample complexity bound for the RC algorithm, thereby answering an open question of Rajkumar & Agarwal (2014).

2. Preliminaries and Background

2.1. Problem Setup

Let $[n] = \{1, \ldots, n\}$ denote the set of n items to be ranked, and $\binom{[n]}{2} = \{(i,j) \in [n] \times [n] : i < j\}$ the set of $\binom{n}{2}$ pairs that can be compared. We assume that there is an underlying (unknown) probability distribution $\mu \in \Delta_{\binom{n}{2}}$ such that each time a comparison is to be made, a pair (i,j) is selected with probability μ_{ij}, and that for each $i < j$, there is an (unknown) parameter $p_{ij} \in [0,1]$ such that when items i and j are compared, item j 'beats' (or is preferred to) item i with probability p_{ij}. For each $i < j$, define $p_{ji} = 1 - p_{ij}$; also define $p_{ii} = 0$ for every i. Denote by $\mathbf{P} = [p_{ij}]$ the resulting (unknown) pairwise preference matrix. Given a finite sample of pairwise comparisons $S = ((i_1, j_1, y_1), \ldots, (i_m, j_m, y_m)) \in \left(\binom{[n]}{2} \times \{0,1\}\right)^m$ drawn according to $(\mu, \mathbf{P})$ as above (where pairs (i_k, y_k) are drawn iid according to μ, and given (i_k, j_k), $y_k \sim \text{Bernoulli}(p_{i_k j_k})$), the goal is to construct a ranking or permutation $\sigma \in \mathcal{S}_n$ that ranks 'good' elements at the top. Most algorithms we consider operate on an empirical pairwise comparison matrix $\widehat{\mathbf{P}}$ constructed from S as follows:

$$\widehat{p}_{ij} = \begin{cases} m_{ij}^{(1)} / m_{ij} & \text{if } i < j \text{ and } m_{ij} > 0 \\ 1 - \left(m_{ji}^{(1)} / m_{ji}\right) & \text{if } i > j \text{ and } m_{ji} > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $m_{ij} = \sum_{k=1}^m \mathbf{1}(i_k = i, j_k = j)$;
$m_{ij}^{(1)} = \sum_{k=1}^m \mathbf{1}(i_k = i, j_k = j, y_k = 1)$.

Binary Relation and Tournament Induced by $\mathbf{P}$. We will assume throughout that the matrix $\mathbf{P}$ does not involve ties, i.e., $p_{ij} \neq \frac{1}{2}$ for all i, j, and will find it convenient to define the binary relation $\succ_{\mathbf{P}}$ on $[n]$ as follows:

$$i \succ_{\mathbf{P}} j \iff p_{ij} < \tfrac{1}{2} \qquad \text{for all } i, j \in [n]\,.$$

We will also define the edge set associated with $\mathbf{P}$ as

$$E_{\mathbf{P}} = \{(i,j) \in [n] \times [n] : i \succ_{\mathbf{P}} j\}\,,$$

and the induced graph $G_{\mathbf{P}} = \left([n], E_{\mathbf{P}}\right)$. Note that under the assumption $p_{ij} \neq \frac{1}{2}$ for all i, j, the graph $G_{\mathbf{P}}$ is always a complete directed graph, i.e., a tournament.

2.2. Measuring Ranking Performance

We will consider several notions of 'goodness' of a ranking with respect to the underlying pairwise preferences $\mathbf{P}$.

Pairwise Disagreement. Define the *pairwise disagreement* (PD) error of a permutation $\sigma \in \mathcal{S}_n$ w.r.t. $\mathbf{P}$ as

$$\mathrm{er}^{\mathrm{PD}}_{\mathbf{P}}[\sigma] = \sum_{i \neq j} \mathbf{1}(p_{ij} < \tfrac{1}{2}) \cdot \mathbf{1}(\sigma(i) > \sigma(j)) \,.$$

The pairwise disagreement error measures the number of pairs on which σ disagrees with $\mathbf{P}$, and is a natural measure of overall ranking performance; a variant of this was studied in (Rajkumar & Agarwal, 2014). In general, however, even if $\mathbf{P}$ is known directly, identifying an optimal ranking w.r.t. PD error is computationally hard (it corresponds to the NP-hard minimum feedback arc set problem). Thus one can hope to recover an optimal ranking w.r.t. PD error only under certain restrictive conditions on $\mathbf{P}$; indeed, all conditions studied in (Rajkumar & Agarwal, 2014) assumed $G_{\mathbf{P}}$ is acyclic.

In this paper, we are interested in algorithms that can ensure 'good' elements are ranked at the top even when the graph $G_{\mathbf{P}}$ contains cycles; we formalize the notion of 'good' below in terms of Condorcet winners, top cycles, and other tournament solution sets (see also Figure 3).

Condorcet Winners. An element $i \in [n]$ is said to be a *Condorcet winner* w.r.t. $\mathbf{P}$ if it beats all other elements under $\mathbf{P}$, i.e., if $i \succ_{\mathbf{P}} j \; \forall j \neq i$. A Condorcet winner does not always exist, but when it does, it is unique, and it is then natural to want this element to be ranked at the top. When a Condorcet winner exists, we will denote it by $\mathrm{CW}(\mathbf{P})$.

Top Cycles. More generally, one can ask that elements of the top cycle of $\mathbf{P}$ be ranked at the top. The *top cycle* of $\mathbf{P}$, denoted $\mathrm{TC}(\mathbf{P})$, is defined as the smallest set of elements $W \subseteq [n]$ such that every element in W beats every element not in W under $\mathbf{P}$, i.e. such that $i \succ_{\mathbf{P}} j \; \forall i \in W, j \notin W$. A Condorcet winner corresponds to a top cycle of size 1, i.e. when $|\mathrm{TC}(\mathbf{P})| = 1$, we have $\mathrm{TC}(\mathbf{P}) = \{\mathrm{CW}(\mathbf{P})\}$. The top cycle is also referred to as the Smith set in voting theory literature (Smith, 1973).

Copeland and Markov Sets. The top cycle is one form of *tournament solution set*, which selects a set of elements considered to be 'winners' in a tournament (Laslier, 1997; Brandt et al., 2015). One can also consider other notions of tournament solution sets associated with $\mathbf{P}$ (or more specifically, associated with the tournament $G_{\mathbf{P}}$ induced by $\mathbf{P}$), and ask that elements of the desired tournament solution set be ranked at the top. Two such tournament solution sets that we will be interested in are the Copeland set and the Markov set, both of which are 'score-based' solution sets that select all elements maximizing a certain 'score'. The *Copeland set* of $\mathbf{P}$, denoted $\mathrm{CO}(\mathbf{P})$, selects elements with maximal out-degree in the (unweighted) tournament $G_{\mathbf{P}}$: $\mathrm{CO}(\mathbf{P}) = \mathrm{argmax}_{i \in [n]} \, d(i)$, where $d(i) = \sum_j \mathbf{1}(i \succ_{\mathbf{P}} j)$. The *Markov set* of $\mathbf{P}$, denoted $\mathrm{MA}(\mathbf{P})$, selects elements with highest stationary probability under a certain Markov chain on the tournament $G_{\mathbf{P}}$: $\mathrm{MA}(\mathbf{P}) = \mathrm{argmax}_{i \in [n]} \, \pi_i$,

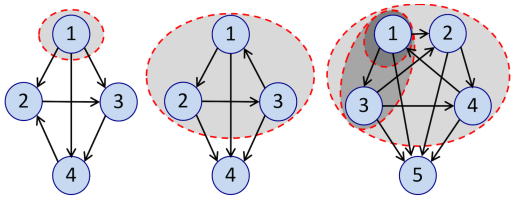

Figure 3. **Left:** A tournament on $n = 4$ nodes in which there are cycles but node 1 is a Condorcet winner. **Middle:** A tournament on $n = 4$ nodes in which there is no Condorcet winner. Here the set of nodes $\{1, 2, 3\}$ forms a top cycle of size 3; in this case this set also corresponds to the Copeland and Markov sets. **Right:** A tournament on $n = 5$ nodes in which the top cycle $\{1, 2, 3, 4\}$ is a strict superset of the Copeland set $\{1, 3\}$ and Markov set $\{1\}$.

where $\boldsymbol{\pi}$ is the stationary probability vector of the Markov chain $\widetilde{\mathbf{P}}$ defined as $\widetilde{p}_{ij} = \frac{1}{n}$ for all $(i, j) : j \succ_{\mathbf{P}} i$ and $\widetilde{p}_{ii} = 1 - \sum_{j \neq i} \widetilde{p}_{ij}$ for all i. It is known that the Copeland and Markov sets are refinements of the top cycle, i.e. that $\mathrm{CO}(\mathbf{P}) \subseteq \mathrm{TC}(\mathbf{P})$ and $\mathrm{MA}(\mathbf{P}) \subseteq \mathrm{TC}(\mathbf{P})$, and that these containments can be strict (see Figure 3 for illustrations).

2.3. Conditions on the Pairwise Preference Matrix P

As noted above, we are interested in understanding the ranking behavior of various algorithms under various conditions on the pairwise preference matrix $\mathbf{P}$. Let $\mathcal{P}_n$ denote the set of all such matrices on n items not involving ties:

$$\mathcal{P}_n = \big\{\mathbf{P} \in [0,1]^{n \times n} : p_{ij} = 1 - p_{ji} \; \forall i \neq j \,; \; p_{ij} \neq \tfrac{1}{2} \; \forall i, j \,; \; p_{ii} = 0 \; \forall i \big\} \,.$$

For each $k \in [n]$, define $\mathcal{P}_n^{\mathrm{TC}(k)}$ to be the set of preference matrices $\mathbf{P}$ in $\mathcal{P}_n$ that have a top cycle of size k.[1] Clearly, $\mathcal{P}_n^{\mathrm{TC}(1)}$ is the set of preference matrices in $\mathcal{P}_n$ that have a Condorcet winner; we will also denote this by $\mathcal{P}_n^{\mathrm{CW}}$. Next, define $\mathcal{P}_n^{\mathrm{DAG}}$ to be the set of preference matrices $\mathbf{P}$ in $\mathcal{P}_n$ for which the directed graph $G_{\mathbf{P}}$ induced by the binary relation $\succ_{\mathbf{P}}$ is acyclic,[2] and define $\mathcal{P}_n^{\mathrm{LN}}$ to be the set of matrices in $\mathcal{P}_n$ satisfying the 'low-noise' (LN) condition (Rajkumar & Agarwal, 2014). Finally, define $\mathcal{P}_n^{\mathrm{BTL}}$ to be the set of preference matrices in $\mathcal{P}_n$ that follow a Bradley-Terry-Luce (BTL) model. Formal definitions of all these sets are given in Figure 1. It can be verified that $\mathcal{P}_n^{\mathrm{TC}(1)}, \ldots, \mathcal{P}_n^{\mathrm{TC}(n)}$ form a partition of $\mathcal{P}_n$, and that

$$\mathcal{P}_n^{\mathrm{BTL}} \subset \mathcal{P}_n^{\mathrm{LN}} \subset \mathcal{P}_n^{\mathrm{DAG}} \subset \mathcal{P}_n^{\mathrm{CW}} = \mathcal{P}_n^{\mathrm{TC}(1)} \,,$$

with each containment above being strict (see Figure 2). When n is clear from context, we will write $\mathcal{P}$, $\mathcal{P}^{\mathrm{BTL}}$, etc.

3. Related Work and Existing Results

As noted above, there has been much work in recent years on developing algorithms for ranking a set of items from

[1] Note that under a strict tournament (where there are no ties), there cannot be a top cycle of size 2, i.e. $\mathcal{P}_n^{\mathrm{TC}(2)} = \emptyset$.

[2] The DAG condition is equivalent to the 'generalized low-noise' (GLN) condition studied in (Rajkumar & Agarwal, 2014).

pairwise preferences; below we discuss three specific algorithms that form the backdrop to our work. Before doing so, we point out that our work differs from that of (Brandt & Fischer, 2007), where any ranking algorithm that given $\mathbf{P} \in \mathcal{P}_n$ produces a score vector $\mathbf{f} \in \mathbb{R}^n$ is considered to induce a corresponding tournament solution $\mathrm{TS}(\mathbf{P}) = \mathrm{argmax}_{i\in[n]} f_i$; in particular, Brandt & Fischer study properties of the tournament solution induced in this manner by PageRank scores. Our work also differs from (Braverman & Mossel, 2009; Wauthier et al., 2013), where pairs are sampled only once and the conditions on $\mathbf{P}$ are significantly stronger than those assumed here.

Three representative ranking algorithms studied in recent years that are most related to our work are the Rank Centrality (RC) algorithm (Negahban et al., 2012), the Matrix Borda (MB) algorithm (Rajkumar & Agarwal, 2014; Borda, 1781), and the SVM-RankAggregation (SVM-RA) algorithm (Rajkumar & Agarwal, 2014). RC produces a ranking by sorting the stationary probabilities of a Markov chain constructed from pairwise data. MB ranks items based on the average probability of an item being preferred over the rest of the items. SVM-RA constructs a binary classification dataset from the pairwise data and ranks items based on the maximum margin hyperplane for this dataset. Detailed descriptions of these algorithms are given in the appendix; below we summarize results of (Rajkumar & Agarwal, 2014) which establish convergence of these algorithms to an optimal permutation w.r.t. PD error within $\mathcal{P}^{\mathrm{BTL}}$, $\mathcal{P}^{\mathrm{LN}}$, and $\mathcal{P}^{\mathrm{DAG}}$, respectively (see Table 2 for notation).[3]

Theorem 1 (Rank Centrality minimizes PD error in $\mathcal{P}^{\mathrm{BTL}}$)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n^{\mathrm{BTL}}$. Let $\boldsymbol{\pi}$ be the stationary probability vector of the Markov chain $\overline{\mathbf{P}}$ defined as $\overline{p}_{ij} = \frac{p_{ij}}{n}\ \forall i \neq j$ and $\overline{p}_{ii} = 1 - \sum_{j\neq i} \overline{p}_{ij}\ \forall i$, and let $r_{\min} = \min_{i,j:\pi_i\neq\pi_j} |\pi_i - \pi_j|$. Let $\delta \in (0,1]$. If*

$$m \geq \max\left(\frac{9216\, n}{r_{\min}^2 \mu_{\min}^2} \frac{1}{p_{\min}^2}\left(\frac{1}{p_{\min}} - 1\right)^3 \ln\left(\frac{16n^2}{\delta}\right),\ B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\mathrm{RC}}$ produced by running the RC algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\mathrm{RC}} \in \mathrm{argmin}_{\sigma\in\mathcal{S}_n} \mathrm{er}_{\mathbf{P}}^{\mathrm{PD}}[\sigma]\,.$$

Theorem 2 (Matrix Borda minimizes PD error in $\mathcal{P}^{\mathrm{LN}}$)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n^{\mathrm{LN}}$. Let $f_i^* = \frac{1}{n}\sum_{k=1}^n p_{ki}\ \forall i$, and let $r_{\min} = \min_{i,j:f_i^*\neq f_j^*} |f_i^* - f_j^*|$. Let $\delta \in (0,1]$. If*

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

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\mathrm{MB}}$ produced by running the MB algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\mathrm{MB}} \in \mathrm{argmin}_{\sigma\in\mathcal{S}_n} \mathrm{er}_{\mathbf{P}}^{\mathrm{PD}}[\sigma]\,.$$

[3] We note that the form of the PD error considered in (Rajkumar & Agarwal, 2014) is slightly different from that considered here, but within $\mathcal{P}^{\mathrm{DAG}}$ both have the same sets of minimizers.

Table 2. Useful quantities associated with μ and $\mathbf{P}$.

$\mu_{\min}$	$\min_{i<j} \mu_{ij}$
B_μ	$3\left(\frac{12}{\mu_{\min}^2} + 3\right) \ln\left(\frac{12}{\mu_{\min}^2} + 3\right)$
$p_{\min}$	$\min_{i\neq j} p_{ij}$
$\gamma_{\min}$	$\min_{i\neq j} \left\lvert p_{ij} - \frac{1}{2}\right\rvert$
γ_{TC}	$\min_{i\in \mathrm{TC}(\mathbf{P}), j\notin \mathrm{TC}(\mathbf{P})} \left\lvert p_{ij} - \frac{1}{2}\right\rvert$

Theorem 3 (SVM-RA minimizes PD error in $\mathcal{P}^{\mathrm{DAG}}$)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n^{\mathrm{DAG}}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^n$ be such that $i \succ_{\mathbf{P}} j \implies \sum_{k=1}^n \alpha_k p_{ki} > \sum_{k=1}^n \alpha_k p_{jk}$,[4] and let $r_{\min}^{\boldsymbol{\alpha}} = \min_{i\neq j} \frac{|\sum_{k=1}^n \alpha_k (P_{ki} - P_{kj})|}{\|\boldsymbol{\alpha}\|_2}$. Let $\delta \in (0,1]$. If*

$$m \geq \max\left(\frac{2048\, n}{(r_{\min}^{\boldsymbol{\alpha}})^2 \mu_{\min}^2} \ln\left(\frac{16n^3}{\delta}\right), \frac{128}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{8n^2}{\delta}\right),\ B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\text{SVM-RA}}$ produced by running SVM-RA on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\text{SVM-RA}} \in \mathrm{argmin}_{\sigma\in\mathcal{S}_n} \mathrm{er}_{\mathbf{P}}^{\mathrm{PD}}[\sigma]\,.$$

We will make use of the following result of (Rajkumar & Agarwal, 2014) on concentration of entries of $\widehat{\mathbf{P}}$ around $\mathbf{P}$:

Lemma 4. *Let $\mu_{\min} > 0$ and $\mathbf{P} \in \mathcal{P}_n$. Fix any $i \neq j$. Let $0 < \epsilon < 4\sqrt{2}$. If $m \geq \max\left(\frac{128}{\epsilon^2 \mu_{\min}^2} \ln(\frac{4}{\delta}), B_\mu\right)$, then with probability at least $1-\delta$ (over S), $|p_{ij} - \widehat{p}_{ij}| < \epsilon$.*

4. Performance of RC, MB and SVM-RA Algorithms Under Cyclic Preferences

It is easy to see that for $\mathbf{P} \in \mathcal{P}_n^{\mathrm{CW}}$,

$$\widehat{\sigma} \in \mathrm{argmin}_{\sigma\in\mathcal{S}_n} \mathrm{er}_{\mathbf{P}}^{\mathrm{PD}}[\sigma] \implies \widehat{\sigma}^{-1}(1) = \mathrm{CW}(\mathbf{P})\,.$$

Therefore it follows from Theorems 1–3 that (when given a sufficiently large sample, with high probability) the RC algorithm ranks the Condorcet winner at the top for $\mathbf{P} \in \mathcal{P}^{\mathrm{BTL}}$; the MB algorithm does so for $\mathbf{P} \in \mathcal{P}^{\mathrm{LN}}$; and the SVM-RA algorithm does so for $\mathbf{P} \in \mathcal{P}^{\mathrm{DAG}}$. However, as the following examples show, outside $\mathcal{P}^{\mathrm{DAG}}$, these algorithms can fail to rank the Condorcet winner, top cycle, or Copeland or Markov sets at the top (in fact RC and MB can fail to do so even in $\mathcal{P}^{\mathrm{LN}} \setminus \mathcal{P}^{\mathrm{BTL}}$ and $\mathcal{P}^{\mathrm{DAG}} \setminus \mathcal{P}^{\mathrm{LN}}$, respectively; see Examples 6–7 in the supplementary material).

Example 1 (For $\mathbf{P} \in \mathcal{P}^{\mathrm{CW}} \setminus \mathcal{P}^{\mathrm{DAG}}$, RC, MB and SVM-RA algorithms can fail to rank the Condorcet winner/Copeland set/Markov set at the top)**.** *Let $n = 7$, and consider*

$$\mathbf{P} = \begin{bmatrix} 0 & 0.49 & 0.49 & 0.49 & 0.49 & 0.49 & 0.49 \\ 0.51 & 0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.51 & 0.9 & 0 & 0.4 & 0.4 & 0.6 & 0.6 \\ 0.51 & 0.9 & 0.6 & 0 & 0.4 & 0.6 & 0.4 \\ 0.51 & 0.9 & 0.6 & 0.6 & 0 & 0.4 & 0.6 \\ 0.51 & 0.9 & 0.4 & 0.4 & 0.6 & 0 & 0.4 \\ 0.51 & 0.9 & 0.4 & 0.6 & 0.4 & 0.6 & 0 \end{bmatrix}.$$

[4] Such an $\boldsymbol{\alpha}$ always exists by equivalence of the DAG and GLN conditions (Rajkumar & Agarwal, 2014).

The tournament induced by $\mathbf{P}$ *is shown in Figure 4 (left). It can be verified that* $\mathbf{P} \in \mathcal{P}^{\text{CW}} \setminus \mathcal{P}^{\text{DAG}}$*, with* $\text{CW}(\mathbf{P}) = 1$ *and correspondingly* $\text{CO}(\mathbf{P}) = \text{MA}(\mathbf{P}) = \{1\}$*. However the permutations* σ_{RC}*,* σ_{MB} *and* $\sigma_{\text{SVM-RA}}$ *produced by running RC, MB and SVM-RA on the above preference matrix* $\mathbf{P}$ *do not rank item 1 at the top; instead, it can be verified that* $\sigma_{\text{RC}}^{-1}(1) = \sigma_{\text{MB}}^{-1}(1) = 2$ *and* $\sigma_{\text{SVM-RA}}^{-1}(1) = 7$*.*

Example 2 (For $\mathbf{P} \notin \mathcal{P}^{\text{CW}}$, RC, MB and SVM-RA algorithms can fail to rank the top cycle/Copeland set/Markov set at the top)**.** *Let* $n = 6$*, and consider*

$$\mathbf{P} = \begin{bmatrix} 0 & 0.3 & 0.7 & 0.4 & 0.45 & 0.45 \\ 0.7 & 0 & 0.3 & 0.4 & 0.45 & 0.45 \\ 0.3 & 0.7 & 0 & 0.4 & 0.45 & 0.45 \\ 0.6 & 0.6 & 0.6 & 0 & 0.1 & 0.1 \\ 0.55 & 0.55 & 0.55 & 0.9 & 0 & 0.6 \\ 0.55 & 0.55 & 0.55 & 0.9 & 0.4 & 0 \end{bmatrix}.$$

The tournament induced by $\mathbf{P}$ *is shown in Figure 4 (middle). It can be verified that* $\mathbf{P} \in \mathcal{P}^{\text{TC}(3)}$*, with* $\text{TC}(\mathbf{P}) = \{1, 2, 3\} = \text{CO}(\mathbf{P}) = \text{MA}(\mathbf{P})$*. However the permutations* σ_{RC}*,* σ_{MB} *and* $\sigma_{\text{SVM-RA}}$ *produced by running RC, MB and SVM-RA on the above matrix* $\mathbf{P}$ *do not rank these tournament solution sets at the top; indeed, it can be verified that* $\sigma_{\text{RC}} = \sigma_{\text{MB}} = (4\ 1\ 2\ 3\ 6\ 5)$ *and* $\sigma_{\text{SVM-RA}} = (4\ 2\ 6\ 1\ 3\ 5)$*.*

The above examples lead us to consider alternative ranking algorithms that rank Condorcet winners, top cycles, and Copeland/Markov sets at the top even when the underlying preference matrix $\mathbf{P}$ induces a cyclic tournament. Proofs of all results can be found in the supplementary material.

5. Recovering Condorcet Winners, Top Cycles and Copeland Sets at the Top

Let us start by considering the Copeland set. Recall that this is defined as the set of items with maximal out-degree in the underlying preference-induced tournament $G_{\mathbf{P}}$. A natural approach to ranking the Copeland set at the top would therefore be to order items by their observed out-degrees in the empirical comparison matrix $\widehat{\mathbf{P}}$; as this empirical matrix approaches the true preference matrix $\mathbf{P}$, one would expect items in the Copeland set to appear at the top. The corresponding algorithm is shown in Algorithm 1; we term this the Matrix Copeland (MC) algorithm due to its similarity to the Copeland voting rule (Copeland, 1951). The following result shows this algorithm indeed ranks the Copeland set at the top for any preference matrix and also minimizes the pairwise disagreement error when the preference matrix is acyclic:

Theorem 5 (Matrix Copeland ranks Copeland set at top for a general $\mathbf{P}$ and minimizes PD error if $\mathbf{P} \in \mathcal{P}^{\text{DAG}}$)**.** *Let* $\mu_{\min} > 0$*. Let* $\mathbf{P} \in \mathcal{P}_n$ *and let* $\delta \in (0, 1]$*. If*

$$m \geq \max\left(\frac{512}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right),\ B_\mu\right),$$

then with probability at least $1-\delta$ *(over* S*), the permutation* $\widehat{\sigma}_{\text{MC}}$ *produced by running the MC algorithm on* $\widehat{\mathbf{P}}$ *satisfies*

Algorithm 1 Matrix Copeland (MC) Algorithm

Input: Pairwise comparison matrix $\overline{\mathbf{P}} \in [0, 1]^{n \times n}$ satisfying the following conditions:
(i) for all $i \neq j$: $\overline{p}_{ij} + \overline{p}_{ji} = 1$ or $\overline{p}_{ij} = \overline{p}_{ji} = 0$
(ii) for every i: $\overline{p}_{ii} = 0$

• For $i = 1$ to n: $\overline{f}_i = \frac{1}{n} \sum_{j=1}^{n} \mathbf{1}(\overline{p}_{ji} > \frac{1}{2})$

Output: Permutation $\overline{\sigma}_{\text{MC}} \in \text{argsort}(\overline{\mathbf{f}})$

$$\widehat{\sigma}_{\text{MC}}(i) < \widehat{\sigma}_{\text{MC}}(j) \quad \textit{for all } i \in \text{CO}(\mathbf{P}), j \notin \text{CO}(\mathbf{P}).$$

Moreover, if $\mathbf{P} \in \mathcal{P}^{\text{DAG}}$*, then*

$$\widehat{\sigma}_{\text{MC}} \in \text{argmin}_{\sigma \in \mathcal{S}_n} \text{er}_{\mathbf{P}}^{\text{PD}}[\sigma]$$

In fact, as the following result shows, the MC algorithm also recovers the top cycle (and therefore the Condorcet winner whenever it exists) at the top:

Theorem 6 (Matrix Copeland ranks top cycle at top)**.** *Let* $\mu_{\min} > 0$*. Let* $\mathbf{P} \in \mathcal{P}_n^{\text{TC}(k)}$ *for some* $k \in [n-1]$*. Let* $\delta \in (0, 1]$*. If*

$$m \geq \max\left(\frac{512}{\gamma_{\text{TC}}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right),\ B_\mu\right),$$

then with probability at least $1-\delta$ *(over* S*), the permutation* $\widehat{\sigma}_{\text{MC}}$ *produced by running the MC algorithm on* $\widehat{\mathbf{P}}$ *satisfies*

$$\widehat{\sigma}_{\text{MC}}(i) < \widehat{\sigma}_{\text{MC}}(j) \quad \textit{for all } i \in \text{TC}(\mathbf{P}), j \notin \text{TC}(\mathbf{P}).$$

While the MC algorithm is good for recovering the Copeland set and the top cycle or the Condorcet winner when it exists, as the following example illustrates, it *does not* necessarily rank the Markov set at the top:

Example 3 (Matrix Copeland can fail to rank Markov set at top)**.** *Let* $n = 8$*, and consider*

$$\mathbf{P} = \begin{bmatrix} 0 & 0.51 & 0.4 & 0.6 & 0.6 & 0.6 & 0.4 & 0.4 \\ 0.49 & 0 & 0.49 & 0.49 & 0.6 & 0.6 & 0.49 & 0.49 \\ 0.6 & 0.51 & 0 & 0.6 & 0.4 & 0.6 & 0.6 & 0.4 \\ 0.4 & 0.51 & 0.4 & 0 & 0.6 & 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.6 & 0.4 & 0 & 0.4 & 0.6 & 0.6 \\ 0.4 & 0.4 & 0.4 & 0.6 & 0.6 & 0 & 0.4 & 0.4 \\ 0.6 & 0.51 & 0.4 & 0.6 & 0.4 & 0.6 & 0 & 0.4 \\ 0.6 & 0.51 & 0.6 & 0.6 & 0.4 & 0.6 & 0.6 & 0 \end{bmatrix}.$$

The tournament induced by $\mathbf{P}$ *is shown in Figure 4 (right). It can be verified that* $\text{MA}(\mathbf{P}) = \{5\}$*. However the permutation* σ_{MC} *produced by running the MC algorithm on the above matrix* $\mathbf{P}$ *does not rank item 5 at the top; indeed, it can be verified that* $\sigma_{\text{MC}} = (2\ 4\ 6\ 5\ 1\ 7\ 3\ 8)$*.*

6. Recovering Condorcet Winners, Top Cycles and Markov Sets at the Top

The Matrix Copeland algorithm considered above can be viewed as running the Matrix Borda algorithm on a transformed input matrix $\overline{\mathbf{H}}$ where $\overline{h}_{ij} = \mathbf{1}(\overline{p}_{ij} > \frac{1}{2})\ \forall i \neq j$, which thresholds the preferences $\overline{p}_{ij}$ at $\frac{1}{2}$ to 'amplify' them to 0 or 1, and has the effect of emphasizing more preferred

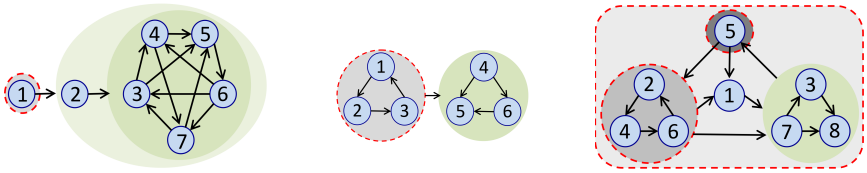

Figure 4. Tournaments induced by the preferences matrices in Examples 1–3. An edge from one group of nodes to another indicates that the graph contains edges from all nodes in the first group to all those in the second group. **Left:** Here $n = 7$ and $\mathrm{TC}(\mathbf{P}) = \mathrm{CO}(\mathbf{P}) = \mathrm{MA}(\mathbf{P}) = \{1\}$. **Middle:** Here $n = 6$ and $\mathrm{TC}(\mathbf{P}) = \mathrm{CO}(\mathbf{P}) = \mathrm{MA}(\mathbf{P}) = \{1, 2, 3\}$. **Right:** Here $n = 8$ and $\mathrm{TC}(\mathbf{P}) = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $\mathrm{CO}(\mathbf{P}) = \{2, 4, 6\}$, and $\mathrm{MA}(\mathbf{P}) = \{5\}$.

elements more strongly. As we saw above, this yielded desirable properties in terms of ranking Condorcet winners, top cycles and Copeland sets (though not Markov sets) at the top, while also maintaining the PD optimality properties of the MB algorithm in $\mathcal{P}^{\mathrm{LN}}$ (in fact, MC recovers a PD-optimal ranking for all $\mathbf{P}$ in the larger set $\mathcal{P}^{\mathrm{DAG}}$).

In order to recover Markov sets at the top, one might consider running the Rank Centrality algorithm, which ranks items according to the stationary vector associated with a certain Markov chain associated with the input matrix, on a similarly transformed matrix. Below we start by considering in Section 6.1 the Unweighted Markov (UM) ranking algorithm, which does exactly this: it runs the RC algorithm on the same transformed matrix $\overline{\mathbf{H}}$ described above. As we will see below, this algorithm indeed ranks Condorcet winners, top cycles and Markov sets (though not Copeland sets) at the top. Unfortunately, however, as we will see, the UM algorithm fails to maintain the PD optimality properties of the RC algorithm in $\mathcal{P}^{\mathrm{BTL}}$.

In Section 6.2, we propose instead the Parametrized Markov (PM) algorithm, which runs the RC algorithm on a matrix $\overline{\mathbf{H}}^c$ obtained by applying a soft transformation parametrized by a real number $c \geq 1$ to the input matrix $\overline{\mathbf{P}}$. This also amplifies the probabilities in the input matrix, but not to the extremes of 0 and 1 (the hard transform $\overline{\mathbf{H}}$ above corresponds to $c = \infty$). We will see that for suitable choices of c, the resulting PM algorithm can be made to both rank Condorcet winners, top cycles and Markov sets at the top, and maintain the PD optimality properties of the RC algorithm in $\mathcal{P}^{\mathrm{BTL}}$. As a by-product of our analysis, we also obtain an improved sample complexity bound for the RC algorithm to recover a PD-optimal ranking, thus answering an open question of Rajkumar & Agarwal (2014).

6.1. Unweighted Markov Algorithm

The UM algorithm is shown in Algorithm 2. We first show this algorithm indeed recovers the Markov set at the top:

Theorem 7 (Unweighted Markov ranks Markov set at top)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n$. Let $\delta \in (0, 1]$. If*

$$m \geq \max\left(\frac{512}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right), B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\mathrm{UM}}$ produced by running the UM algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\mathrm{UM}}(i) < \widehat{\sigma}_{\mathrm{UM}}(j) \quad \textit{for all } i \in \mathrm{MA}(\mathbf{P}), j \notin \mathrm{MA}(\mathbf{P}).$$

Algorithm 2 Unweighted Markov (UM) Algorithm

Input: Pairwise comparison matrix $\overline{\mathbf{P}} \in [0, 1]^{n \times n}$ satisfying the following conditions:
(i) for all $i \neq j$: $\overline{p}_{ij} + \overline{p}_{ji} = 1$ or $\overline{p}_{ij} = \overline{p}_{ji} = 0$
(ii) for every i: $\overline{p}_{ii} = 0$

- Define $\overline{\mathbf{H}} \in [0, 1]^{n \times n}$ as

$$\overline{h}_{ij} = \begin{cases} \mathbf{1}(\overline{p}_{ij} > \frac{1}{2}) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

- Run Rank Centrality algorithm on input matrix $\overline{\mathbf{H}}$; obtain stationary probability vector $\overline{\boldsymbol{\pi}}$

Output: Permutation $\overline{\sigma}_{\mathrm{UM}} \in \mathrm{argsort}(\overline{\boldsymbol{\pi}})$

In fact, as with the MC algorithm, the UM algorithm also recovers the top cycle (and therefore the Condorcet winner whenever it exists) at the top:

Theorem 8 (Unweighted Markov ranks top cycle at top)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n^{\mathrm{TC}(k)}$ for some $k \in [n-1]$. Let $\delta \in (0, 1]$. If*

$$m \geq \max\left(\frac{512}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right), B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\mathrm{UM}}$ produced by running the UM algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\mathrm{UM}}(i) < \widehat{\sigma}_{\mathrm{UM}}(j) \quad \textit{for all } i \in \mathrm{TC}(\mathbf{P}), j \notin \mathrm{TC}(\mathbf{P}).$$

Again, as with the MC algorithm, the following example shows that while UM is good for recovering the Markov set and the top cycle or the Condorcet winner when it exists, it *does not* necessarily rank the Copeland set at the top:

Example 4 (Unweighted Markov can fail to rank Copeland set at top)**.** *Let $n = 8$, and consider again the matrix $\mathbf{P}$ considered in Example 3. It can be verified that $\mathrm{CO}(\mathbf{P}) = \{2, 4, 6\}$. However the permutation σ_{UM} produced by running the UM algorithm on $\mathbf{P}$ does not rank this set at the top; indeed, it can be verified that $\sigma_{\mathrm{UM}} = (5\ 2\ 4\ 6\ 7\ 3\ 8\ 1)$.*

Unfortunately, however, unlike the MC algorithm, the UM algorithm fails to maintain the PD optimality properties of the RC algorithm in $\mathcal{P}^{\mathrm{BTL}}$:

Example 5 (Unweighted Markov can fail to minimize PD error in $\mathcal{P}^{\text{BTL}}$)**.** *Let $n = 3$, and consider*

$$\mathbf{P} = \begin{bmatrix} 0 & 0.4 & 0.25 \\ 0.6 & 0 & 0.33 \\ 0.75 & 0.67 & 0 \end{bmatrix}.$$

It can be verified that $\mathbf{P} \in \mathcal{P}^{\text{BTL}}$ (consider the score vector $\mathbf{w} = (3, 2, 1)^\top$), and that the PD error is uniquely minimized by $\sigma = (1\ 2\ 3)$. However the permutation σ_{UM} produced by running the UM algorithm on $\mathbf{P}$ is $\sigma_{\text{UM}} = (1\ 3\ 2)$, and therefore this does not minimize the PD error.

Next we consider an alternative algorithm that will achieve the best properties of both the UM and RC algorithms.

6.2. Parametrized Markov Algorithm

We now consider the Parametrized Markov (PM) ranking algorithm shown in Algorithm 3, which effectively applies the RC algorithm to a matrix $\overline{\mathbf{H}}^c$ obtained by applying a 'soft' transform $g_c : [0,1] \to [0,1]$, parametrized by $c \geq 1$, to the entries of the input matrix $\overline{\mathbf{P}}$:

$$g_c(p) = \frac{p^c}{p^c + (1-p)^c}.$$

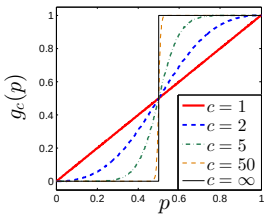

Figure 5. The function g_c.

See Figure 5 for an illustration. With $c = 1$, one recovers the RC algorithm; with $c = \infty$, one recovers the UM algorithm. The PM algorithm can therefore be viewed as interpolating between these two extremes. As we show below, with a suitable choice of c, the PM algorithm gives the best of both worlds: as with RC, it minimizes the PD error in $\mathcal{P}^{\text{BTL}}$; and as with UM, it ranks Condorcet winners, top cycles, and Markov sets at the top.

6.2.1. PM Algorithm Minimizes PD Error in $\mathcal{P}^{\text{BTL}}$

We first show that for any choice of $1 \leq c < \infty$, the PM algorithm minimizes the PD error in $\mathcal{P}^{\text{BTL}}$:

Theorem 9 (Parametrized Markov minimizes PD error in $\mathcal{P}^{\text{BTL}}$)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n^{\text{BTL}}$. Let $r_{\min} = \min_{i \neq j, k \neq i,j} |p_{ik} - p_{jk}|$. Let $\delta \in (0,1]$. Let $1 \leq c < \infty$. If*

$$m \geq \max\left(\frac{512}{\min(r_{\min}^2, \gamma_{\min}^2)\mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right), B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\text{PM}}$ produced by running the PM algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\text{PM}} \in \operatorname{argmin}_{\sigma \in \mathcal{S}_n} \text{er}_{\mathbf{P}}^{\text{PD}}[\sigma].$$

The proof of Theorem 9 in fact establishes PD-optimality of the PM algorithm (and therefore the RC algorithm, which is a special case with $c = 1$) for a slightly larger set of preference matrices than $\mathcal{P}^{\text{BTL}}$, namely, for $\mathbf{P}$ satisfying the *restricted low-noise* (RLN) property, defined as

Algorithm 3 Parametrized Markov (PM) Algorithm

Input: Pairwise comparison matrix $\overline{\mathbf{P}} \in [0,1]^{n \times n}$ satisfying the following conditions:
(i) for all $i \neq j$: $\overline{p}_{ij} + \overline{p}_{ji} = 1$ or $\overline{p}_{ij} = \overline{p}_{ji} = 0$
(ii) for every i: $\overline{p}_{ii} = 0$

Parameter: $c \geq 1$

- Define $\overline{\mathbf{H}}^c \in [0,1]^{n \times n}$ as

$$\overline{h}_{ij}^c = \begin{cases} g_c(\overline{p}_{ij}) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

- Run Rank Centrality algorithm on input matrix $\overline{\mathbf{H}}^c$; obtain stationary probability vector $\overline{\boldsymbol{\pi}}^c$

Output: Permutation $\overline{\sigma}_{\text{PM}} \in \text{argsort}(\overline{\boldsymbol{\pi}}^c)$

$$\mathcal{P}_n^{\text{RLN}} = \left\{\mathbf{P} \in \mathcal{P}^n : \forall\, i \neq j \neq k : i \succ_{\mathbf{P}} j \implies p_{kj} < p_{ki}\right\}.$$

It is not hard to show that $\mathcal{P}_n^{\text{BTL}} \subset \mathcal{P}_n^{\text{RLN}} \subset \mathcal{P}_n^{\text{LN}}$. The proof of Theorem 9 then follows from the following three observations: (a) for $\mathbf{P} \in \mathcal{P}_n^{\text{RLN}}$, running the RC algorithm on $\mathbf{P}$ yields a PD-optimal permutation; (b) for large enough sample size, if $\mathbf{P}$ is in $\mathcal{P}_n^{\text{RLN}}$, then with high probability so is $\widehat{\mathbf{P}}$; and (c) the RLN property is preserved by the g_c transform (see supplementary material for details).

Remark (Improved sample complexity bound for RC algorithm)**.** *As noted above, the sample complexity bound in Theorem 9 holds for any $1 \leq c < \infty$, and so in particular it holds for the RC algorithm. Comparing this with the previous bound of (Rajkumar & Agarwal, 2014) in Theorem 1, we see that the bound in Theorem 9 is in general considerably tighter; in particular, it removes an extraneous factor of n present in the earlier bound. This answers in the affirmative an open question of (Rajkumar & Agarwal, 2014).*

6.2.2. For Suitable c, PM Algorithm Recovers Markov Sets and Top Cycles at the Top

We now show that for sufficiently large (but finite) c, which can be chosen based on the observed empirical comparison matrix $\widehat{\mathbf{P}}$, the PM algorithm also recovers Markov sets and top cycles (and therefore Condorcet winners) at the top:

Theorem 10 (Parametrized Markov ranks Markov set at top)**.** *Let $\mu_{\min} > 0$. Let $\mathbf{P} \in \mathcal{P}_n$. Let $\beta_2(\widehat{\mathbf{P}}) = \widehat{\pi}_{\max} - \max_{k \notin \text{MA}(\widehat{\mathbf{P}})} \widehat{\pi}_k$ where $\widehat{\boldsymbol{\pi}}$ is obtained by running UM on $\widehat{\mathbf{P}}$. Let $\delta \in (0,1]$. Let $\widehat{\alpha}_{\min} = \min_{i \succ_{\mathbf{P}} j} \frac{\widehat{p}_{ji}}{\widehat{p}_{ij}}$. If $c \geq \frac{\ln(4n/\beta_2(\widehat{\mathbf{P}}))}{\ln(\widehat{\alpha}_{\min})}$ and*

$$m \geq \max\left(\frac{512}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right), B_\mu\right),$$

then with probability at least $1-\delta$ (over S), the permutation $\widehat{\sigma}_{\text{PM}}$ produced by running the PM algorithm on $\widehat{\mathbf{P}}$ satisfies

$$\widehat{\sigma}_{\text{PM}}(i) < \widehat{\sigma}_{\text{PM}}(j) \quad \textit{for all } i \in \text{MA}(\mathbf{P}), j \notin \text{MA}(\mathbf{P}).$$

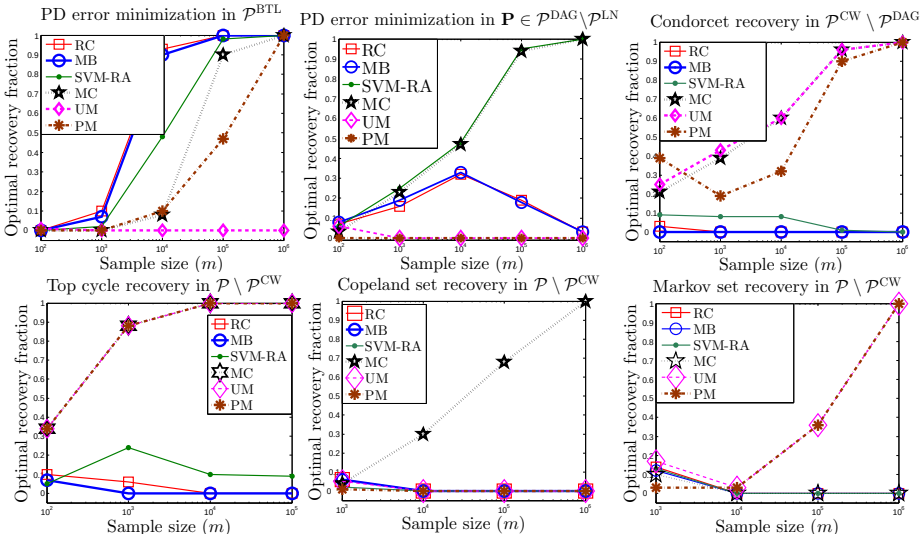

Figure 6. Experimental results. See text for details.

Theorem 11 (Parametrized Markov ranks top cycle at top)**.** *Let* $\mu_{\min} > 0$. *Let* $\mathbf{P} \in \mathcal{P}_n^{\mathrm{TC}(k)}$ *for some* $k \in [n-1]$. *Let* $\widehat{\alpha}_{\min} = \min_{i \succ_{\mathbf{P}} j} \frac{\widehat{p}_{ji}}{\widehat{p}_{ij}}$. *Let* $\delta \in (0,1]$. *If* $c \geq \frac{(n+1)\ln(4n)}{\ln(\widehat{\alpha}_{\min})}$ *and*

$$m \geq \max\left(\frac{512}{\gamma_{\min}^2 \mu_{\min}^2} \ln\left(\frac{4n^2}{\delta}\right), B_\mu\right),$$

then with probability at least $1-\delta$ *(over* S*), the permutation* $\widehat{\sigma}_{\mathrm{PM}}$ *produced by running the PM algorithm on* $\widehat{\mathbf{P}}$ *satisfies*

$$\widehat{\sigma}_{\mathrm{PM}}(i) < \widehat{\sigma}_{\mathrm{PM}}(j) \quad \text{for all } i \in \mathrm{TC}(\mathbf{P}), j \notin \mathrm{TC}(\mathbf{P}).$$

The proofs of Theorems 10–11 make use of the Cho-Meyer perturbation bound for Markov chains (Cho & Meyer, 2001) to bound the difference between the stationary vector $\boldsymbol{\pi}^c$ and $\boldsymbol{\pi}$. (see supplementary material for details).

7. Experiments

We conducted experiments to compare the performance of different algorithms, including existing algorithms (RC, MB, SVM-RA) and the proposed algorithms (MC, UM, PM), both in terms of minimizing PD error and recovering 'good' items at the top (the parameter c for PM algorithm was chosen as the maximum of the values prescribed by Theorems 10 and 11). In all our experiments, we generated 100 training samples S (as described in Section 2.1) from an underlying preference matrix $\mathbf{P}$ of interest (using a uniform distribution μ on pairs) each of a number of sizes m, ran the various algorithms on the samples, and for each algorithm, counted the fraction of times (out of 100) that the returned permutation satisfied a desired property, such as PD error minimization or top cycle recovery at the top.

PD error minimization. For this we used two acyclic preference matrices: a matrix $\mathbf{P} \in \mathcal{P}_{10}^{\mathrm{BTL}}$ generated from a score vector $\mathbf{w}$ drawn uniformly from $[0,1]^{10}$, and the matrix $\mathbf{P} \in (\mathcal{P}_5^{\mathrm{DAG}} \setminus \mathcal{P}_5^{\mathrm{LN}})$ described in (Rajkumar & Agarwal, 2014). The results are shown in the first two plots in Figure 6. As expected, for the first $\mathbf{P}$, all algorithms except UM recover an optimal ranking w.r.t. PD error, while for the second $\mathbf{P}$, only SVM-RA and MC do so.

Condorcet winner and top cycle recovery. For this we used two preference matrices with cycles: the matrix $\mathbf{P} \in \mathcal{P}_7^{\mathrm{CW}} \setminus \mathcal{P}_7^{\mathrm{DAG}}$ described in Example 1 (see also Figure 4, left), and the matrix $\mathbf{P} \in \mathcal{P}_6^{\mathrm{TC}(3)} \subset (\mathcal{P}_6 \setminus \mathcal{P}_6^{\mathrm{CW}})$ described in Example 2 (see also Figure 4, middle). The results are shown in the 3rd and 4th plots in Figure 6. In this case, only the new algorithms considered in this paper (MC, UM, PM) recover the Condorcet winner and the top cycle at the top.

Copeland set recovery. For this we used the cyclic preference matrix $\mathbf{P} \in (\mathcal{P}_8 \setminus \mathcal{P}_8^{\mathrm{CW}})$ described in Example 3 (see also Figure 4, right). The results are shown in the 5th plot in Figure 6. As expected, only the MC algorithm successfully recovers the Copeland set at the top.

Markov set recovery. For this we again used the cyclic matrix $\mathbf{P} \in (\mathcal{P}_8 \setminus \mathcal{P}_8^{\mathrm{CW}})$ described in Example 3, for which $\mathrm{MA}(\mathbf{P}) \neq \mathrm{CO}(\mathbf{P})$. The results are shown in the 6th plot in Figure 6. As expected, in this case only the UM and PM algorithms correctly recover the Markov set at the top.

8. Conclusion

In this paper, we investigated convergence properties of algorithms for ranking from pairwise preferences when preferences can contain cycles. When a globally optimal ranking is hard to compute, it is natural to ask that 'good' items, such as Condorcet winners or suitable tournament solution sets, be ranked at the top. We showed that several ranking algorithms developed in recent years fail in this respect, and designed new algorithms that rank Condorcet winners, top cycles, Copeland and Markov sets at the top, while retaining or improving guarantees of previous algorithms for acyclic preferences. Future work includes exploring alternative complexity parameters for our bounds and considering similar objectives in an active ranking setting.

Acknowledgements. Thanks to the anonymous reviewers for their helpful comments. AR is supported by a Microsoft Research India PhD Fellowship. The work of LHL is supported by AFOSR FA9550-13-1-0133, NSF DMS-1209136, and NSF DMS-1057064. SA thanks DST and Indo-US Science & Technology Forum for their support.

References

Ailon, Nir. Active learning ranking from pairwise preferences with almost optimal query complexity. In *Advances in Neural Information Processing Systems*, 2011.

Borda, Jean-Charles de. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

Brandt, Felix and Fischer, Felix. PageRank as a weak tournament solution. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, 2007.

Brandt, Felix, Brill, Markus, and Harrenstein, Paul. Tournament solutions. In *Handbook of Computational Social Choice*. Cambridge University Press, To appear, 2015.

Braverman, Mark and Mossel, Elchanan. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009.

Busa-Fekete, Róbert, Hüllermeier, Eyke, and Szörényi, Balázs. Preference-based rank elicitation using statistical models: The case of Mallows. In *Proceedings of the 31st International Conference on Machine Learning*, 2014a.

Busa-Fekete, Róbert, Szörényi, Balázs, and Hüllermeier, Eyke. PAC rank elicitation through adaptive sampling of stochastic pairwise preferences. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014b.

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

Copeland, A. H. A 'reasonable' social welfare function. In *Seminar on Mathematics in Social Sciences, University of Michigan*, 1951.

De Donder, Philippe, Le Breton, Michel, and Truchon, Michel. Choosing from a weighted tournament. *Mathematical Social Sciences*, 40(1):85–109, 2000.

Fürnkranz, Johannes and Hüllermeier, Eyke. *Preference Learning*. Springer, 2010.

Jamieson, Kevin G. and Nowak, Rob. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, 2011.

Jiang, Xiaoye, Lim, Lek-Heng, Yao, Yuan, and Ye, Yinyu. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

Laslier, J.-F. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.

Lu, Tyler and Boutilier, Craig. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Moulin, Hervé. Choosing from a tournament. *Social Choice and Welfare*, 3(4):271–291, 1986.

Negahban, Sahand, Oh, Sewoong, and Shah, Devavrat. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, 2012.

Osting, Braxton, Brune, Christoph, and Osher, Stanley. Enhanced statistical rankings via targeted data collection. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Rajkumar, Arun and Agarwal, Shivani. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Seneta, E. Perturbation of the stationary distribution measured by ergodicity coefficients. *Advances in Applied Probability*, pp. 228–230, 1988.

Smith, John H. Aggregation of preferences with variable electorate. *Econometrica: Journal of the Econometric Society*, pp. 1027–1041, 1973.

Wauthier, Fabian L., Jordan, Michael I., and Jojic, Nebojsa. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.