The Limits of Maxing, Ranking, and Preference Learning

Moein Falahatgar [1] **Ayush Jain** [1] **Alon Orlitsky** [1] **Venkatadheeraj Pichapati** [1] **Vaishakh Ravindrakumar** [1]

Abstract

We present a comprehensive understanding of three important problems in PAC preference learning: maximum selection (maxing), ranking, and estimating *all* pairwise preference probabilities, in the adaptive setting. With just Weak Stochastic Transitivity, we show that maxing requires $\Omega(n^2)$ comparisons and with slightly more restrictive Medium Stochastic Transitivity, we present a linear complexity maxing algorithm. With Strong Stochastic Transitivity and Stochastic Triangle Inequality, we derive a ranking algorithm with optimal $\mathcal{O}(n \log n)$ complexity and an optimal algorithm that estimates all pairwise preference probabilities.

1. Introduction

1.1. Background and motivation

Maximum selection (maxing) and sorting (ranking) are fundamental problems in Computer Science with numerous important applications. Deterministic versions of these problems are well studied.

In practical applications, comparisons are rarely deterministic. For example in soccer, when Real Madrid plays Barcelona the outcome is not always the same. Similarly, individual preferences in restaurants vary a lot. Other practical applications are in areas such as social choice (Caplin & Nalebuff, 1991; Soufiani et al., 2013), web search and information retrieval (Radlinski & Joachims, 2007; Radlinski et al., 2008), crowdsourcing (Chen et al., 2013; gif), recommender systems (Baltrunas et al., 2010) and several others.

These practical applications and the intrinsic theoretical interest, has led to significant work on the probabilistic version of maxing and ranking. Yet the most general model for which maxing can be done using near-linear comparisons is not known. We consider the most general transitive model that guarantees the existence of maximum and show that under this model any maxing algorithm requires quadratic many comparisons. We also consider a slightly more restrictive transitive model and propose a linear complexity maxing algorithm, making it the most general model known for which linear complexity maxing is possible. Also, for the most general known model with sub-quadratic complexity for ranking, we improve the complexity, making it orderwise optimal. We also propose an optimal algorithm that can simulate all pairwise comparisons.

1.2. Notation and problem formulation

Without loss of generality, let $[n] \stackrel{\text{def}}{=} \{1, 2, ..., n\}$ be the set of n elements. We consider probabilistic noisy comparisons i.e., whenever two elements i and j are compared, i is returned with an unknown probability $p_{i,j}$. There are no "ties" i.e., $p_{j,i} = 1 - p_{i,j}$. Let $\tilde{p}_{i,j} \stackrel{\text{def}}{=} p_{i,j} - \frac{1}{2}$ be the centered preference probability.

A maximal is an element i that is preferable to every other element i.e., $\tilde{p}_{i,j} \geq 0 \ \forall j$. A ranking is a permutation $\sigma_1, \sigma_2, ..., \sigma_n$ of $[n]$ such that $\tilde{p}_{\sigma_i, \sigma_j} \geq 0$ whenever $i > j$.

But sometimes maximal and ranking might not even exist. For example, consider the popular Rock-Paper-Scissor game i.e., $p_{1,2} = p_{2,3} = p_{3,1} = 1$. Notice that under this model there is neither a maximal nor a ranking. Hence we need additional constraints on pairwise probabilities $p_{i,j}$.

Notice that for ranking to exist, there must exist an ordering ($\succ$) among elements s.t. whenever $i \succ j$, $\tilde{p}_{i,j} \geq 0$. The models that have such an ordering are said to satisfy *Weak Stochastic Transitivity (WST)*. Observe that WST is sufficient for existence of both maximal and ranking.

More restrictive notions of transitivity are motivated and used in different contexts. *Strong Stochastic Transitivity (SST)* which assumes that whenever $i \succ j \succ k$, $\tilde{p}_{i,k} \geq \max(\tilde{p}_{i,j}, \tilde{p}_{j,k})$, as its name suggests is a stronger notion of transitivity that confines the model more than WST, hence less general. *Medium Stochastic Transitivity (MST)* (Skorepa, 2010) sitting in between WST and SST, assumes that whenever $i \succ j \succ k$, $\tilde{p}_{i,k} \geq \min(\tilde{p}_{i,j}, \tilde{p}_{j,k})$. From WST to MST to SST, the model becomes more restrictive.

Another model restriction used in some of the previous

[1]University of California, San Diego. Correspondence to: Venkatadheeraj Pichapati <dheerajpv7@ucsd.edu>.

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

works *Stochastic Triangle Inequality (STI)*, assumes that whenever $i \succ j \succ k$, $\tilde{p}_{i,k} \leq \tilde{p}_{i,j} + \tilde{p}_{j,k}$. In this paper we propose maxing and ranking algorithms for models under various set of constraints.

There is also a concern with finding an exact maximal and ranking. Consider the case of $n = 2$ and $\tilde{p}_{1,2} \approx 0$. Notice that in this case where n is just 2, finding maximal and ranking could take arbitrarily many comparisons. Easy fix to alleviate this problem is to consider *Probably Approximately Correct (PAC)* formulation which we also adopt.

An element i is said to be ϵ-preferable to j if $\tilde{p}_{i,j} \geq -\epsilon$. For $\epsilon \in (0, 1/2)$, an ϵ-maximal is an element i that is ϵ-preferable to all elements i.e., $\tilde{p}_{i,j} \geq -\epsilon \ \forall j$. Given $0 < \epsilon < 1/2$, $0 < \delta \leq 1/2$, a PAC maxing algorithm must output an ϵ-maximal with probability $\geq 1-\delta$. Similarly, an ϵ-ranking is a permutation $\sigma_1, \sigma_2, ..., \sigma_n$ of $[n]$ such that σ_i is ϵ-preferable to σ_j whenever $i > j$. Given $0 < \epsilon < 1/2$, $0 < \delta \leq 1/2$, a PAC ranking algorithm must output an ϵ-ranking with probability $\geq 1 - \delta$.

1.3. Related work

Researchers initially considered more restrictive models. (Feige et al., 1994) considered constant noise model i.e., $\tilde{p}_{i,j} = \alpha > 0$ if $i \succ j$ and presented a maxing algorithm that uses $\mathcal{O}\big(\frac{n}{\alpha^2} \log \frac{1}{\delta}\big)$ comparisons and outputs maximal with probability $\geq 1 - \delta$. It also presented a ranking algorithm that uses $\mathcal{O}\Big(\frac{n \log n}{\alpha^2}\Big)$ comparisons and outputs ranking with probability $\geq 1 - 1/n$.

Another set of widely-studied restrictive models are parametric ones. (Szörényi et al., 2015) considered one of the most popular parametric models, Plackett-Luce (Plackett, 1975; Luce, 2005) and presented PAC maxing and ranking algorithms that use $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{n}{\epsilon\delta}\big)$ and $\mathcal{O}\Big(\frac{n \log n}{\epsilon^2} \log \frac{n}{\epsilon\delta}\Big)$ comparisons respectively.

Researchers also considered models that are more general than parametric models, yet still more restrictive than WST. (Yue & Joachims, 2011) considered models that satisfy both SST and STI and derived a PAC maxing algorithm that uses $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{n}{\epsilon\delta}\big)$ comparisons. Later (Falahatgar et al., 2017b) considered same model and proposed an optimal PAC maxing algorithm that uses $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\big)$ comparisons. It also proposed a PAC ranking algorithm that with probability $\geq 1 - 1/n$, outputs an ϵ-ranking using $\mathcal{O}\Big(\frac{n \log n (\log \log n)^3}{\epsilon^2}\Big)$ comparisons, $(\log \log n)^3$ times the known lower bound. Until now, it was not known if the additional $(\log \log n)^3$ factor is necessary for PAC ranking.

(Falahatgar et al., 2017a) considered models that satisfy only SST but not necessarily STI and proposed an optimal PAC maxing algorithm that uses $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\big)$ comparisons. They also showed that there exists a model which satisfies SST and yet no algorithm can find an ϵ-ranking for this model using $o(n^2)$ comparisons, establishing a lower bound of $\Omega(n^2)$ comparisons once STI property is dropped.

Among other related works we can point out (Busa-Fekete et al., 2014b; Lee et al., 2014; Dudík et al., 2015; Hüllermeier et al., 2008), who considered models more general than WST under different definitions of maximum and ranking. More discussion about these models can be found in Appendix G. (Busa-Fekete et al., 2014a; Mohajer et al., 2017) considered the non-PAC version and (Rajkumar & Agarwal, 2014; Negahban et al., 2012; 2016; Jang et al., 2016) considered the non-adaptive version of this problem. Also (Acharya et al., 2016; Ajtai et al., 2015) considered the deterministic adversarial version of maxing and ranking. (Shah et al., 2016b; Chatterjee et al., 2015; Shah et al., 2016a) studied the problem of estimating pairwise probabilities in non-adaptive setting.

2. New results and Outline

Maxing Linear-complexity maxing algorithm under SST by (Falahatgar et al., 2017a) encourages the search for a linear-complexity maxing algorithm for models with only WST properties. Two questions then arise: 1a) Is a linear complexity PAC maxing algorithm possible for models with only WST property? 1b) If not, does there exist a model more general than SST and less general than WST for which a linear complexity PAC maxing is possible?

We resolve both questions in this paper: 1a) No. Theorem 1 in Section 3 shows that there are WST models for which any PAC maxing algorithm requires $\Omega(n^2)$ comparisons. 1b) Yes. In Theorem 8 in Section 4, we derive a PAC maxing algorithm for MST model that uses $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\big)$ comparisons for $\delta \geq \min(1/n, e^{-n^{1/4}})$.

Ranking Motivated by the previous results of ranking under SST + STI, three questions arise: 2a) For models with SST + STI, is the additional $(\log \log n)^3$ factor necessary for PAC ranking algorithms? 2b) Since the near-linear complexity of ranking under SST + STI changes to quadratic complexity by dropping STI (Falahatgar et al., 2017a), is there a sub-quadratic algorithm for ranking under MST + STI? 2c) For models with SST + STI, since PAC ranking is possible with near linear complexity, is it also possible to approximate all pairwise probabilities to accuracy of ϵ using near linear number of comparisons?

We essentially resolve all three questions. 2a) No. In Theorem 9 in Section 5, we improve the PAC ranking algorithm for models with SST + STI removing additional $(\log \log n)^3$ factor and hence making it optimal. 2b) No. Theorem 10 in Section 6 shows that there is a model with MST+STI, for which any PAC ranking algorithm requires

Model	Maxing	Ranking	Finding $p_{i,j}$
SST with STI	$\Theta\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$ (Falahatgar et al., 2017b)	$\Theta\left(\frac{n\log n}{\epsilon^2}\right)^*$ Section 5	$\Theta\left(\frac{n\min(n,1/\epsilon)\log n}{\epsilon^2}\right)^*$ Section 7
SST	$\Theta\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$ (Falahatgar et al., 2017a)	$\Omega(n^2)$ (Falahatgar et al., 2017a)	$\Omega(n^2)$
MST with STI and MST	$\Theta\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)^{**}$ Section 4	$\Omega(n^2)$ Section 6	$\Omega(n^2)$
WST with STI and WST	$\Omega(n^2)$ Section 3	$\Omega(n^2)$ Section 6	$\Omega(n^2)$

Table 1. Comprehensive results for maxing, ranking and finding $p_{i,j}$
$*$: for $\delta \geq \frac{1}{n}$, $**$: for $\delta \geq \min(1/n, e^{-n^{1/4}})$

$\Omega(n^2)$ comparisons. 2c) Yes. For models with SST + STI, in Theorems 11 and 12 in Sections 7, we present an optimal algorithm that uses $\mathcal{O}\left(\frac{n\min(n,1/\epsilon)\log n}{\epsilon^2}\right)$ comparisons and approximates all pairwise probabilities to accuracy of ϵ with probability $\geq 1 - 1/n$.

We present experiments over simulated data in Section 8 and end with our conclusions in Section 9.

Interpretation Table 1 summarizes all known results for problems of maxing, ranking, and finding pairwise probabilities under different transitive properties. Notice that under the most general model WST, all these problems require quadratic many comparisons and under the most restrictive model SST + STI, all problems have optimal algorithms with near-linear complexity. For MST and WST models adding STI property does not influence complexity for any problem. But for SST model adding STI property facilitates near-linear complexity algorithms for PAC ranking and approximating pairwise probabilities.

It is easy to see that once all pairwise probabilities are approximated to accuracy of $\epsilon/2$, one can find an ϵ-maximum and an ϵ-ranking. Hence approximating pairwise probabilities is harder than PAC ranking and lower bound for PAC ranking implies a lower bound for problem of approximating pairwise probabilities. Therefore in Table 1 lower bounds for finding p_{ij} follow from lower bounds for ranking. Further in Appendix B.1, under WST model, we present a trivial algorithm that with probability $\geq 1 - \delta$, estimates all pairwise probabilities to accuracy of ϵ using $\mathcal{O}\left(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta}\right)$ comparisons. Hence upper bound of $\mathcal{O}\left(\frac{n^2}{\epsilon^2}\log\frac{n}{\delta}\right)$ follows for all problems.

3. PAC maxing for WST

We show the lower bound of $\Omega(n^2)$ for maxing under WST by presenting an example for which any algorithm requires $\Omega(n^2)$ comparisons to output a $1/4$-maximum for $\delta \leq 1/8$.

To establish the lower bound, we reduce the problem of finding a $1/4$-maximum to finding the left most piece of a linear jigsaw puzzle. We consider the following model with n elements $S = \{a_1, a_2, \ldots, a_n\}$: $\tilde{p}_{a_i,a_{i+1}} = \frac{1}{2}\forall i < n$, and $\tilde{p}_{a_i,a_j} = \mu(0 < \mu < 1/n^{10}), \forall j > i+1$. This model satisfies WST since there exists an underlying order $\succ$, $a_i \succ a_j$ if $i < j$ (because $\tilde{p}_{a_i,a_j} > 0$) and a_1 is the only $1/4$-maximum under this model.

Observe that a_i is always preferred to a_{i+1}, but for every non consecutive pair, comparison output is almost a fair coin flip. We make the problem simpler by giving the extra information of whether two non consecutive elements are being compared. Notice that this only makes the problem easier, namely, complexity for modified problem is smaller than that of original problem.

The modified problem is similar to a linear jigsaw puzzle where if we compare two pieces we will know if pieces are adjacent or not and if adjacent, which piece is on the left, the goal is to find the left most piece. We show that w.h.p., any algorithm neither finds more than $n/32$ connections (a set of neighbors) nor asks $\Omega(n)$ comparisons for the left most piece. We use this to show the lower bound. The proof is in Appendix A.

Theorem 1. *There exists a model that satisfies WST for which any algorithm requires $\Omega(n^2)$ comparisons to find a $1/4$-maximum with probability $\geq 7/8$.*

4. PAC maxing for MST

Outline In this section, we propose OPT-MAX, a linear complexity maxing algorithm for MST. In the process, we present two other suboptimal maxing algorithms SOFT-SEQ-ELIM, NEAR-OPT-MAX and use them as building blocks in OPT-MAX. SOFT-SEQ-ELIM finds an ϵ-maximum with quadratic complexity. Its performance depends on the starting element (anchor). NEAR-OPT-MAX

first finds a good anchor and then uses SOFT-SEQ-ELIM, guaranteeing near linear comparison complexity. OPT-MAX builds on NEAR-OPT-MAX and finds an ϵ-maximum in linear-complexity for $\delta \geq \min(1/n, e^{-n^{1/4}})$.

4.1. SOFT-SEQ-ELIM

Before presenting SOFT-SEQ-ELIM, we first present the subroutine COMPARE we use to compare two elements.

COMPARE COMPARE takes 5 parameters : two elements i, j that need to be compared, lower bias ϵ_l, upper bias ϵ_u, confidence δ and deems if $\tilde{p}_{i,j} < \epsilon_l$ or $\tilde{p}_{i,j} > \epsilon_u$. It compares i and j for $\frac{8}{(\epsilon_u-\epsilon_l)^2}\log\frac{2}{\delta}$ times. Let $\hat{p}_{i,j}$ be the fraction of times i won and $\hat{\tilde{p}}_{i,j} = \hat{p}_{i,j} - 1/2$. If $\hat{\tilde{p}}_{i,j} < \frac{3\epsilon_l}{4} + \frac{\epsilon_u}{4}$, then COMPARE deems $\tilde{p}_{i,j} < \epsilon_l$ (returns 1), if $\hat{\tilde{p}}_{i,j} > \frac{\epsilon_l}{4} + \frac{3\epsilon_u}{4}$, then COMPARE deems $\tilde{p}_{i,j} > \epsilon_u$ (returns 3) and for other ranges of $\hat{\tilde{p}}_{i,j}$, COMPARE not able to take a decision, returns 2.

Lemma 2 bounds comparisons used by COMPARE and proves its correctness. COMPARE and its analysis is presented in Appendix C.2.

Lemma 2. *For $\epsilon_u > \epsilon_l$, COMPARE$(i, j, \epsilon_l, \epsilon_u, \delta)$ uses $\leq \frac{8}{(\epsilon_u-\epsilon_l)^2}\log\frac{2}{\delta}$ comparisons and if $\tilde{p}_{i,j} < \epsilon_l$, then w.p.$\geq 1-\delta$, it returns 1, else if $\tilde{p}_{i,j} > \epsilon_u$, w.p.$\geq 1-\delta$, it returns 3. Further if $\tilde{p}_{i,j} \leq (\epsilon_l + \epsilon_u)/2$, w.p.$\geq 1-\delta$, it does not return 3 and if $\tilde{p}_{i,j} > (\epsilon_l + \epsilon_u)/2$, w.p.$\geq 1-\delta$, it does not return 1.*

SOFT-SEQ-ELIM SOFT-SEQ-ELIM takes 5 parameters: input set S, starting anchor element r, lower bias ϵ_l, upper bias ϵ_u and confidence δ. SOFT-SEQ-ELIM happens in rounds. In each round, it compares the current anchor a with remaining elements one by one using COMPARE. Due to probabilistic nature, we cannot exactly compare if $\tilde{p}_{e,a} > \epsilon_u$ vs $\tilde{p}_{e,a} \leq \epsilon_u$. Hence we compare if $\tilde{p}_{e,a} > \epsilon_u$ vs $\tilde{p}_{e,a} < \epsilon_l$. For an element e, if COMPARE deems $\tilde{p}_{e,a} < \epsilon_l$, then SOFT-SEQ-ELIM eliminates that element and if COMPARE deems $\tilde{p}_{e,a} > \epsilon_u$, then SOFT-SEQ-ELIM updates current anchor element to e and eliminates a. This process is continued until the current anchor element is not updated after comparing with all remaining elements and then SOFT-SEQ-ELIM outputs final anchor element.

If $\tilde{p}_{e,a} < \epsilon_l$ or $\tilde{p}_{e,a} > \epsilon_u$, COMPARE deems correctly. If $\epsilon_l \leq \tilde{p}_{e,a} \leq \epsilon_u$, then COMPARE can sometimes fail to output any decision and in that case, SOFT-SEQ-ELIM neither eliminates that element nor updates the anchor element, it just moves to next remaining element in S.

Theoretically, performance of SOFT-SEQ-ELIM strongly depends on the starting anchor element r. To define a good anchor element, similar to (Falahatgar et al., 2017a), an element a is called an (ϵ, m)-*good anchor* if a is not ϵ-preferable to at most m elements, i.e., $|\{e : e \in S \text{ and } \tilde{p}_{e,a} > \epsilon\}| \leq m$. We show that every element for which initial anchor r is ϵ_l-preferable is deemed bad and gets eliminated after its first comparison round and hence comparisons spent on all such elements is $\mathcal{O}(|S|)$. Since initial anchor r is an (ϵ_l, m)-*good anchor* element, there are only m elements for which r is not ϵ_l-preferable. We later show that only these elements can become anchors, leading to at most m changes of anchors. Therefore each such element gets compared in at most m rounds and hence we can bound total comparison rounds by $\mathcal{O}(|S| + m^2)$. Lemma 3 bounds comparisons used by SOFT-SEQ-ELIM and proves its correctness. Proof is in Appendix C.3.

Algorithm 1 SOFT-SEQ-ELIM

inputs
  Set S, element r, lower bias ϵ_l, upper bias ϵ_u, confidence δ
$Q = S \setminus \{r\}$
while $Q \neq \emptyset$ **do**
  $r' = r$, $Q' = \emptyset$
  for $c \in Q$ **do**
    $k =$ COMPARE$(c, r, \epsilon_l, \epsilon_u, \frac{2\delta}{|S|^2})$
    if $k == 1$ **then**
      $Q' = Q' \bigcup \{c\}$.
    else if $k == 3$ **then**
      $r \leftarrow c$
      $Q' = Q' \bigcup \{c\}$
      break
    end if
  end for
  if $r == r'$ **then**
    break
  end if
  $Q = Q \setminus Q'$
end while
return r

Lemma 3. *If r is an (ϵ_l, m)-good anchor element, w.p.$\geq 1-\delta$, SOFT-SEQ-ELIM$(S, r, \epsilon_l, \epsilon_u, \delta)$ uses $\mathcal{O}\left(\frac{|S|+m^2}{(\epsilon_u-\epsilon_l)^2}\log\frac{|S|}{\delta}\right)$ comparisons and outputs $\hat{r}$, an ϵ_u maximum of S, such that either $\hat{r} = r$ or $\tilde{p}_{\hat{r},r} > \frac{\epsilon_l+\epsilon_u}{2}$.*

Corollary 4 bounds comparisons used by SOFT-SEQ-ELIM for any starting anchor. Proof follows from Lemma 3

Corollary 4. *For any r, w.p.$\geq 1-\delta$, SOFT-SEQ-ELIM$(S, r, \epsilon_l, \epsilon_u, \delta)$ uses $\mathcal{O}\left(\frac{|S|^2}{(\epsilon_u-\epsilon_l)^2}\log\frac{|S|}{\delta}\right)$ comparisons and outputs $\hat{r}$, an ϵ_u maximum of S, such that either $\hat{r} = r$ or $\tilde{p}_{\hat{r},r} > \frac{\epsilon_l+\epsilon_u}{2}$.*

Now we build on SOFT-SEQ-ELIM and propose a near linear algorithm NEAR-OPT-MAX.

4.2. NEAR-OPT-MAX

NEAR-OPT-MAX(S, ϵ, δ) w.p.$\geq 1-\delta$, uses $\mathcal{O}\left(\frac{|S|}{\epsilon^2}\left(\log\frac{|S|}{\delta}\right)^2\right)$ comparisons and outputs an ϵ-maximum of S.

Since complexity of SOFT-SEQ-ELIM depends on the initial anchor element, if we can pick a good initial anchor element, then we can reduce the number of comparisons. One way to pick a good initial anchor element is to find an $\epsilon/2$-maximum of a randomly picked subset.

Lemma 5 shows that an ϵ-maximum of a randomly picked subset is a good anchor element. Proof in Appendix C.4.

Lemma 5. *If r is an ϵ-maximum of a set Q, formed by picking m elements randomly from S, then w.p.$\geq 1-\delta$, r is an $\left(\epsilon, \frac{|S|}{m}\log\frac{|S|}{\delta}\right)$-good anchor element of S.*

NEAR-OPT-MAX(S, ϵ, δ) first picks a random subset Q of size $\sqrt{|S|\log\frac{4|S|}{\delta}}$ and uses SOFT-SEQ-ELIM to find an $\epsilon/2$-maximum of Q.

By Lemma 5, w.p.$\geq 1-\delta/4$, an $\epsilon/2$-maximum of Q will be an $(\epsilon/2, \sqrt{|S|\log\frac{4|S|}{\delta}})$-*good anchor* element. NEAR-OPT-MAX then uses SOFT-SEQ-ELIM with $\epsilon/2$-maximum of Q as initial anchor to find an ϵ-maximum of S. Since the initial anchor is provably good, we are able to bound the comparisons.

Algorithm 2 NEAR-OPT-MAX

inputs
  Set S, bias ϵ, confidence δ
Form a set Q by selecting $\sqrt{|S|\log\frac{4|S|}{\delta}}$ random elements from S without replacement.
$a \leftarrow$ random element from Q, $Q = Q \setminus \{a\}$
$r \leftarrow$ SOFT-SEQ-ELIM$\left(Q, a, 0, \frac{\epsilon}{2}, \frac{\delta}{4}\right)$, $S = S \setminus \{r\}$
return SOFT-SEQ-ELIM$(S, r, \epsilon/2, \epsilon, \delta/2)$

Lemma 6 bounds the comparisons used by NEAR-OPT-MAX and proves its correctness.

Lemma 6. *With probability $\geq 1-\delta$, NEAR-OPT-MAX(S, ϵ, δ) uses $\mathcal{O}\left(\frac{|S|}{\epsilon^2}\left(\log\frac{|S|}{\delta}\right)^2\right)$ comparisons and outputs an ϵ-maximum of S.*

We build on NEAR-OPT-MAX and derive an optimal algorithm for $\delta \geq \min(1/|S|, e^{-|S|^{1/4}})$.

4.3. Optimal linear Algorithm

We first present an algorithm that is optimal for low ranges of δ i.e., $\min(e^{-|S|^{1/4}}, 1/|S|) \leq \delta \leq \frac{1}{|S|^{1/3}}$.

4.3.1. LOW RANGES OF δ

We first find a good anchor, this time using NEAR-OPT-MAX and then use SOFT-SEQ-ELIM with NEAR-OPT-MAX output as initial anchor.

OPT-MAX-LOW picks a random subset of size $|S|^{3/4}$ and finds an $\epsilon/2$-maximum of this set using NEAR-OPT-MAX. We later show that output is an $(\epsilon/2, \mathcal{O}(\sqrt{|S|}))$-*good anchor* element of S. OPT-MAX-LOW then uses SOFT-SEQ-ELIM with the previous output as initial anchor to find an ϵ-maximum of S. Since initial anchor is good, we are able to bound comparisons used by OPT-MAX-LOW.

Observe that in OPT-MAX-LOW, we call SOFT-SEQ-ELIM three times in total: two times during NEAR-OPT-MAX and once to produce the final output. Each successive call of SOFT-SEQ-ELIM acts on higher size, namely first we find $\epsilon/4$-maximum in a small set and using this element as anchor, then we find $\epsilon/2$-maximum in a larger set and finally using this new element as anchor, we find an ϵ-maximum of the whole set S.

Algorithm 3 OPT-MAX-LOW

inputs
  Set S, bias ϵ, confidence δ
Form a set Q by selecting $|S|^{3/4}$ random elements from S without replacement
$r \leftarrow$ NEAR-OPT-MAX$(Q, \frac{\epsilon}{2}, \frac{\delta}{3})$
return SOFT-SEQ-ELIM$(S, r, \frac{\epsilon}{2}, \epsilon, \frac{\delta}{3})$

Lemma 7 bounds comparisons used by OPT-MAX-LOW and proves its correctness. Proof is in Appendix C.6.

Lemma 7. *For $\frac{1}{|S|^{1/3}} \geq \delta \geq \min(1/|S|, e^{-|S|^{1/4}})$, w.p.$\geq 1-\delta$, OPT-MAX-LOW$(S, \epsilon, \delta)$ uses $\mathcal{O}(\frac{|S|}{\epsilon^2}\log\frac{1}{\delta})$ comparisons and outputs r, an ϵ-maximum*

4.3.2. HIGHER RANGES OF CONFIDENCE δ

For low ranges of confidence δ $\left(\delta \leq \frac{1}{|S|^{1/3}}\right)$, notice that $\log\frac{1}{\delta}$ and $\log\frac{|S|}{\delta}$ are of same order and hence if we use SOFT-SEQ-ELIM with a good anchor, we can guarantee complexity of $\mathcal{O}\left(\frac{|S|}{\epsilon^2}\log\frac{|S|}{\delta}\right) = \mathcal{O}\left(\frac{|S|}{\epsilon^2}\log\frac{1}{\delta}\right)$.

However, for high values of δ, this is not the case. We solve this problem by pruning S to a smaller set of size $|S|/\log|S|$ such that it contains all good elements and then use SOFT-SEQ-ELIM. Due to space constraint, we present PRUNE, the pruning algorithm, OPT-MAX-MEDIUM, and OPT-MAX-HIGH, linear complexity maxing algorithms for higher ranges of confidence in Appendix C.8.

4.4. Full Algorithm

In Theorem 8 we bound comparisons used by OPT-MAX and prove its correctness. Proof follows from Lemmas 7

Algorithm 4 OPT-MAX

inputs
Set S, bias ϵ, confidence δ
if $\delta \leq \frac{1}{|S|^{1/3}}$ **then**
return OPT-MAX-LOW(S, ϵ, δ)
end if
if $\delta \leq \frac{1}{\log |S|}$ **then**
return OPT-MAX-MEDIUM(S, ϵ, δ)
end if
return OPT-MAX-HIGH(S, ϵ, δ)

and corresponding Lemmas 19 and 20 for OPT-MAX-MEDIUM and OPT-MAX-HIGH given in Appendix C.8.

Theorem 8. *For* $\delta \geq \min(1/|S|, e^{-|S|^{1/4}})$, *w.p.*$\geq 1-\delta$, OPT-MAX$(S, \epsilon, \delta)$ *uses* $\mathcal{O}\Big(\frac{|S|}{\epsilon^2} \log \frac{1}{\delta}\Big)$ *comparisons and outputs an* ϵ*-maximum of* S.

5. Ranking for SST+STI

(Falahatgar et al., 2017b) provides a ranking algorithm that w.p.$\geq 1 - 1/|S|$, uses $\mathcal{O}\Big(\frac{|S|}{\epsilon^2} \log |S| (\log \log |S|)^3\Big)$ comparisons and outputs an ϵ-ranking of input set S.

We build on their algorithm BINARY-SEARCH-RANKING, improving two components which lead to additional $(\log \log |S|)^3$ factor, thereby proposing an optimal ϵ-ranking algorithm that uses $\mathcal{O}\Big(\frac{|S|}{\epsilon^2} \log |S|\Big)$ comparisons.

In Appendix 5, we outline the algorithm proposed in (Falahatgar et al., 2017b), pointing out the two components that lead to additional factor, and present ideas that improve over these components. For detailed explanation of BINARY-SEARCH-RANKING we refer readers to (Falahatgar et al., 2017b). Now we explain the high-level idea of how we improve over these components.

The two components that we improve upon share the property that each is being called for $\Omega(|S|/(\log |S|)^3)$ times and at each time finds a correct output w.p.$\geq 1 - 1/|S|^5$.

Instead of finding a correct output w.p.$\geq 1 - 1/|S|^5$ in one shot, and incurring high complexity, we propose the following. First use the component to find a correct output w.p.$\geq 1 - 1/\log |S|$, then check if the output is correct or not. If the output is deemed to be not correct, run the component again, finding a correct output w.p.$\geq 1 - 1/|S|^6$.

Thus to show the potency of this idea, it suffices to show: One, the second run is only invoked a few times and two, the complexity of checking whether an output is correct is not high. Our main contribution is RANK-CHECK algorithm that checks if an ordered set is ϵ-ranked or not 3ϵ-ranked. We present RANK-CHECK in Appendix D.3

Theorem 9. BINARY-SEARCH-RANKING(S, ϵ) *(Falahatgar et al., 2017b) with new improved components presented here, w.p.*$\geq 1 - 1/|S|$, *uses* $\mathcal{O}\Big(\frac{|S| \log |S|}{\epsilon^2}\Big)$ *comparisons and outputs an* ϵ*-ranking of* S.

6. Lower bound for ranking for MST+STI

In this section we show that there exists a model with both MST and STI properties under which any PAC ranking algorithm requires quadratic many comparisons. Consider the model $S = \{a_1, a_2, ..., a_n\}$ s.t. a_1 is preferable to a_2 i.e., $\tilde{p}_{a_1,a_2} = 1/2$ and comparison between any other pair is almost a fair coin flip i.e., $\tilde{p}_{a_i,a_j} = \mu \ \forall i < j$ and $\{i, j\} \neq \{1, 2\}$ for some $\mu < 1/n^{10}$. This model satisfies both MST and STI. Any permutation which has a_1 coming after a_2 is a $1/4$-ranking. But since comparison between any pair other than (a_1, a_2) is essentially a fair coin toss, any strategy that does not compare a_1 and a_2 will not have them in correct order in the output w.p.$\approx 1/2$ and hence won't be a $1/4$-ranking. Therefore this problem is similar to finding a single biased coin among $\binom{n}{2}$ coins which needs $\Omega(n^2)$ comparisons.

Theorem 10 bounds the complexity required for ϵ-ranking of models with MST and STI. Proof is in Appendix E.

Theorem 10. *There exists a model with MST and STI properties for which any algorithm requires* $\Omega(n^2)$ *comparisons to output a* $1/4$*-ranking w.p.*$\geq 7/8$.

7. Finding pairwise probabilities for SST+STI

Theorem 9 shows that for a model satisfying both SST and STI, an ϵ-ranking can be found using $\mathcal{O}\Big(\frac{|S| \log |S|}{\epsilon^2}\Big)$ comparisons. In this section we answer the question whether under same model we can approximate all pairwise probabilities to accuracy of ϵ using almost same complexity.

We first show a lower bound of $\Omega\Big(\frac{|S| \min(|S|, 1/\epsilon)}{\epsilon^2} \log |S|\Big)$ utilizing a model for which $\Omega(|S| \min(|S|, 1/\epsilon))$ pairwise probabilities need to be approximated using comparisons. Later we present APPROX-PROB that uses comparisons only for $\mathcal{O}(|S| \min(|S|, 1/\epsilon))$ pairs and hence obtain orderwise same upper bound as lower bound.

7.1. Lower Bound

We show that any algorithm requires $\Omega\Big(\frac{|S| \min(|S|, 1/\epsilon) \log |S|}{\epsilon^2}\Big)$ comparisons to approximate all pairwise probabilities to ϵ accuracy.

We prove the lower bound by using the model: $(4k+4)\epsilon \leq \tilde{p}_{a_{i+k},a_i} \leq (4k+8)\epsilon$ for $1 \leq k \leq \min(n-i, \lfloor \frac{1}{16\epsilon} - 2 \rfloor)$ and $\tilde{p}_{a_{i+k},a_i} = 1/4$ for $k > \min(n-i, \lfloor \frac{1}{16\epsilon} - 2 \rfloor)$.

It can be shown that this model satisfies both SST and STI.

Under this model, the only way to approximate unfixed pairwise probabilities is by comparing those pairs. Since pairwise probabilities are not fixed for $\Omega(n\min(n,1/\epsilon))$ pairs, any algorithm needs to approximate those many probabilities to accuracy of ϵ, hence the lower bound.

Theorem 11 bounds the required complexity to approximate all pairwise probabilities. Proof is in Appendix F.1

Theorem 11. *For $\epsilon < 1/48$, there exists a model that satisfies both SST and STI for which any algorithm requires $\Omega\Big(\frac{|S|\min(|S|,1/\epsilon)}{\epsilon^2}\log|S|\Big)$ comparisons to approximate all pairwise probabilities to ϵ accuracy w.p. $\geq 3/4$.*

7.2. Upper Bound

Here we propose an algorithm to approximate all pairwise probabilities to an accuracy of ϵ.

The proposed algorithm, first finds an $\epsilon/8$-ranking of the input set S and then approximates pairwise probabilities. By Theorem 9, w.p.$\geq 1-\frac{1}{|S|^2}$ we can find an $\epsilon/8$-ranking of the input set S using $\mathcal{O}\Big(\frac{|S|\log|S|}{\epsilon^2}\Big)$ comparisons. We present APPROX-PROB that given an $\epsilon/8$-ranked set, approximates all pairwise probabilities to an accuracy of ϵ.

APPROX-PROB APPROX-PROB takes an $\epsilon/8$-ranked ordered set S i.e., $\tilde{p}_{S(i),S(j)} \leq \epsilon/8\ \forall i < j$ and bias ϵ and approximates all pairwise probabilities to an accuracy of ϵ.

Note that it is enough to approximate $\tilde{p}_{S(j),S(i)}$ for $j \geq i$ since $\tilde{p}_{S(i),S(j)} = -\tilde{p}_{S(j),S(i)}$. For all $i > 1$, APPROX-PROB compares $S(i)$ and $S(1)$, $\frac{16\log|S|^4}{\epsilon^2}$ times and approximates $\tilde{p}_{S(i),S(1)}$ by $\hat{\tilde{p}}_{S(i),S(1)}$, the fraction of times $S(i)$ won rounded off to the nearest multiple of ϵ. Since for perfectly ranked ordered set $\tilde{p}_{S(i+1),S(1)} \geq \tilde{p}_{S(i),S(1)}$, if $\hat{\tilde{p}}_{S(i+1),S(1)} < \hat{\tilde{p}}_{S(i),S(1)}$, then APPROX-PROB corrects $\hat{\tilde{p}}_{S(i+1),S(1)}$, setting it equal to $\hat{\tilde{p}}_{S(i),S(1)}$. It can be shown that $\tilde{p}_{S(i),S(1)}$ is approximated to an accuracy of $\frac{7\epsilon}{8}$.

APPROX-PROB continues this process by approximating $\tilde{p}_{S(i),S(2)}$ for $i \geq 2$ by increasing i one at a time. For a perfectly ranked set, $\tilde{p}_{S(i-1),S(2)} \leq \tilde{p}_{S(i),S(2)} \leq \tilde{p}_{S(i),S(1)}$ and hence if $\hat{\tilde{p}}_{S(i-1),S(2)} = \tilde{p}_{S(i),S(1)}$, APPROX-PROB does not use comparisons to approximate $\tilde{p}_{S(i),S(2)}$, instead assigns $\hat{\tilde{p}}_{S(i),S(2)} = \hat{\tilde{p}}_{S(i-1),S(2)}$. Whenever $\hat{\tilde{p}}_{S(i-1),S(2)} \neq \tilde{p}_{S(i),S(1)}$, APPROX-PROB approximates $\tilde{p}_{S(i),S(2)}$ by comparing $S(i)$ and $S(2)$. It can be shown that $\tilde{p}_{S(i),S(2)}$ is approximated to accuracy of ϵ.

APPROX-PROB continues this process for $S(3)$, then $S(4)$ and so on until $S(n)$. Notice that whenever $\hat{\tilde{p}}_{S(i-1),S(j)} = \hat{\tilde{p}}_{S(i),S(j-1)}$, APPROX-PROB does not use comparisons to approximate $\tilde{p}_{S(i),S(j)}$ but simply assigns $\hat{\tilde{p}}_{S(i),S(j)} = \hat{\tilde{p}}_{S(i-1),S(j)}$. We show this in fact happens at many places and only $\mathcal{O}(|S|\min(|S|,1/\epsilon))$ pairwise probabilities are approximated using comparisons. This enables obtaining orderwise same upper bound as the lower bound.

Algorithm 5 APPROX-PROB

inputs
  Ordered Set S, bias ϵ
$\hat{\tilde{p}}_{S(1),S(1)} = 0$
for i from 2 to $|S|$ **do**
  Compare $S(1)$ and $S(i)$ for $\frac{16}{\epsilon^2}\log|S|^4$ times
  $\hat{\tilde{p}}_{S(i),S(1)} = \Big[\frac{\text{fraction of times } S(i) \text{ won}}{\epsilon} - \frac{1}{2}\Big]\epsilon$
  if $\hat{\tilde{p}}_{S(i),S(1)} < \hat{\tilde{p}}_{S(i-1),S(1)}$ **then**
    $\hat{\tilde{p}}_{S(i),S(1)} = \hat{\tilde{p}}_{S(i-1),S(1)}$
  end if
end for
for j from 2 to $|S|$ **do**
  $\hat{\tilde{p}}_{S(j),S(j)} = 0$
  for k from $j+1$ to $|S|$ **do**
    if $\hat{\tilde{p}}_{S(k-1),S(j)} = \hat{\tilde{p}}_{S(k),S(j-1)}$ **then**
      $\hat{\tilde{p}}_{S(k),S(j)} = \hat{\tilde{p}}_{S(k-1),S(j)}$
    else
      Compare $S(j)$ and $S(k)$ for $\frac{16}{\epsilon^2}\log|S|^4$ times
      $\hat{\tilde{p}}_{S(k),S(j)} = \Big[\frac{\text{fraction of times } S(k) \text{ won}}{\epsilon} - \frac{1}{2}\Big]\epsilon$
    end if
  end for
end for

Theorem 12 shows the correctness of APPROX-PROB and bounds its comparisons. Proof is in Appendix F.3

Theorem 12. *Given an $\epsilon/8$-ranked ordered set S i.e., $\tilde{p}_{S(i),S(j)} \leq \epsilon/8\ \forall i < j$, APPROX-PROB$(S,\epsilon)$ uses $\mathcal{O}(\frac{|S|\min(|S|,1/\epsilon)}{\epsilon^2}\log|S|)$ comparisons and w.p.$\geq 1-\frac{1}{|S|^2}$ approximates all pairwise probabilities to accuracy of ϵ.*

8. Experiments

In this section, we compare the performance of our maxing algorithms with previous work on synthetic data. All results presented here are averaged over 1000 runs.

We compare our maxing algorithms SOFT-SEQ-ELIM, NEAR-OPT-MAX, and OPT-MAX with **SEQ-ELIMINATE** (Falahatgar et al., 2017a), **KNOCK-OUT** (Falahatgar et al., 2017b), **MallowsMPI** (Busa-Fekete et al., 2014a), **AR** (Heckel et al., 2016) and **BTM-PAC** (Yue & Joachims, 2011). **KNOCKOUT** and **BTM-PAC** are PAC maxing algorithms for models with both SST and STI properties. **SEQ-ELIMINATE** is a PAC maxing algorithm for SST model. **MallowsMPI**, originally designed for Mallows model, finds a condorcet winner which exists under WST. **AR** is a maxing algorithm that finds Borda winner that is same as condorcet winner

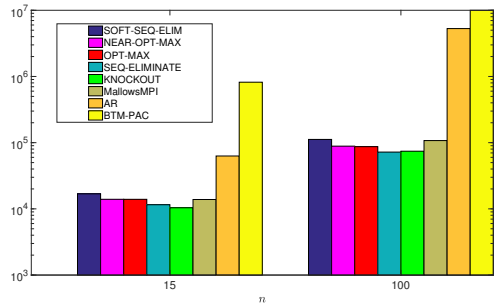

Figure 1. Maxing Algorithms for model with SST and STI

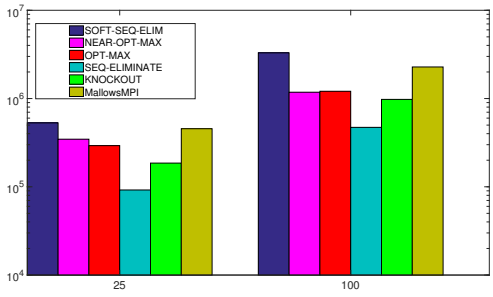

Figure 2. Maxing Algorithms for model with MST but not SST

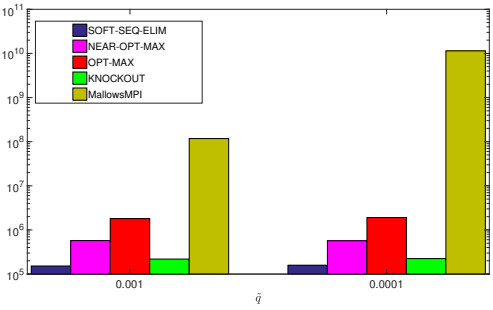

Figure 3. Maxing algorithms for model without STI

under WST. In all experiments, we use maxing algorithms to find a 0.05-maximum with $\delta = 0.1$.

We first consider the model $p_{i,j} = 0.6\ \forall i < j$ same as in (Yue & Joachims, 2011; Falahatgar et al., 2017b;a) that satisfies both SST and STI properties. Note that $i = 1$ is the only 0.05-maximum under this model. Figure 1 presents number of comparisons used by each maxing algorithm. Observe that compared to other algorithms, **BTM-PAC** uses too many comparisons even for $n = 15$. The reason might be **BTM-PAC** is mainly intended for reducing regret in the conventional bandits setting. The bar for **BTM-PAC** complexity for $n = 100$ is not fully shown in the figure to better scale the other complexity bars. Comparison complexity of AR is high for $n = 100$ mainly because **AR** eliminates elements based on Borda scores and Borda scores are very close to each other for large n. We drop **BTM-PAC** and **AR** henceforth.

Now we consider a model that satisfies MST but not SST, i.e., $p_{5i+l,5i+k} = 0.6\ \forall i < n/5 - 1, 1 \le l < k \le 5$ and $p_{5i+l,5j+k} = 0.52\ \forall i < j < n/5 - 1, 0 < l, k \le 5$. Notice that under this model elements are divided into groups of five where within each group $|\tilde{p}_{i,j}| = 0.1$ and for elements in two different groups $|\tilde{p}_{i,j}| = 0.02$, hence there is a 0.05-maximum in each group. Figure 2 demonstrates comparison complexity of algorithms under this model. **SEQ-ELIMINATE** uses fewer comparisons, but it fails to output a 0.05-maximum with probability 0.21 for $n = 25$ and 0.19 for $n = 100$. Hence **SEQ-ELIMINATE** fails once SST is not satisfied. This is because when you compare a 0.05-maximum of a group with an element in other group, 0.05-maximum can get eliminated with probability ≈ 0.5. Hence with lots of groups **SEQ-ELIMINATE** fails. Other algorithms find a 0.05-maximum in all runs. We drop **SEQ-ELIMINATE** henceforth.

Now we consider a model that does not satisfy STI but satisfies MST i.e., $n = 10$ and $p_{1,j} = 1/2 + \tilde{q}\ \forall j \le n/2$, $p_{1,j} = 1\ \forall j > n/2$ and $p_{i,j} = 1/2 + \tilde{q}\ \forall 1 < i < j$, $\tilde{q} < 0.05$. Under this model any $i \le 5$ is a 0.05-maximum. Figure 3 shows the average comparison complexity of algorithms under this model. **KNOCKOUT** uses fewer comparisons, but fails to output a 0.05-maximum with probability 0.12 for $\tilde{q} = 0.001$ and 0.25 for $\tilde{q} = 0.0001$, hence fails to meet the confidence requirement once STI is dropped. Other algorithms find a 0.05-maximum in all runs.

It is interesting to note that **MallowsMPI** uses more comparisons as $\tilde{q}$ decreases, whereas the complexity of other algorithms remains almost same. This is because **MallowsMPI** tries to find absolute maximum which is not always practical. Further note that the performance of SOFT-SEQ-ELIM is better than NEAR-OPT-MAX, and NEAR-OPT-MAX is better than OPT-MAX. This is because the bias gap for SOFT-SEQ-ELIM, NEAR-OPT-MAX and OPT-MAX is ϵ, $\epsilon/2$ and $\epsilon/4$ respectively, resulting in higher constants for NEAR-OPT-MAX and OPT-MAX. While the theoretical order complexity is higher for SOFT-SEQ-ELIM, in practice it can find a good anchor quickly and seems to have near-linear order complexity.

9. Conclusion

We studied the problem of maxing, ranking, and estimating comparison probabilities under different stochastic transitivity constraints. We showed that under WST, maxing needs quadratic comparisons. We also presented a linear-complexity algorithm for maxing under MST. We also proposed an optimal ranking algorithm for SST models with Stochastic Triangle Inequality, closing $(\log \log n)^3$ gap. For the same model, we proposed an optimal algorithm for estimating the comparison probabilities.

ACKNOWLEDGMENTS

We thank NSF for supporting this work through grants CIF-1564355 and CIF-1619448.

References

http://www.gif.gf/.

Acharya, J., Falahatgar, M., Jafarpour, A., Orlitsky, A., and Suresh, A. T. Maximum selection and sorting with adversarial comparators and an application to density estimation. *arXiv preprint arXiv:1606.02786*, 2016.

Ajtai, M., Feldman, V., Hassidim, A., and Nelson, J. Sorting and selection with imprecise comparisons. *ACM Transactions on Algorithms (TALG)*, 12(2):19, 2015.

Baltrunas, L., Makcinskas, T., and Ricci, F. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pp. 119–126. ACM, 2010.

Busa-Fekete, R., Hüllermeier, E., and Szörényi, B. Preference-based rank elicitation using statistical models: The case of mallows. In *Proc. of the ICML*, pp. 1071–1079, 2014a.

Busa-Fekete, R., Szörényi, B., and Hüllermeier, E. Pac rank elicitation through adaptive sampling of stochastic pairwise preferences. In *AAAI*, 2014b.

Caplin, A. and Nalebuff, B. Aggregation and social choice: a mean voter theorem. *Econometrica: Journal of the Econometric Society*, pp. 1–23, 1991.

Chatterjee, S. et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

Chen, X., Bennett, P. N., Collins-Thompson, K., and Horvitz, E. Pairwise ranking aggregation in a crowd-sourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 193–202. ACM, 2013.

Dudík, M., Hofmann, K., Schapire, R. E., Slivkins, A., and Zoghi, M. Contextual dueling bandits. *arXiv preprint arXiv:1502.06362*, 2015.

Falahatgar, M., Hao, Y., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems*, pp. 7063–7073, 2017a.

Falahatgar, M., Orlitsky, A., Pichapati, V., and Suresh, A. T. Maximum selection and ranking under noisy comparisons. In *International Conference on Machine Learning*, pp. 1088–1096, 2017b.

Feige, U., Raghavan, P., Peleg, D., and Upfal, E. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Heckel, R., Shah, N. B., Ramchandran, K., and Wainwright, M. J. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arXiv preprint arXiv:1606.08842*, 2016.

Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

Jang, M., Kim, S., Suh, C., and Oh, S. Top-k ranking from pairwise comparisons: When spectral ranking is optimal. *arXiv preprint arXiv:1603.04153*, 2016.

Lee, D. T., Goel, A., Aitamurto, T., and Landemore, H. Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

Luce, R. D. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

Mohajer, S., Suh, C., and Elmahdy, A. Active learning for top-k rank aggregation from noisy comparisons. In *International Conference on Machine Learning*, pp. 2488–2497, 2017.

Negahban, S., Oh, S., and Shah, D. Iterative ranking from pair-wise comparisons. In *NIPS*, pp. 2474–2482, 2012.

Negahban, S., Oh, S., and Shah, D. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 2016.

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

Radlinski, F. and Joachims, T. Active exploration for learning rankings from clickthrough data. In *Proceedings of the 13th ACM SIGKDD*, pp. 570–579. ACM, 2007.

Radlinski, F., Kurup, M., and Joachims, T. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 43–52. ACM, 2008.

Rajkumar, A. and Agarwal, S. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proc. of the ICML*, pp. 118–126, 2014.

Shah, N., Balakrishnan, S., Guntuboyina, A., and Wainwright, M. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pp. 11–20, 2016a.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. Feeling the bern: Adaptive estimators for bernoulli probabilities of pairwise comparisons. *arXiv preprint arXiv:1603.06881*, 2016b.

Skorepa, M. *Decision making: a behavioral economic approach*. Palgrave Macmillan, 2010.

Soufiani, H. A., Chen, W., Parkes, D. C., and Xia, L. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems*, pp. 2706–2714, 2013.

Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. Online rank elicitation for plackett-luce: A dueling bandits approach. In *NIPS*, pp. 604–612, 2015.

Yue, Y. and Joachims, T. Beat the mean bandit. In *Proc. of the ICML*, pp. 241–248, 2011.