Efficient Rank Aggregation Using Partial Data

Ammar Ammar
Massachusetts Institute of Technology
32 Vassar St.
Cambridge, USA
ammar@mit.edu

Devavrat Shah*
Massachusetts Institute of Technology
32 Vassar St.
Cambridge, USA
devavrat@mit.edu

ABSTRACT

The need to rank items based on user input arises in many practical applications such as elections, group decision making and recommendation systems. The primary challenge in such scenarios is to decide on a global ranking based on partial preferences provided by users. The standard approach to address this challenge is to ask users to provide explicit numerical ratings (cardinal information) of a subset of the items. The main appeal of such an approach is the ease of aggregation. However, the rating scale as well as the individual ratings are often arbitrary and may not be consistent from one user to another. A more natural alternative to numerical ratings requires users to compare pairs of items (ordinal information). On the one hand, such comparisons provide an "absolute" indicator of the user's preference. On the other hand, it is often hard to combine or aggregate these comparisons to obtain a consistent global ranking.

In this work, we provide a tractable framework for utilizing comparison data as well as first-order marginal information (see Section 2) for the purpose of ranking. We treat the available information as partial samples from an unknown distribution over permutations. We then reduce ranking problems of interest to performing inference on this distribution. Specifically, we consider the problems of (a) finding an aggregate ranking of n items, (b) learning the mode of the distribution, and (c) identifying the top k items. For many of these problems, we provide efficient algorithms to infer the ranking directly from the data *without* the need to estimate the underlying distribution. In other cases, we use the Principle of Maximum Entropy to devise a concise parameterization of a distribution consistent with observations using only $O(n^2)$ parameters, where n is the number of items in question. We propose a distributed, iterative algorithm for estimating the parameters of the distribution. We establish the correctness of the algorithm and identify its rate of convergence explicitly.

Categories and Subject Descriptors

G.3 [**Information Systems Applications**]: Statistical Computing

Keywords

Ranking, Aggregation, Maximum Entropy

1. INTRODUCTION

Judging, rating, or ranking objects is omnipresent: whether it be restaurants in a city, movies on Netflix, books on Amazon, candidates interviewed for faculty positions or papers submitted to the ACM Sigmetrics conference. In all such instances, a global ranking of objects is achieved based on the inputs about partial rankings provided by a large number of people.

The current practice is to seek input in terms of scores, e.g. assign between 1 to 5 stars to a restaurant/movie or score between 1 to 5 for a paper. The key advantage of seeking such quantitative input is that it is easy to achieve global aggregation: in Sigmetrics, for example, each paper may receive scores from, say 5 TPC reviewers, between 1 to 5; the average of these scores will lead to the global ranking of all the submitted papers to assist in making the final acceptance/rejection decisions in the TPC meeting.

On the flip side, the key disadvantage stems from the fact that scores are *relative*: the score of 4, for example, may be interpreted differently by different individuals. Furthermore, the same individual may score objects differently depending upon the contextual details, such as the order in which s/he reviewed the assigned papers. While one may argue that it could be possible to correct for such reviewer "biases" using auxiliary information, such an approach is somewhat ad-hoc and even not feasible when the ratings are obtained anonymously.

An alternative approach of seeking input,which we advocate in this paper, is qualitative [1] : for example, ask reviewers explicitly to compare the papers they reviewed (score assignments do not necessarily achieve this as there could

*This work was supported in parts by AFOSR Complex Networks Program, MURI on Tomography of Social Networks and NSF CMMI Program. We would like to acknowledge useful conversations with Vivek Farias, Srikanth Jagabathula, and Andrea Montanari.

SIGMETRICS'12, June 11–15, 2012, London, England, UK.

[1]We believe that in an ideal system, inputs of both forms should be obtained for better decision making. In this paper, we focus on qualitative inputs primarily to understand what sorts of information, on its own, does it contain.

be a tie and in the context of anonymous ratings, this is totally different from quantitative rating). One key advantage of seeking such information is that it is more *absolute*: when two individuals say they like A over B, they do mean the same; or a reviewer is likely to compare two papers in the same way despite the order in which they review them. It is no surprise that some polling sites (e.g. Washington Post [1]) have started using such interfaces to collect information. Further, in many settings data is naturally available in this form; for instance, customers reveal their preferences among items on display at a store by purchasing one of them, cf. [15].

The key challenge with qualitative information arises in the aggregation phase, due to possible contradictions: for three items A, B and C, we could have a scenario where one person prefers A over B, another prefers B over C, and two people prefer C over A as shown in Figure 1. Such apparent conflicts have created challenges for aggregation over the centuries starting with the celebrated work of Condorcet [9]; also see work on impossibility of existence of rankings pioneered by Arrow [5]. Unlike the standard setting of the, so called, ranked elections considered in the literature following the works [9] [5], in our context, we have access to partial ranking information of objects: in general, we have a fraction of population comparing a given pair of objects unlike in the standard ranked election literature where each individual provides complete ranking.

The main contribution of this paper lies in a proposal of a novel method for aggregation of such partial (qualitative) ranking information to come up with a global ranking. The key insight is to view the collected data as the partial information about an underlying distribution over complete orderings of all objects. For example, the votes shown on the left in Figure 1 could have originated from the complete user preferences shown on the right.

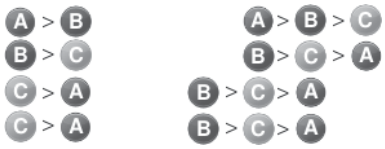

Figure 1: Data and Underlying Distribution

That said, the problem of aggregation reduces to making certain inferences on the underlying distribution. For many aggregation tasks, these inferences can be made directly from the data, with or without making assumptions about the underlying distribution. For other tasks, learning the distribution is necessary. Both cases are presented, and when the latter is the case, our approach involves finding the distribution with maximal entropy (near) consistent with the observed data. The inference problems and our solutions are summarized in the table below.

Problem	First-Order	Comparison
Aggregate Ranking	From Data	From Data
Mode	From Data	Max-Ent
Top-K Ranking	From Data	Max-Ent

Table 1: Summary of results.

In Table 1, we use "From Data" to indicate the combination of data type and problem that can be solved directly from the data, and "Max-Ent" to indicate those where learning an underlying distribution (a maximum entropy one in our case), is necessary. Before we describe these contributions in further detail, we quickly recall related work.

Related Work. The question of learning distribution over permutations from partial or limited information has been well studied in the recent literature. Notably, in the work of Huang, Guestrin and Guibas [18], the task of interest is to infer the most likely permutation of identities of objects that are being tracked through noisy sensing by maintaining distribution over permutations. To deal with the 'factorial blowup', authors propose to maintain only the first-order marginal information of the distribution (essentially corresponding to certain Fourier coefficients), then use the Fourier inversion formula to recover the distribution and subsequently predict its mode as the likely assignment. In the work by Jagabathula and Shah [19], authors took a different approach to the same problem where they proposed to learn the distributed over permutations by finding the sparsest distribution consistent with the observed partial information. Finally, this approach was further extended and integrated with the decision making in the context of revenue management in work by Farias, Jagabathula and Shah [15]. None of these works, however, deals with the question of aggregation or achieving ranking. While the mode of the distribution is a candidate for such a ranking, it might not necessarily be a robust one. It should also be noted that, through maximum entropy distribution learning, we are trying to be maximally unconstrained subject to observed data, unlike the above cited approaches which implicitly or explicitly impose additional constraints (e.g. sparsity).

The task of ranking objects or assigning scores has been of great interest over the past decade or so with similar concerns. There is a long list of works, primarily in the context of bipartite ranking, including the RankBoost by Freund et al. [16], label ranking by Dekel et al. [11], Crammer and Singer [10], Shalev-Shwartz and Singer [25] as well as analytic, learning results on bipartite ranking including those of Agarwal et al. [3], Usunier et al. [27] and Rudin and Schapire [24]. The algorithm that will be closest to our proposal is the p-norm push algorithm by Rudin [23] which uses ℓ_p norm of information to achieve ranking.

The algorithmic view on rank aggregation was revived in work by Dwork et al [13] where they consider design of approximation algorithms to find 'optimal' ranking with respect to a specific metric on permutations. Very recently, a high-dimensional statistical inference view for learning distribution over permutations based on comparison data has been introduced by Mitliagkas et al [22].

The maximum entropy approach for learning distribution is a classical one dating back to the work of Boltzman. The maximum entropy (max-ent) distribution, a member of an appropriate exponential distribution family, is maximum likelihood estimation of the parameters in that family (cf. see [29]). Indeed, the use of exponential family distribution over rankings has been around for more than few decades now (cf. see [12, Chapter 9]). We provide a careful analysis of a stochastic sub-gradient algorithm for learning the parameters of this max-entropy distribution. This algorithm is distributed and iterative. It directly builds upon the algorithm used in [21] for distributed wireless scheduling. It

is worth taking note of use of maximum entropy distribution over permutation based on given marginal information to learn the "missing" marginals by Agrawal et al [4] in the context of parimutuel betting.

Our contributions. The primary contribution of this paper is the use of the framework of distributions over permutations as means to reach rank aggregation from collection of partial preferences.

In our setting, the data available for ranking comes in two different flavors: (a) pair-wise comparison data (e.g. item i is preferred to item j), and (b) first-order marginal data (e.g. item i is ranked in position k). Given data in either form, we focus our attention on three aggregation problems: (1) finding an aggregate ranking over a collection of items (e.g. NetFlix movies), (2) finding the most likely ordering of the items (e.g. object tracking a la [18]), and (3) identifying the top-k items in a collection (e.g. selecting accepted papers for Sigmetrics 2012, such as this one). We solve (1) by introducing a general method which gives each item a score that reflects its importance according to the distribution. We then present a specific instance of this method which allows us to compute the desired scores from the data (comparison or first-order marginal) directly, without the need to learn the distribution. More importantly, we show that the ranking induced by this scoring method is equivalent to the ranking obtained from the family of Thurstone (1927, [26]; also see [12, Ch 9]) models, a popular family of parametric distributions used in a wide range of applications (e.g. online gaming and airline ticket pricing). For (2), we use the principle of maximum entropy to derive a concise parameterization of an underlying distribution (most) consistent with the data. Given the form of the max-ent distribution (an exponential family), computing the mode reduces to solving a maximum-weight matching problem on a bipartite graph with weights induced from the parameters. For the case of first-order marginals, this is an easy instance of the network-flow problem (can be solved, for example, using belief propagation [6]). Furthermore, we propose a heuristic for mode computation as well that bypasses the step of learning the max-ent parameters but uses directly the available partial preference data. Such a heuristic, for example, can speed up computation of [18] drastically. Somewhat curiously, we show that this heuristic is first-order approximation of the mode finding of the max-ent distribution. For pair-wise comparisons representation, the problem is not known to be solvable in polynomial time. We propose a simple randomized scheme that is a 2-approximation of it. We solve problem (3) using another distribution-based scoring scheme where the scores can be computed directly from the data for first order marginals, or by learning a max-ent distribution in the case of comparisons.

We present a stochastic gradient algorithm for learning the max-ent distribution needed for some of the aforementioned problems. This algorithm is derived from [21], however the proof is different (and simpler). It provides explicit rate of convergence for both data types (comparisons and first-order marginals). In both cases, the algorithm uses an oracle to compute intermediate marginal expectations of the max-ent distibution. We prove that the exact computation of such marginal expectations is $\#P$-hard. Using standard MCMC methods and their known mixing time bounds, our analysis suggests that for a collection of n items, the computation time scales exponentially in n and polynomially in n respectively for the pair-wise comparisons and first-order marginals respectively. Two remarks are in order: first, the result for first-order marginals also suggests a distributed scheduling algorithm for input-queued switches with polynomial time learning complexity (unlike exponential for wireless network model). Second, the standard stochastic approximation based approaches cf. [8] do not apply as is (due to compactness of domain related issues).

2. MODEL AND PROBLEM STATEMENT

Model. We consider a universe of n available items, $\mathcal{N} = \{1, 2, ..., n\}$. Each user has preference order, represented as permutation, over these n items. Specifically, if σ is the permutation, the user prefers item i over j if $\sigma(i) < \sigma(j)$. We assume that there is a distribution, say μ, over the space of permutations of n items, S_n, that defines the collective preferences of the entire user population.

Data. We consider scenarios where we have access to partial or limited information about μ. Specifically, we shall restrict our attention to two popular types of data: first-order ranking and comparisons. Each of these two types correspond to some sort of marginal distribution of μ as follows:

First-order marginals: For any $1 \leq i, k \leq n$, the fraction of population that ranks item i as their kth choice is the first-order marginal information for distribution μ. Specifically,

$$m_{ik} \stackrel{\triangle}{=} \mathbb{P}_\mu[\{\sigma(i) = k\}] = \sum_{\sigma \in S_n} \mu(\sigma) \mathbb{I}_{\{\sigma(i)=k\}} \tag{1}$$

where $\mathbb{I}_{\{E\}}$ denotes the indicator variable for event E. Collectively, we have the $n \times n$ matrix $[m_{ij}]$ of the first-order marginals, that we shall denote by M. This is the type of information that was maintained for tracking agents in the framework introduced by Huang, Guestrin and Guibas [18].

Comparison Data: For any $1 \leq i, j \leq n$, the fraction of population that prefers item i over item j is the comparison marginal information. Specifically,

$$c_{ij} \stackrel{\triangle}{=} \mathbb{P}_\mu[\{\sigma(i) < \sigma(j)\}] = \sum_{\sigma \in S_n} \mu(\sigma) \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}. \tag{2}$$

Collectively, we have access to the $n \times n$ matrix $[c_{ij}]$ of comparison marginals, denoted by C. Such data is available through customer transactions in many businesses, cf. [15].

Remarks. First, while we assume m_{ik} (resp. c_{ij}) available for all i, k (resp. i, j), if only a subset of it is available, the algorithm with that information works equally well, with the obvious caveat that the quality of the output is dependant on the richness of our data. Second, we shall assume that $m_{ik} \in (0, 1)$ for all i, k (resp. $c_{ij} \in (0, 1)$ for all i, j). Finally, in practice one may have a noisy version of M or C data. However, the procedures we describe are inherently robust (as they are simple continuous functions of the observed data) with respect to small noise in the data. Therefore, for the purpose of conceptual development, such an idealized assumption is reasonable.

Goal. Roughly speaking, the goal is to utilize data of type M or C to obtain various useful rankings of the objects of interest. Specifically, we are interested in (a) finding an aggregate or representative ranking over the items in question,

(b) finding 'most likely' (or mode) ranking, and (c) finding a ranking that emphasizes the top k objects.

To address these questions, we propose the following approach: (a) assume that the data originates from some underlying distribution over permutations. (b) Use the data to answer the question directly, without learning the distribution, whenever possible. (c) Otherwise, learn a distribution that is consistent with the data (M or C), and use said distribution to answer the question. In principle, there could be multiple, possibly infinitely many, distributions that are consistent with the observed data (M or C, assuming it is generated by a consistent underlying unknown distribution). As mentioned earlier, we shall choose the max-ent distribution that is consistent with the observed data.

Result outline. Here we provide somewhat detailed explanation of the results summarized in the table presented earlier. Specifically, we solve problems (a), (b), and (c) as follows. To find an aggregate ranking (Section 3.1), we assign each item a score derived from the distribution over permutations. We then propose an efficient algorithm to compute said score directly from the data, without learning the distribution . We show that the ranking induced by the computed scores is equivalent to the ranking induced by the parametric family of Thurstone models [26][12, Ch 9] (Section 3.1), a popular family of distributions used in applications ranging from online gaming to airline ticket pricing. Effectively, our result implies that if one learns *any* of the distributions in this family from the data, and uses the learned parameters to obtain a ranking, then this ranking is *identical* to the ranking we obtain directly from the data (i.e. no need to learn the distribution)!

As for the mode of the distribution, we assume a maximum entropy underlying model, and derive a concise parameterization of the model using $O(n^2)$ parameters (Section 3.2). We also show that finding the mode of the distribution is equivalent to solving an optimization problem on these parameters. In the case of First-Order Marginals (see [18]), this problem is easy and can be solved using max-weight matching on a bipartite graph. We also provide an efficient heuristic for finding the mode in the case of first-order marginal data directly from the data, without learning the max-ent distribution. In the case of comparison data, the problem is more challenging. In this case, we provide an 2-approximation algorithm for finding the mode. In Section 3.3, for the top-k ranking problem, we propose a score that emphasizes the top-k items. We show that this score can be computed exactly and directly from First-Order Marginal data, and approximated using the max-ent distribution in the case of comparison data.

3. MAIN RESULTS

Before we get into the details of estimation the distribution, lets consider the problem of ranking given said distribution. More precisely, lets assume that we are given a distribution over permutations μ, and asked to obtain an ordered list of the items of interest that reflects the collective preference implied by the distribution. A classical approach in this setting is the axiomatic one. In this approach one comes up with a set of axioms that the ranked list should satisfy, and then try to come up with a ranking function or algorithm that satisfies these axioms. Unfortunately, seemingly natural axioms cannot be satisfied by any algorithm [5].

In this section, we opt for a non-axiomatic approach to aggregate preferences. We address the problems of finding: (a) an overall aggregate ranking of all items, (b) the mode of the distribution, and a (c) top-k ranking. These problems demonstrate the utility of having or assuming an underlying distribution, and give rise to situations where one can bypass the learning step and use the data directly for ranking. In the latter situation, one can make the conceptual use of assuming a distribution, without performing complicated computations to obtain a ranking.

3.1 Aggregate Ranking

Here we propose a method to obtain an entire ranking of all objects. Building on the intuition followed by popular voting rules, the basic premise is that the objects that are ranked higher more frequently should be getting higher ranking. This can be formalized as follows: for any monotonically strictly increasing non-negative function $f : \mathbb{N} \to [0, \infty]$, define score $S_f(i)$ for object i as

$$S_f(i) = \sum_{k=1}^{n} f(n-k)\mathbb{P}(\sigma(i) = k). \tag{3}$$

The choice of $f(x) = x^p$ assigns the p^{th} norm of the distribution of $\sigma(i)$ as score to object i.

One can take this line of reasoning further by noting that the exponential function for a given $\Theta > 0$, $f_\Theta(x) = \exp(\Theta x)$, effectively captures the combined effect of all p-norms. Therefore, we propose, what we call the Θ-ranking, with scores defined as:

$$S_\Theta(i) = \sum_{k=1}^{n} \exp(-\Theta k)\mathbb{P}(\sigma(i) = k). \tag{4}$$

By selection $\Theta \approx \ln k$, the scores are effectively capturing the occurrence of objects in top k positions only; and for Θ near 0 they are capturing the effect of lower p moments more prominently. Furthermore, intermediate choices of Θ give effective ranking for various scenarios.

We focus our attention on the case where $p = 1$, and present a score that can take us directly from the data to the ranking, without the intermediate step of learning the distribution. We refer to this ranking as the ℓ_1 ranking.

ℓ_1 *Ranking*

The ℓ_1 score is given by:

$$S_1(i) = \sum_{k=1}^{n} (n-k) \cdot \mathbb{P}[\sigma(i) = k]$$

In the case of first-order marginal data, this score can be computed in a straightforward way. For comparison data, however, the marginals $\mathbb{P}[\sigma(i) = k]$ are not available, without having the distribution. Fortunately, the score above can be computed from the data directly in the following form:

$$S(i) = \frac{1}{n-1} \sum_{j \neq i} \mathbb{P}\big[\sigma(i) < \sigma(j)\big] = \frac{1}{n-1} \sum_{j \neq i} c_{ij}$$

using the following lemma:

LEMMA 1. *Given the definition of $S(i)$ and $S_1(i)$ above, we have*

$$S_1(i) = S(i)$$

A proof of this lemma is provided in Section 6. One interesting aspect of this shortcut is that the equivalence between the different scores does not assume any particular distribution. It only assumes that the underlying distribution is consistent with the data. This suggests that the produced ranking should work with different distributions. One family of such distributions is the one based on the celebrated model of Thurstone [26][12, Ch 9], as we shall see in the next section.

Why ℓ_1 Ranking?

Here we demonstrate the utility of our ℓ_1 ranking by showing its equivalence to the ranking obtained using a Thurstone model. In a Thurstone model, preferences over n items come from a "hidden" process as follows: the "favorability" of each item i is a random variable $X_i = u_i + Z_i$, where u_i is an unknown parameter (also known as the skill parameter), and Z_i is a random variable with some distribution. Furthermore, the random variables $Z_1, ..., Z_n$ are identically distributed. If we take the tuple $(x_1, x_2, ..., x_n)$ to be the outcome of some trial, then item j is ranked in position k if x_j is ranked kth among the values $x_1, ..., x_n$. Equivalently, item i is preferred to item j if $x_i > x_j$. In a typical application of such models one observes these comparisons or positional rankings, and uses these observations to infer the values of the unkown parameters $u_1, ..., u_n$. These values are then used to find a ranking over all items. More precisely, items $i > j > ... > k$ if $u_i > u_j > ... > u_k$.

As it turns out, the ranking obtained by following the algorithm based on ℓ_1 scores is equivalent to the ranking one would get by fitting a Thurstone model. The formal statement is as follows:

THEOREM 1. *Let u_i and u_j be the (skill) parameters assigned to item i and j (respectively) in a Thurstone model, and let $S(i)$ and $S(j)$ be the score assigned to the same items using our method (ℓ_1 scores), then:*

$$u_j \leq u_i \qquad \Leftrightarrow \qquad S(j) \leq S(i) \qquad \forall i, j \tag{5}$$

A proof of this theorem is provided in Section 6. Thurstone models have been used in a wide range of applications such as revenue management in airline ticket sales, and player ranking in online gaming platforms (e.g. a variant of this model is used in Microsoft's TrueSkill [17]).

3.2 The Mode

Given a distribution over permutations that is consistent with the data, in the context of object tracking (a la Huang et al. [18]) one would like to find the most likely permutation under said distribution, or the mode. It is easy to see that the mode of a distribution over permutations is hard to compute in general. To address this difficulty, one might want to follow some criteria for selecting a tractable class of distributions to deal with. Ideally, we would like distributions from this class to obey the constraints given by the data, without imposing any additional structure. This intuitive requirement is captured by the Maximum Entropy criterion, whereby we choose a distribution that maximizes the information entropy while satisfying the data constraints. In the following section, we provide a formal derivation of the maximum entropy distribution along those lines.

The Maximum Entropy Model

Formally, the observations M or C impose the constraints that the distribution, μ, should belong to class $\mathcal{M}$:

$$\sum_{\sigma \in S_n} \mu(\sigma) \mathbb{I}_{\{\sigma(i)=k\}} \;=\; m_{ik}, \quad \forall i, k \in \mathcal{N} \tag{6}$$

or class $\mathcal{C}$:

$$\sum_{\sigma \in S_n} \mu(\sigma) \mathbb{I}_{\{\sigma(i)<\sigma(j)\}} \;=\; c_{ij}, \quad \forall i, j \in \mathcal{N} \tag{7}$$

with the the normalization and non-negativity constraints in both cases.

$$\sum_{\sigma \in S_n} \mu(\sigma) \;=\; 1, \quad \mu(\sigma) \;\geq\; 0, \quad \forall \sigma \in S_n. \tag{8}$$

$\mathcal{M}$ (resp. $\mathcal{C}$) is non-empty only if M (resp. C) is generated by a distribution over S_n to begin with. For clarity of exposition, we will assume that this is the case. When this is not the case, The algorithm that we shall present is based on the solving the Lagrangian dual of an appropriate optimization problem in which the constraints imposed by M (resp. C) are "dualized". Therefore, by construction such algorithm is robust.

Now $|S_n| = n!$ and the data of type M (resp. C) imposes $O(n^2)$ constraints. Therefore, there could be multiple solutions. The max-ent principle suggests that we choose the one that has maximal entropy in the class $\mathcal{M}$ (resp. $\mathcal{C}$). Philosophically, we follow this approach since we wish to utilize the information provided by the data and nothing else, i.e. we do not wish to impose any additional structure beyond what data suggests. It is also well known that such a distribution provides maximum likelihood estimation over certain class of exponential family distributions (cf. [29]). In effect, the goal is to find the distribution that solves the following optimization:

$$\begin{aligned} \max_{\nu} \quad & H_{\text{ER}}(\nu) \triangleq -\sum_{\sigma \in S_N} \nu(\sigma) \log \nu(\sigma) \\ & \nu \in \mathcal{M} \quad \text{or} \quad \mathcal{C}. \end{aligned} \tag{9}$$

It can be checked that the Lagrangian dual of this problem is as follows (since all entries of M, C in $(0, 1)$): let λ_{ik} be the dual variables associated with marginal consistency constraint for $\mathcal{M}$ in (6). Then, the dual takes the following form:

$$\max_{\lambda} \sum_{i,k} \lambda_{ik} m_{ik} - \log\Big(\sum_{\sigma} \exp\Big(\sum_{ik} \lambda_{ik} \mathbb{I}_{\{\sigma(i)=k\}}\Big)\Big) \tag{10}$$

It can be shown that this is a strictly concave optimization and has a unique optimal solution. Let it be $\lambda^* = [\lambda^*_{ik}]$. Then the corresponding primal optimal solution of (9) (with $\mathcal{M}$) is given by

$$\mu(\sigma) \propto \exp\Big(\sum_{i,k \in \mathcal{N}} \lambda^*_{ik} \cdot \mathbb{I}_{\{\sigma(i)=k\}}\Big). \tag{11}$$

Similarly, for the comparison data, the dual optimization takes the form

$$\max_{\lambda} \sum_{i,j} \lambda_{i<j} c_{ij} - \log\Big(\sum_{\sigma} \exp\Big(\sum_{ij} \lambda_{i<j} \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}\Big)\Big), \quad (12)$$

and the optimal primal of (9) given optimal dual $\lambda^* = [\lambda^*_{i<j}]$ is

$$\mu(\sigma) \propto \exp\Big(\sum_{i\neq j\in\mathcal{N}} \lambda^*_{i<j} \cdot \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}\Big). \quad (13)$$

As can be seen, in either case the maximum entropy distribution is parameterized by at most n^2 parameters, which is the same as the degrees of freedom of the received data. For future purposes and with a slight abuse of notation, we shall use $F(\lambda)$ to represent the objective of both Lagrangian dual optimization problems (10) and (12).

Computing the Mode

Having restricted our attention to the maximum entropy distribution, we now proceed to compute the mode. We begin by providing an algorithm for computing the mode exactly in the case of First-Order Marginal data. We then present a more efficient algorithm for approximating the same mode directly from the data without the need to learn the max-ent distribution. Finally, we present an algorithm that uses the max-ent distribution to compute a 2-approximation of the mode in the general case.

Recall that under the maximum-entropy distribution, the logarithm of the probability of a permutation σ is proportional to $\sum_{i,k} \lambda_{ik} \mathbb{I}_{\{\sigma(i)=k\}}$ for first-order marginal data, and $\sum_{i,j} \lambda_{i<j} \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}$ for comparison data. Since the log function is monotone, finding the mode, in both cases, boils down to finding:

$$\sigma^* \in \arg\max_{\sigma\in S_n} \Big(\sum_{i,k} \lambda_{ik} \mathbb{I}_{\{\sigma(i)=k\}}\Big) \quad (14)$$

$$\sigma^* \in \arg\max_{\sigma\in S_n} \Big(\sum_{i,j} \lambda_{i<j} \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}\Big) \quad (15)$$

Solving the problem in (14) exactly is equivalent to the following maximum weight matching problem: consider an $n \times n$ complete bipartite graph with edge between node i on left and node k on right having weight λ_{ik}. A matching is a subset (of size n) edges so that no two edges are incident on same vertext. Let the weight of the matching be the summation of the weights of the edge chosen by it. Then the maximum weight matching in this graph is precisely solving (14). This is a well known instance of the classical network flow problem and has strongly polynomial time algorithms [14]. It also allows for distributed iterative algorithm for finding it including the auction algorithm of Bertsekas [7] and the recently popular (max-product) belief propagation [6]. Thus, overall finding the mode of the distribution for the case of first-order marginal is *easy* and admits distributed algorithmic solution.

Next, we describe a (heuristic) method for finding the mode without requiring the intermediate step of finding the max-ent parameters λ in the case of first-order marginal data. Declare the solution of the following optimization as the mode:

$$\max \sum_{i,k} m_{ik} \mathbb{I}_{\{\sigma(i)=k\}}.$$

That is, in place of λ_{ik}, use m_{ik}. The intuition is that λ_{ik} is higher if m_{ik} is and vice versa. While there is no direct relation between this heuristic and mode of the max-ent approximation, we state the following result which establishes the heuristic to be a 'first-order' approximation. A proof is provided in Section 6.

THEOREM 2. *For $\lambda = [\lambda_{ik}]$ in small enough neighborhood of $\mathbf{0} = [0]$,*

$$m_{ik} \approx \frac{1}{n} + \frac{1}{n-1}\lambda_{ik}.$$

For comparison data, the problem in (15) is also equivalent to a combinatorial problem with the space of objects being the matchings. However, it does not admit the nice representation as above. One way to represent the matchings in comparison form is $n \times n$ matrices, say $B = [B_{ij}]$ with (a) each entry B_{ij} being $+1$ or -1 for all $1 \le i, j \le n$, (b) for all $1 \le i, j \le n$, $B_{ij} + B_{ji} = 0$ (anti-symmetric), and (c) if $B_{ij} = B_{jk} = 1$, then $B_{ik} = 1$ for all $1 \le i, j, k \le n$. The goal is to find B so that $\sum_{ij} B_{ij}\lambda_{i<j}$ is maximized. It is not clear if this is an easy problem.

To address this problem, we have the following 2-approx. algorithm to compute the mode using the parameters of the max-ent distribution: choose L permutatations uniformly at random, compute their weights (defined as per (15)) and select the one with maximal weight among these L permutations. For L large enough, this is essentially with $1/2$ weight of the maximum weight. This requires λ to have all non-negative components. This is not an issue since given the structure of the permutations (each having equal number comparisons, $\sigma(i) < \sigma(j)$, correct) and hence an affine transformation of λ by vector with all components being same constant does change the distribution. Therefore, in principle, we could require the subgradient algorithm to be restricted to the non-negative domain (projected verison). The formal statement about this algorithm is stated below.

THEOREM 3. *Let $\lambda = [\lambda_{i<j}]$ be non-negative vector. Let OPT be the maximum of $\sum_{ij} \lambda_{i<j} \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}$ among all permutation $\sigma \in S_n$. Then in the above described randomized algorithm, if we choose $L \ge \frac{1}{2\delta} \ln \frac{1}{\epsilon}$, then*

$$\mathbb{P}\Big[W(\hat{\sigma}) < \frac{1}{2}(1-\delta)OPT\Big] < \epsilon$$

A proof of this theorem is included in Section 6. To complete the solution, we only need to estimate the parameters of the max-ent distribution. An algorithm is provided in Section 4.

3.3 Top-K Ranking

Here the interest is in finding a ranking that emphasizes the top k objects (the favorites). To do this, we can compute the aggregate ranking, or the mode, and then declare the top k ranked objects in resulting list. We propose a natural way to emphasize the favorites. Intuitively, if an object is ranked among top k positions by a large fraction (probability-wise) of the permutations in the distribution, then it ought to be among favorites. This suggests that for a distribution λ, each object i can be given a score $S_k(i)$, defined as

$$S_k(i) = \mathbb{P}_\lambda[\sigma(i) \le k],$$

In the case of first-order marginal data, this score is nothing but $\sum_{\ell \le k} m_{i\ell}$, and can be computed directly from the data. In the case of comparison data, this score can be inferred from the max-ent distribution, which can be learned by the procedure outlined in Section 4. Finally, once the score is computed, we can now declare the top k objects with highest scores as per $S_k(\cdot)$ as the result of top k.

4. LEARNING THE MAX-ENT MODEL

Here we describe an iterative, distributed sub-gradient algorithm that solves the dual optimization problems (10), (12). First, we describe an idealized procedure that calls certain oracle that estimates marginals of distribution from exponential family. We can, in general, only hope to estimate these marginals approximately because the exact estimation, as we show later, is $\#P$-hard. Therefore, the main result that we state is for a sub-gradient algorithm based on such an approximate oracle. In a later section, we shall describe how to design such an approximate oracle in a distributed manner along with its associated computational cost.

Algorithm 1 MaxEnt Estimation: Using Ideal Oracle

Require: Ranking data $m_{ik} \quad \forall i, k$.

1: **Initialize:** $\lambda^0_{ik} = 0 \ \forall i, k$.
2: **for** $t = 1 \to T$ **do**
3: $\lambda^{t+1}_{ik} \leftarrow \lambda^t_{ik} + \frac{1}{\sqrt{t}}\Big(m_{ik} - \mathsf{E}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}]\Big)$ ($\mathsf{E}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}]$ is provided by an oracle)
4: **end for**
5: Choose $\tau \in \{1, \ldots, T\}$ at random so that $\mathbb{P}(\tau = t) \propto 1/\sqrt{t}$
6: **return** λ^τ

Here, $\mathsf{E}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}] = \sum_{\sigma \in S_n} \mathbb{P}_{\lambda^t}(\sigma)\mathbb{I}_{\{\sigma(i)=k\}}$ where

$$\mathbb{P}_{\lambda^t}(\sigma) \ = \ \frac{1}{Z(\lambda^t)} \exp\Big(\sum_{i,k} \lambda^t_{ik}\mathbb{I}_{\{\sigma(i)=k\}}\Big),$$

with normalizing constant (partition function) defined as:

$$Z(\lambda^t) = \sum_{\sigma \in S_n} \exp\Big(\sum_{i,k} \lambda^t_{ik}\mathbb{I}_{\{\sigma(i)=k\}}\Big)$$

Instead of $\mathsf{E}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}]$, we will use a randomized estimation, $\tilde{\mathsf{E}}_{\lambda^t}(i,k) = \tilde{\mathsf{E}}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}]$, such that the error vector $\mathbf{e}(t) = [\mathbf{e}_{ik}(t)]$ where each component

$$\mathbf{e}_{ik}(t) = \mathsf{E}_{\lambda^t}[\mathbb{I}_{\{\sigma(i)=k\}}] - \tilde{\mathsf{E}}_{\lambda^t}(i,k)$$

is sufficiently small. We state the following result about the convergence of this algorithm.

THEOREM 4. *Suppose that each iteration of the sub-gradient algorithm uses an approximate estimate $\tilde{E}.(\cdot,\cdot)$ such that $\|\mathbf{e}(t)\|_1 \le \frac{1}{A(t)+\|\lambda^*\|_\infty+\|\lambda^*\|_2^2}$, where $A(t) = \sum_{s=1}^t 1/\sqrt{s}$ and λ^* is a solution of the optimization problem. Then, for any $\gamma > 0$, for choice of $T = \Theta\Big(\epsilon^{-2-\delta}\Big(n^2+\|\lambda^*\|_\infty+\|\lambda^*\|_2^2\Big)^{2+\delta}\Big)$, we have*

$$\mathsf{E}\Big[F(\lambda^\tau)\Big] \ge F(\lambda^*) - \epsilon,$$

where $F(\cdot)$ is the objective of dual optimization (10)*. The identical result holds for the comparison information* (12).

A proof of this theorem is included in Section 6.

An Approximate Oracle

Theorem 4 relies on existence of an oracle that can produce an estimation of marginals approximately with appropriate accuracy for each time step t. Computing marginals exactly is computationally hard. For first-order marginal data, this follows from [4]. We prove a similar result for comparison data. Both results are summarized by the following theorem:

THEOREM 5. *Given a max-ent distribution λ, computing $\mathsf{E}_\lambda[\mathbb{I}_{\{\sigma(i)=k\}}]$ and $\mathsf{E}_\lambda[\mathbb{I}_{\{\sigma(i)<\sigma(j)\}}]$ is $\#P$-hard.*

We skip the proof due to space constraint. We now describe an approximate oracle. We shall restrict our description to the Markov Chain Monte Carlo (MCMC) based oracle. In principle, one may use heuristics like Belief Propagation to estimate these marginals instead of MCMC (of course, this may lead to the loss of the performance guarantee).

Now the computation of marginals requires computing $\mathbb{P}_{\lambda^t}(\sigma)$ for any $\sigma \in S_n$. From its form, the basic challenge is in computing the partition function $Z(\lambda^t)$. The partition function $Z(\lambda^t)$ is the same as computation of permanent of a non-negative valued matrix $A = [A_{ik}]$ where $A_{ik} = \mathbf{e}^{\lambda_{ik}}$. In an amazing work, Jerrum, Sinclair and Vigoda [20] have designed Fully Polynomial Time Randomized Approximation Scheme (FPRAS) for computing permanent of any non-negative valued matrix. That is, $Z(\lambda^t)$ (hence $\mathbb{P}_{\lambda^t}(\sigma)$) can be computated within multiplicative accuracy $(1\pm\varepsilon)$ in time polynomial in $1/\varepsilon, n, \log(1/\delta)$ with probability at least $1-\delta$. Therefore, it follows that the desired guarantee in Theorem 4 can be provided for all timesteps (using union bound) with probability at least $1 - 1/n$ within polynomial in n building upon the algorithm of [20].

For the case of comparison information, however no such FPRAS algorithm for computing the partition function is known. Therefore, we suggest a simple MCMC based algorithm and provide the obvious (exponential) bound for it. To that end, define $W_\lambda(\sigma) = \sum_{i,k \in \mathcal{N}} \lambda_{ik} \cdot \mathbb{I}_{\{\sigma(i)<\sigma(j)\}}$, and construct a Markov chain, $\mathfrak{M}(\lambda)$, whose state space is the set of all permutations, S_n, and whose transitions from a given state σ to a new state σ' are given as follows:

1: With probability $\frac{1}{2}$ let $\sigma' = \sigma$.
2: Otherwise, construct σ' as follows:

- Choose two elements i and j uniformly at random; set $\tilde{\sigma}(i) = \sigma(j)$, $\tilde{\sigma}(j) = \sigma(i)$ and $\tilde{\sigma}(k) = \sigma(k)$ for all $k \neq i, j$.
- Set $\sigma' = \tilde{\sigma}$ with probability $\min\{1, \exp\big(W_\lambda(\tilde{\sigma}) - W_\lambda(\sigma)\big)\}$; else set $\sigma' = \sigma$.

Using this Markov chain, we estimate $\mathsf{E}_\lambda[\mathbb{I}_{\{\sigma(i)<\sigma(j)\}}]$ as follows: starting from any initial state, run the Markov chain for T_m steps and then record the state of the Markov chain, say σ^{T_m}. If $\sigma^{T_m}(i) < \sigma^{T_m}(j)$, then record 1 else record 0. Repeat this for S times and obtain the empirical average of the recorded 0/1 values. Declare this as the estimate of $\mathsf{E}_\lambda[\mathbb{I}_{\{\sigma(i)<\sigma(j)\}}]$. Indeed, one simultaneously obtains such

estimates for all i, j. We have the following bound on T_c, which we establish in Section 6:

THEOREM 6. *The above stated Markov chain has stationary distribution μ^* so that*

$$\mu^*(\sigma) \propto \exp\Big(W_\lambda\Big).$$

Let $\mu(t)$ be the distribution of the Markov chain after t steps starting from any initial condition. Then for any given $\delta > 0$, there exists

$$T_c \;=\; \Theta\Big(\exp\Big(\Theta\big(n^2\|\lambda\|_\infty + n\log n\big)\Big)\log\frac{1}{\delta}\Big),$$

such that for $t \geq T_c$,

$$\Big\|\frac{\mu(t)}{\mu^*} - 1\Big\|_{2,\mu^*} < \delta,$$

where $\|\cdot\|_{2,\mu}$ is the χ^2 distance.

Now the total variation distance between $\mu(t)$ and μ^* is smaller than the χ^2 distance between them. Therefore, by Theorem 6, it follows that the estimation error of $\mathbb{P}_\lambda(\sigma)$ using $\mu(t)$ will be at most δ. From Chernoff's bound, by selecting S (mentioned above) to be $O(\delta^{-2}\log n)$ (with large enough constant), it will follow that the estimated empirical marginals for all i, j components must be within error $O(\delta)$ with probability $1 - 1/\text{poly}(n)$. Given that increment in each component of λ as part of the sub-gradient algorithm is $O(\sqrt{T})$ by time T, from Theorem 4, it follows that the $\|\lambda\|_\infty = O\big((n + \|\lambda^*\|_\infty + \|\lambda^*\|_2^2)^{1+\gamma}\big)$ (for any choice of $\gamma > 0$ in Theorem 4). Finally, the smallest δ required in Theorem 4 is an inverse polynomial in n, ϵ, from above discussion it follows that the overall cost of the approximate oracle required for the comparisons effectively scales exponentially in $n^{3+\gamma}$ (ignoring other smaller order terms).

5. EXPERIMENTS AND SCALABILITY

Here we provide results from a simple experiment to demonstrate that the ranking produced by our ℓ_1 algorithm converges to the right ranking for the Multinomial Logit Model, an instance of Thurstone model (choose Z_is to be i.i.d. logit distribution). Specifically, we sample distinct items i and j from $1, ..., n$ uniformly at random as per the distribution. We then consult an MNL model, defined using n parameters, for the value of $\mathbb{I}_{\{\sigma(i)<\sigma(j)\}}$. All the samples are then combined into a matrix $[c_i j]$, which is used to find the ℓ_1 ranking. In Figure 2, we show a plot of the error measured using the normalized number of discordant pairs versus the number of samples used for $n = 10$. As we can see, beyond 500 samples, the error induced is extremely small.

To test the scalability of our method, we implemented a voting/survey tool that enables a large number of participants to vote on any number of items in real time. By doing so, we had the following two questions in mind: in addition to being theoritically interesting, can our algorithm be applied in real time? is comparison based voting practical and simple enough for adoption? through this exeperiment, we believe the answer to both questions to be affirmative.

Our tool was installed in voting booths that were made available to the visitors of the MIT150 event [2], a university-wide public open house. Voting categories included movies, actors, musicians, atheletes, among others, and the results at any time were continuously displayed on a large screen. The participation was impressive, and the feedback was mostly positive, which makes us believe that adopting comparison as form of voting is worth a serious consideration.

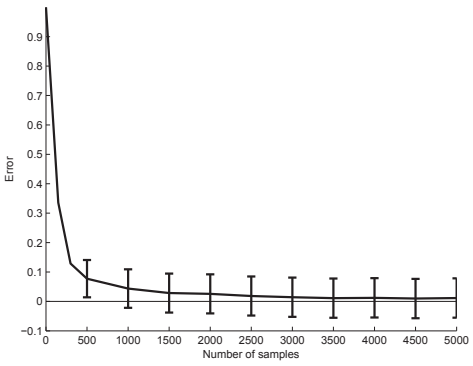

Figure 2: Plot of the error induced by ℓ_1 ranking algorithm for the MNL model, a specific instance of Thurstone's model.

6. PROOFS

This section provides detailed proofs of all the results stated earlier in the paper. Due to space constraints, proof of Theorems 5 and 6 are omitted from this version. [2].

6.1 Proof of Lemma 1

With some arithmetic manipulation, we get

$$\begin{aligned}
S(i) &= \frac{1}{n-1}\sum_{j\neq i}\sum_{\sigma_l\in S_n}\mathbb{P}\big[\sigma(i)<\sigma(j)|\sigma=\sigma_l\big]\mathbb{P}\big[\sigma=\sigma_l\big] \\
&= \frac{1}{n-1}\sum_{\sigma_l\in S_n}\sum_{j\neq i}\mathbb{P}\big[\sigma(i)<\sigma(j)|\sigma=\sigma_l\big]\mathbb{P}\big[\sigma=\sigma_l\big] \\
&= \frac{1}{n-1}\sum_{\sigma_l\in S_n}\big(n-\sigma_l(i)\big)\mathbb{P}\big[\sigma=\sigma_l\big] \\
&= \frac{1}{n-1}\big[n-\mathsf{E}[\sigma(i)]\big] = \sum_{k=1}^{n}(n-k)^1\cdot\mathbb{P}\big[\sigma(i)=k\big] \\
&= S_1(i)
\end{aligned}$$

□

6.2 Proof of Theorem 1

Recall that, under Thurstone's model, each item i has "skill" parameter u_i associated with it. The random "favorability" $X_i = u_i + Z_i$ where Z_i are i.i.d. random variables with some distribution. Our algorithm, with access to exact partial marginal data (first-order or comparison), computes scores for each item i: $S_1(i)$ using first-order data and $S(i)$ using comparison data. As proved in Lemma 1, these two scores are equivalent. Therefore, if we establish that $u_i > u_j$ if and only if $S(i) > S(j)$, it is equivalent to being $S_1(i) > S_2(j)$ as well. We shall establish this statement in two parts: (a) $u_i > u_j$, and (b) $u_i = u_j$.

[2]A full version of this paper is available at `web.mit.edu/ammar/www/rankaggregation2012full.pdf`

Let us start with the first case, $u_i > u_j$. Recall that, score for an item i is

$$S(i) = \frac{1}{n-1}\sum_{k\neq i}\mathbb{P}\big[X_i > X_k\big]$$

Therefore, for $i \neq j$,

$$\begin{aligned} S(i) - S(j) &\propto \Big(\sum_{k\neq i}\mathbb{P}\big[X_i > X_k\big]\Big) - \Big(\sum_{\ell\neq j}\mathbb{P}\big[X_j > X_\ell\big]\Big) \\ &= \Big(\mathbb{P}\big[X_i > X_j\big] - \mathbb{P}\big[X_j > X_i\big]\Big) + \\ &\quad \Big(\sum_{\ell\neq i,j}\Big(\mathbb{P}\big[X_i > X_\ell\big] - \mathbb{P}\big[X_j > X_\ell\big]\Big) \\ &= \Big(\mathbb{P}\big[X_j \leq X_i\big] - \mathbb{P}\big[X_i \leq X_j\big]\Big) + \\ &\quad \Big(\sum_{\ell\neq i,j}\Big(\mathbb{P}\big[X_j \leq X_\ell\big] - \mathbb{P}\big[X_i \leq X_\ell\big]\Big). \quad (16) \end{aligned}$$

Recall that $X_i = u_i + Z_i$ and $X_j = u_j + Z_j$ where u_i, u_j are the "skill" parameters of i and j respectively while Z_i, Z_j are i.i.d. random variables with some distribution. Define, $W_{ij} = Z_i - Z_j$. Then for all i, j, W_{ij} are identically distributed, say with distribution similar to a random variable W which has CDF given by F_W, i.e. $F_W(x) = \mathbb{P}[W \leq x]$. Since W is difference of independent and identically distributed random variables, by definition it is 'symmetric' around 0. That is, for any $x \geq 0$,

$$\mathbb{P}\big[W < -x\big] = \mathbb{P}\big[W > x\big]. \quad (17)$$

Given these notations, it follows that

$$\begin{aligned} \mathbb{P}\big[X_i \leq X_j\big] &= \mathbb{P}\big[W_{ij} \leq u_j - u_i\big] \\ &= F_W(u_j - u_i). \quad (18) \end{aligned}$$

Similarly,

$$\begin{aligned} \mathbb{P}\big[X_j \leq X_i\big] &= F_W(u_i - u_j) \\ \mathbb{P}\big[X_i \leq X_\ell\big] &= F_W(u_\ell - u_i) \\ \mathbb{P}\big[X_j \leq X_\ell\big] &= F_W(u_\ell - u_j). \quad (19) \end{aligned}$$

Since $u_i > u_j$, we have $u_\ell - u_j > u_\ell - u_i$ for any $\ell \neq i, j$. Since F_W is a CDF and hence monotonically non-decreasing, i.e. $F_W(x) \leq F_W(y)$ for all $x \leq y$,

$$F_W(u_\ell - u_j) - F_W(u_\ell - u_i) \geq 0, \quad (20)$$

for all ℓ. Also, let $\delta = u_i - u_j > 0$. Then, from above discussion, (16) becomes

$$\begin{aligned} S(i) - S(j) \propto \Big(F_W(\delta) - F_W(-\delta)\Big) + \\ \Big(\sum_{\ell\neq i,j}\Big(F_W(u_\ell - u_j) - F_W(u_\ell - u_i)\Big)\Big) \end{aligned} \quad (21)$$

Now,

$$\begin{aligned} F_W(\delta) - F_W(-\delta) &= \mathbb{P}[W \in (-\delta, \delta]] \\ &\geq \mathbb{P}[|W| \leq \delta/2]. \quad (22) \end{aligned}$$

As we shall show next, for any distribution of Zs, W is such that for any $\gamma > 0$,

$$\mathbb{P}[|W| \leq \gamma] > 0. \quad (23)$$

From (20)-(23) (and $\gamma = \delta/2$ in last equation), it follows that if $u_i > u_j$, then

$$S(i) - S(j) > 0. \quad (24)$$

Now we establish (23). For this note that due to Z (distributed as Z_i, Z_j) being a distribution, there exists (tightness) $[-a, a] \subset \mathbb{R}$ for some $a > 0$ so that $\mathbb{P}\Big[Z \in [-a, a]\Big] > \frac{1}{2}$. Given any $\gamma > 0$, partition this interval into at most $N = \lceil \frac{4a}{\gamma} \rceil$ disjoint contiguous intervals, each of length $\gamma/2$. One of these intervals must have probability at least $1/2N$. Call this interval I. That is, $\mathbb{P}[Z \in I] \geq 1/2N$. Since Z_i, Z_j are distributed independently and identically distributed manner with distribution same as that of Z, we have that

$$\mathbb{P}[Z_i \in I, Z_j \in I] \geq \frac{1}{4N^2} \;>\; 0. \quad (25)$$

But when both Z_i and Z_j are in I, their difference $W = Z_i - Z_j$ must be withint $[-\gamma/2, \gamma/2]$. This completes the justification of (23).

For the case (b), $u_i = u_j$, using identical arguments as above, one can argue that $S(i) = S(j)$. This complete the proof of Theorem 1.

□

6.3 Proof of Theorem 2

Let λ be in neighborhood of $\mathbf{0} = [0]$. We shall establish claim by means of Taylor's expansion of m as function of λ around $\mathbf{0}$. For simplicity, let us denote $\sigma_{ij} = \mathbb{I}_{\{\sigma(i)=j\}}$. Then

$$m_i j(\lambda) = \sum_{\sigma\in S_n}\sigma_{ij}\frac{1}{Z(\lambda)}\exp\big(\sum_{kl}\lambda_{kl}\sigma_{kl}\big),$$

where partition function $Z(\lambda) = \sum_{\sigma\in S_n}\frac{1}{Z(\lambda)}\exp\big(\sum_{k,l}\lambda_{kl}\sigma_{kl}\big)$. For $\lambda = \mathbf{0}$, we have $m_{ij}(\mathbf{0}) = \frac{1}{n}$ for all i, j. By the first-order Taylor expansion, for λ near 0.

$$m_{ij}(\lambda) \approx m_{ij}(\mathbf{0}) + \sum_{kl}\lambda_{kl}\frac{\partial m_{ij}(\star)}{\partial\lambda_{kl}}\Big|_{\star=\mathbf{0}}. \quad (26)$$

By the property of exponential family (see [29] for example), it follows that

$$\begin{aligned} \frac{\partial m_{ij}(\lambda)}{\partial\lambda_{kl}} &= \mathsf{E}_\lambda\big[\sigma_{ij}\sigma_{kl}\big] - E_\lambda\big[\sigma_{ij}\big]E_\lambda\big[\sigma_{kl}\big] \\ &= \mathsf{E}_\lambda\big[\sigma_{ij}\sigma_{kl}\big] - m_{ij}(\lambda)m_{kl}(\lambda). \quad (27) \end{aligned}$$

From (26) and (27), it follows that for λ near $\mathbf{0}$,

$$m_{ij}(\lambda) \approx \frac{1}{n} + \Big(\sum_{k,l}\lambda_{kl}\mathsf{E}_\lambda\big[\sigma_{ij}\sigma_{kl}\big]\Big) - \frac{1}{n^2}\Big(\sum_{k,l}\lambda_{kl}\Big) \quad (28)$$

We state the following proposition.

PROPOSITION 1. *All distributions can be represented by λ s.t.*

$$\sum_k\lambda_{ik} = 0, \qquad \sum_k\lambda_{kj} = 0, \quad \textit{for } \; 1 \leq i, j \leq n. \quad (29)$$

PROOF. Consider a λ such that (29) is not satisfied. We will transform λ to ν which satisfies (29) but induces exactly the same distribution. Specifically, we shall prove that for each $\sigma, \tilde{\sigma} \in S_n$

$$\sum_{k,l}\nu_{kl}(\sigma_{kl} - \tilde{\sigma}_{kl}) = \sum_{k,l}\lambda_{kl}(\sigma_{kl} - \tilde{\sigma}_{kl}).$$

To that end, define

$$\nu_{ij} = \lambda_{ij} - \frac{1}{n}\lambda_{i\cdot} - \frac{1}{n}\lambda_{\cdot j} + \frac{1}{n^2}\lambda_{\cdot\cdot},$$

where

$$\lambda_{i\cdot} = \sum_{k=1}^{n}\lambda_{ik} \quad , \quad \lambda_{\cdot j} = \sum_{k=1}^{n}\lambda_{kj} \quad , \quad \lambda_{\cdot\cdot} = \sum_{k,l=1}^{n}\lambda_{kl}. \qquad (30)$$

Then,

$$\begin{aligned}\sum_{k=1}^{n}\nu_{ik} &= \sum_{k=1}^{n}\lambda_{ik} - \frac{1}{n}\lambda_{i\cdot}\sum_{k=1}^{n}1 - \frac{1}{n}\sum_{k=1}^{n}\lambda_{\cdot k} + \frac{1}{n^2}\lambda_{\cdot\cdot}\sum_{k=1}^{n}1 \\ &= \lambda_{i\cdot} - \lambda_{i\cdot} - \frac{1}{n}\lambda_{\cdot\cdot} + \frac{1}{n}\lambda_{\cdot\cdot} = 0, \quad \forall i.\end{aligned}$$

Similarily, we can check $\sum_{k=1}^{n}\nu_{kj}$ for all j. Now we have:

$$\begin{aligned}\sum_{k,l}\nu_{kl}\sigma_{kl} &= \sum_{k,l}\lambda_{kl}\sigma_{kl} - \frac{1}{n}\sum_{k}\lambda_{k\cdot}\big(\sum_{l=1}^{n}\sigma_{kl}\big) \\ &\quad - \frac{1}{n}\sum_{l}\lambda_{\cdot l}\big(\sum_{k=1}^{n}\sigma_{kl}\big) + \frac{1}{n^2}\lambda_{\cdot\cdot}\big(\sum_{k,l}\sigma_{kl}\big) \\ &= \sigma_{kl}\lambda_{kl}\sigma_{kl} - \frac{1}{n}\sum_{k=1}^{n}\lambda_{k\cdot} - \frac{1}{n}\sum_{l=1}^{n}\lambda_{\cdot l} + \frac{n}{n^2}\lambda_{\cdot\cdot}\end{aligned}$$

$$= \sum_{kl}\lambda_{kl}\sigma_{kl} - \frac{2}{n}\lambda_{\cdot\cdot} + \frac{1}{n}\lambda_{\cdot\cdot} = \sum_{kl}\lambda_{kl}\sigma_{kl} - \frac{1}{n}\lambda_{\cdot\cdot}$$

Therefore:

$$\begin{aligned}\sum_{kl}\nu_{kl}(\sigma_{kl} - \tilde{\sigma}_{kl}) &= \sum_{kl}\lambda_{kl}(\sigma_{kl} - \tilde{\sigma}_{kl}) - \frac{1}{n}\lambda_{\cdot\cdot} + \frac{1}{n}\lambda_{\cdot\cdot} \\ &= \sum_{kl}\lambda_{kl}(\sigma_{kl} - \tilde{\sigma}_{kl}).\end{aligned}$$

This implies that the distributions induced by λ and ν are identical. □

Given Proposition 1, we shall assume λ satisfying (29) without loss of generality. Then, from (28)

$$\begin{aligned}m_{ij}(\lambda) &= \frac{1}{n} + \sum_{k,l}\lambda_{kl}\mathsf{E}\big[\sigma_{ij}\sigma_{kl}\big] - \frac{1}{n}\lambda_{\cdot\cdot} \\ &= \frac{1}{n} + \sum_{k,l}\lambda_{kl}\mathsf{E}\big[\sigma_{ij}\sigma_{kl}\big] \qquad (31)\end{aligned}$$

Now,

$$\mathsf{E}\big[\sigma_{ij}\sigma_{kl}\big] = \begin{cases}\frac{1}{n} & : k = i, j = l \\ \frac{1}{n(n-1)} & : k \neq i, j \neq l \\ 0 & : k = i, j \neq l\end{cases}$$

Then from (31)

$$m_{ij} = \frac{1}{n} + \frac{1}{n}\lambda_{ij} + \sum_{k\neq i, l\neq j}\frac{1}{n(n-1)}\lambda_{kl}. \qquad (32)$$

Focusing on the last term, we have

$$\begin{aligned}\sum_{k\neq i,l\neq j}\frac{1}{n(n-1)}\lambda_{kl} &= \frac{1}{2n(n-1)}\Big[\sum_{k\neq i}\big(\sum_{q=1,q\neq j}^{n}\lambda_{kq}\big) \\ &\quad + \sum_{l\neq j}\big(\sum_{q=1,q\neq i}^{n}\lambda_{ql}\big)\Big] \\ &= \frac{1}{2n(n-1)}\Big[\sum_{k\neq i}(\lambda_{k\cdot} - \lambda_{kj}) + \\ &\quad \sum_{l\neq j}(\lambda_{\cdot l} - \lambda_{il})\Big] \\ &= -\frac{1}{2n(n-1)}\Big[\sum_{k\neq i}\lambda_{kj} + \sum_{l\neq j}\lambda_{il}\Big] \\ &= -\frac{1}{2n(n-1)}\Big[\lambda_{\cdot j} = \lambda_{ij} + \lambda_{i\cdot} - \lambda_{ij}\Big] \\ &= -\frac{1}{2n(n-1)}(-2\lambda_{ij}) = \frac{\lambda_{ij}}{n(n-1)}\end{aligned}$$

Combining this with (32), we get

$$m_{ij}(\lambda) = \frac{1}{n} + \lambda_{ij}\big(\frac{1}{n} + \frac{1}{n(n-1)}\big) = \frac{1}{n} + \frac{1}{n-1}\lambda_{ij},$$

as desired. □

6.4 Proof of Theorem 3

Denote the difference between the weight of a permutation, σ^i, and the mode by Δ_i, defined as:

$$\Delta_i \stackrel{\triangle}{=} W(\sigma^*) - W(\sigma^i)$$

Since the permutations $\sigma^1, \ldots, \sigma^k$ are drawn uniformly at random, for each permutation σ^i we have:

$$\mathsf{E}\big[W(\sigma^i)\big] = \sum_{i,j}\lambda_{i<j}\mathsf{E}\big[\sigma_{i<j}\big] = \frac{1}{2}\sum_{i,j}\lambda_{i<j} \geq \frac{1}{2}W(\sigma^*)$$

And therefore,

$$\mathsf{E}\big[\Delta_i\big] = W(\sigma^*) - \mathsf{E}\big[W(\sigma^i)\big] \leq \frac{1}{2}W(\sigma^*)$$

Since $\hat{\sigma}$ is chosen to have the maximum weight $W(\cdot)$ of all permutations, and since these permutations are drawn independently, we have:

$$\begin{aligned}\mathbb{P}\Big[W(\hat{\sigma}) < (\frac{1}{2} - \delta)W(\sigma^*)\Big] &= \prod_{i=1}^{k}\mathbb{P}\Big[W(\sigma^i) < (\frac{1}{2} - \delta)W(\sigma^*)\Big] \\ &= \prod_{i=1}^{k}\mathbb{P}\Big[\Delta_i > (\frac{1}{2} + \delta)W(\sigma^*)\Big]\end{aligned}$$

Using the Markov inequality we get:

$$\begin{aligned}\mathbb{P}\Big[W(\hat{\sigma}) < (\frac{1}{2} - \delta)W(\sigma^*)\Big] &\leq \prod_{i=1}^{k}\frac{1}{1+2\delta} \approx \prod_{i=1}^{k}(1-2\delta) \\ &\leq \prod_{i=1}^{k}\mathbf{e}^{-2\delta} \leq \mathbf{e}^{-2\delta k}\end{aligned}$$

Where the approximation is valid for sufficiently small δ. Setting $k > \frac{1}{2\delta}\log\frac{1}{\epsilon}$, we have:

$$\mathbb{P}\Big[W(\hat{\sigma}) < (\frac{1}{2} - \delta)W(\sigma^*)\Big] < \epsilon$$

6.5 Proof of Theorem 4: subgradient algorithm

We shall establish result for the first-order marginal. The proof for comparison is identical. To that end, recall that the optimization problem of interest is

$$\max_{\lambda} \quad F(\lambda) \triangleq \sum_{i,k} \lambda_{ik} m_{ik} - \log\Big(\sum_{\sigma} \exp\Big(\sum_{ik} \lambda_{ik} \mathbb{I}_{\{\sigma(i)=k\}}\Big)\Big). \tag{33}$$

Let λ^* be an optimizer of the objective function with optimal value $F(\lambda^*)$. Now $F(\cdot)$ is a concave function. As before, we shall use t as the index of algorithm's iteration, λ^t be parameter value in iteration t, g^t be the subgradient of $F(\lambda^t) = m - \mathsf{E}_{\lambda^t}[[\mathbb{I}_{\{\sigma(i)=k\}}]]$ and $e(t)$ be the error in this subgradient. Then

$$\begin{aligned}
\|\lambda^{t+1} - \lambda^*\|^2 &= \|\lambda^t + \alpha_t(g^t + e(t)) - \lambda^*\|^2 \\
&= \|\lambda^t - \lambda^*\|^2 + \alpha_t^2\|g^t + e(t)\|^2 \\
&\quad + 2\alpha_t\langle g^t, \lambda^t - \lambda^*\rangle + 2\alpha_t\langle e(t), \lambda^t - \lambda^*\rangle \\
&\le \|\lambda^t - \lambda^*\|^2 + \alpha_t^2\|g^t + e(t)\|^2 \\
&\quad + 2\alpha_t(F(\lambda^t) - F(\lambda^*)) + 2\alpha_t\langle e(t), \lambda^t - \lambda^*\rangle
\end{aligned}$$

where the last inequality follows from the fact that g^t is a subgradient of F at λ^t. Applying this inequality recursively, and keeping in mind that $\|\cdot\| \ge 0$, we get:

$$\begin{aligned}
0 \le \|\lambda^0 - \lambda^*\|^2 &+ 2\sum_{s=0}^{t} \alpha_s(F(\lambda^s) - F(\lambda^*)) \\
&+ \sum_{s=0}^{t} \alpha_s^2\|g^s + e(s)\|^2 + 2\sum_{s=0}^{t} \alpha_s\langle e(s), \lambda^s - \lambda^*\rangle
\end{aligned}$$

Therefore

$$\begin{aligned}
2\sum_{s=0}^{t} \alpha_s(F(\lambda^*) - F(\lambda^s)) &\le \|\lambda^0 - \lambda^*\|^2 \\
&\quad + \sum_{s=0}^{t} \alpha_s^2\|g^s + e(s)\|^2 \\
&\quad + 2\sum_{s=0}^{t} \alpha_s\langle e(s), \lambda^s - \lambda^*\rangle
\end{aligned}$$

Let λ be chosen to be λ^s with probability $p_s = \frac{\alpha_s}{\sum_{q=1}^{t}\alpha_q}$. Then on average, we have

$$\begin{aligned}
\mathsf{E}\big[F(\lambda^*) - F(\lambda)\big] &\le \frac{\|\lambda^0 - \lambda^*\|^2 + \sum_{s=0}^{t} \alpha_s^2\|g^s + e(s)\|^2}{2\sum_{s=0}^{t} \alpha_s} \\
&\quad + \frac{2\sum_{s=0}^{t} \alpha_s\langle e(s), \lambda^s - \lambda^*\rangle}{2\sum_{s=0}^{t} \alpha_s}. \qquad (*)
\end{aligned}$$

To simplify the term in $(*)$, note that $g^s + e(s)$ is a vector whose elements are in $[-1, 1]$. Therefore $\|g^s + e(s)\| \le n^2$, where n is the dimension of the vector. Furthermore, the term $\langle e(s), \lambda^s - \lambda^*\rangle$ can be bounded as follows:

$$\begin{aligned}
\langle e(s), \lambda^s - \lambda^*\rangle \le |\langle e(s), \lambda^s - \lambda^*\rangle| &\le \|e(s)\|_1\|\lambda^s - \lambda^*\|_\infty \\
&\le \|e(s)\|_1(\|\lambda^s - \lambda^0\|_\infty + \|\lambda^0 - \lambda^*\|_\infty)
\end{aligned}$$

And,

$$\|\lambda^s - \lambda^0\|_\infty \le \sum_{q=0}^{s} \alpha_q\|\Delta_q\|_\infty \le \sum_{q=0}^{s} \alpha_q$$

where Δ_q is the change in the value of λ at step q. Combining this with the previous inequality, we get:

$$\langle e(s), \lambda^s - \lambda^*\rangle \le \|e(s)\|_1(\sum_{q=0}^{s} \alpha_s + \|\lambda^0 - \lambda^*\|_\infty)$$

Combining this with $(*)$, and letting $B = \max\{\|\lambda^0 - \lambda^*\|_\infty, \|\lambda^0 - \lambda^*\|^2\}$, we get:

$$\begin{aligned}
\mathsf{E}\big[F(\lambda^*) - F(\lambda)\big] &\le \frac{B + \sum_{s=0}^{t} \alpha_s^2 n^2}{2\sum_{s=0}^{t} \alpha_s} \\
&\quad + \frac{2\sum_{s=0}^{t} \alpha_s\|e(s)\|_1(\sum_{q=0}^{s} \alpha_s + B)}{2\sum_{s=0}^{t} \alpha_s}
\end{aligned}$$

Using our approximation oracle, we can choose the value of $\|e(s)\|$ to be $\frac{\alpha_s}{\sum_{q=0}^{s}\alpha_q + B}$, which yields:

$$\begin{aligned}
\mathsf{E}\big[F(\lambda^*) - F(\lambda)\big] &\le \frac{B + \sum_{s=0}^{t} \alpha_s^2 n^2 + 2\sum_{s=0}^{t} \alpha_s^2}{2\sum_{s=0}^{t} \alpha_s} \\
&= \frac{B + (n^2 + 2)\sum_{s=0}^{t} \alpha_s^2}{2\sum_{s=0}^{t} \alpha_s} \qquad (34)
\end{aligned}$$

Recall that $\alpha_s = \frac{1}{\sqrt{s}}$. Therefore, $\sum_{s=0}^{t} \alpha_s = \Theta(\sqrt{t})$ and numerator scales as $\log t$. Therefore, the quantity above converges to zero, and $F(\lambda^t)$ converges to $F(\lambda^*)$. Now (ignoring constants), the bound in (34) scales like $(B + n^2 \log t)/\sqrt{t}$. Therefore, for $t \ge T$,

$$\mathsf{E}\big[F(\lambda^*) - F(\lambda)\big] \le \epsilon,$$

for any $\gamma > 0$,

$$T = \Theta\Big(\epsilon^{-2-\gamma}\big(\|\lambda^*\|_\infty + \|\lambda^*\|_2^2 + n^2\big)^{2+\gamma}\Big).$$

7. CONCLUSION

In this paper, we have introduced a novel approach for rank aggregation from observed partial data. The key conceptual contribution is viewing the partial data as coming from an underlying 'ground truth' that is distribution over permutations and thus providing a consistent resolution of paradoxes like that of Condorcet. We make this approach feasible by providing efficient algorithms for solving three important classes of rank aggregation problems: (a) selecting an entire ranking, (b) selection of most likely ranking as per the underlying distribution, and (c) choosing top k items. Interestingly, in many of these problems, we can devise algorithms that reach decision directly from data (without learning the underlying distribution) that is consistent with the approach in which one first learns the distribution and then processes the distribution to obtain the desired answer. This makes algorithmic solutions of this paper very attractive for designing large scale ranking systems such as recommendation systems. We strongly believe that algorithmic result of this paper will be of great practical utility across variety of applications and developing such system design could be an interesting direction going forward.

8. REFERENCES

[1] Who had the "worst year in washington"? `http://voices.washingtonpost.com/thefix/worst-week-in-washington/worst-year-in-washington.html`.

[2] MIT open house "Under the dome", 150 years of celebration, Massachusetts Institute of Technology, http://mit150.mit.edu/open-house.

[3] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research*, 6(1):393, 2006.

[4] S. Agrawal, Z. Wang, and Y. Ye. Parimutuel betting on permutations. *Internet and Network Economics*, pages 126–137, 2008.

[5] K.J. Arrow. *Social choice and individual values*. Number 12. Yale Univ Pr, 1963.

[6] M. Bayati, D. Shah, and M. Sharma. Max-product for maximum weight matching: Convergence, correctness, and lp duality. *Information Theory, IEEE Transactions on*, 54(3):1241–1251, 2008.

[7] D.P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, 1988.

[8] V.S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge Univ Pr, 2008.

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

[10] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[11] O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. *Advances in neural information processing systems*, 16, 2003.

[12] P. Diaconis. *Group representations in probability and statistics*, volume 11. Inst of Mathematical Statistic, 1988.

[13] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.

[14] J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

[15] V. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. *Advances in Neural Information Processing Systems*, 22:504–512, 2009.

[16] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.

[17] R. Herbrich, T. Minka, and T. Graepel. Trueskilltm: A bayesian skill rating system. *Advances in Neural Information Processing Systems*, 20:569–576, 2007.

[18] J. Huang, C. Guestrin, and L. Guibas. Efficient inference for distributions on permutations. *Advances in neural information processing systems*, 20:697–704, 2008.

[19] S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[20] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

[21] L. Jiang, D. Shah, J. Shin, and J. Walrand. Distributed random access algorithm: scheduling and congestion control. *Information Theory, IEEE Transactions on*, 56(12):6182–6207, 2010.

[22] I. Mitliagkas, A. Gopalan, C. Caramanis, and S. Vishwanath. User rankings from comparisons: Learning permutations in high dimensions. In *Proceedings of Allerton Conference*, 2011.

[23] C. Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *The Journal of Machine Learning Research*, 10:2233–2271, 2009.

[24] C. Rudin and R.E. Schapire. Margin-based ranking and an equivalence between adaboost and rankboost. *The Journal of Machine Learning Research*, 10:2193–2232, 2009.

[25] S. Shalev-Shwartz and Y. Singer. Efficient learning of label ranking by soft projections onto polyhedra. *The Journal of Machine Learning Research*, 7:1567–1599, 2006.

[26] L.L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

[27] N. Usunier, M.R. Amini, and P. Gallinari. A data-dependent generalisation error bound for the auc. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*. Citeseer, 2005.

[28] L.G. Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[29] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[30] D.J.A. Welsh. *Complexity: knots, colourings and counting*. Number 186. Cambridge Univ Pr, 1993.