Spectral Methods for Ranking with Scarce Data

Umang Varma
Google*

Lalit Jain
University of Washington

Anna C. Gilbert
Yale University*

Abstract

Given a number of pairwise preferences of items, a common task is to rank all the items. Examples include pairwise movie ratings, New Yorker cartoon caption contests, and many other consumer preferences tasks. What these settings have in common is two-fold: a scarcity of data (it may be costly to get comparisons for all the pairs of items) and additional feature information about the items (e.g., movie genre, director, and cast). In this paper we modify a popular and well studied method, RankCentrality for rank aggregation to account for few comparisons and that incorporates additional feature information. This method returns meaningful rankings even under scarce comparisons. Using diffusion based methods, we incorporate feature information that outperforms state-of-the-art methods in practice. We also provide improved sample complexity for RankCentrality in a variety of sampling schemes.

1 INTRODUCTION

In this paper we are interested in the problem of rank aggregation from pairwise preferences under settings where the amount of data is *scarce* but we may have additional *structural* information. For example, consider a setting where a set of pairwise comparisons on a set of n movies have been collected from a set of critics and the goal is to give an overall ranking. If n is large, for example, all movies released in the last two decades, it may be extremely costly to get a comparison for each of the $\binom{n}{2}$ pairs. A more realistic regime is to hope that each movie has been viewed at least once. Standard methods of ranking suggest that the number of comparisons needed is roughly $O(n\log(n))$—when n is large, even hoping for $\log(n)$ comparisons may be hopeless! However, each movie has additional feature information $x_i \in \mathbb{R}^d$. For example, the dimensions could encapsulate the production budget, the number of A-list actors, the writer, studio, animated or live action, etc. In general, we may suspect that these features inform the comparisons: if movies A and B have the same Oscar-winning director, and movie A beats movie C in a comparison, we may expect movie B to also perform well against movie C. In an extreme setting, even if we don't have any comparisons involving movie B, we may still hope to infer a meaningful ranking. In this paper we focus on modifying a popular and well studied method arising in the ranking literature for this setting and demonstrate gains in the *scarce* setting when the number of comparisons is very small.

A common model in the literature of particular interest to us is the Bradley-Terry-Luce (BTL) model. We assume that we have n items and associated to each item i is a positive score w_i so that the probability that j is preferred to i ("j beats i") in a comparison is

$$P_{ij} := P(i \prec j) = \frac{w_j}{w_i + w_j}, \tag{1}$$

and that we see m comparisons. The underlying ranking on the items is then given by the scores w, with an item with a larger score being ranked higher than an item with a smaller score. In the structured setting above, we may expect movies with similar features to have similar scores. Traditional methods of learning w using the BTL model, e.g., maximum likelihood estimation (MLE) or spectral methods such as Rank Centrality (both discussed below), do not naturally incorporate this kind of side information.

We have two main contributions.

1. Our main contribution is Algorithm 1, *Regularized RankCentrality*, in Section 4. We propose a novel method for regularizing the RankCentrality algorithm that returns meaningful rankings even under scarcity. Using diffusion

*This work was done while at University of Michigan.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

based methods, we propose a way of incorporating feature information that is empirically competitive with other feature based methods such as RankSVM or Siamese Networks on both synthetic and real-world datasets in scarce settings. In a specific context, we provide a sample complexity result for this regularized method.
2. Along the way, we discuss traditional RankCentrality and, under a natural sampling scheme extending that in (Rajkumar and S. Agarwal 2014), we show an improved sample complexity bound for the RankCentrality algorithm. For example, when pairs are sampled uniformly, we improve the bound from $O(n^5 \log n)$ to $O(n \log n)$.

2 RELATED WORKS

There is an extensive amount of literature on ranking from pairwise comparisons under various models, and we refer the interested reader to the survey in (Rajkumar and S. Agarwal 2014). Roughly speaking, most frameworks either fall into the parametric setting, i.e., a model such as BTL is assumed, or non-parametric where general assumptions on the pairwise comparison matrix P, where P_{ij} is the probability that i beats j in a comparison, are made.

In the latter setting, several different conditions on P, such as stochastic transitivity and low noise described in (Rajkumar and S. Agarwal 2014), or low rank as in (Koren, Bell, and Volinsky 2009), and generalized low permutation rank models have been proposed (see (N. B. Shah, Balakrishnan, and Wainwright 2018)). All of these models include the BTL model as a specific case. Other estimators such as the Borda count and Condorcet winner (for finding the best item rather than a ranking) have been analyzed in (N. B. Shah and Wainwright 2017). A variant of the ranking problem also falls under the category of active ranking where the comparisons that are queried are chosen by an active ranker rather than passively considered offline, see (Katariya et al. 2018; Heckel et al. 2019; Jamieson and Nowak 2011).

A great deal of attention has been paid to the BTL model. A natural approach to this setting is to compute an estimate for w using the MLE. More precisely given a set of comparisons $S = \{(i_k, j_k, y_k)\}_{k=1}^m$ where the k-th comparison is between items i_k and j_k, and $y_k = 0$ denotes that i_k was preferred in this observation, whereas $y_k = 1$ denotes that j_k was preferred. Then the MLE is given by

$$\operatorname*{argmax}_{v \in \mathbb{R}^n} \sum_{i=1}^{m} -\log\left(1 + e^{(2y_k - 1)(v_{j_k} - v_{i_k})}\right) \qquad (2)$$

and our estimate is $\hat{w}_i = \exp(v_i)$.

We can also consider a constrained MLE where we add an additional constraint[1], e.g., on the maximum entry of w, $\|w\|_\infty < B$, or, alternatively, we can add add an ℓ_2 regularizer $\lambda\|v\|_2$ to the objective. The BTL-MLE in any of these formulations is a popular objective since it is convex. We briefly review the known results on the BTL-MLE. (N. B. Shah, Balakrishnan, Bradley, et al. 2016) have shown the constrained BTL-MLE is minimax optimal for the ℓ_2 error. Note that low ℓ_2 loss does not necessarily guarantee a correct recovery of a ranking. (Chen et al. 2019) shows that the (regularized) MLE and spectral ranking methods (discussed below) are minimax optimal for recovery of a ranking. The critical parameter for recovery is the minimum gap between any two different BTL scores—which does not show up when one is interested in the ℓ_2 norm only.

In the next section we discuss the class of algorithms that are the main study of this work: spectral methods and the RankCentrality algorithm.

3 SPECTRAL METHODS

We assume that we have access to a collection of m independent and identically distributed pairwise comparisons $S = \{(i_k, j_k, y_k)\}_{k=1}^m$ where each $i_k < j_k \in [n]$. Furthermore we assume that each pair is i.i.d drawn: $(i, j) \sim_\mu \{(i, j), 1 \le i < j \le n\}$, where μ is an *unknown* sampling distribution on the set of ordered pairs. Although μ_{ij} is defined for $i < j$, we assume it is understood that $\mu_{ij} = \mu_{ji}$ when $i > j$. Denote $\mu_{\min} := \min_{i<j} \mu_{ij}$ and $\mu_{\max} := \max_{i<j} \mu_{ij}$. In addition, we assume that the label is an independent Bernoulli draw, i.e.

$$y_k = \begin{cases} 1 & \text{with probability } P_{i_k j_k} = \frac{w_{j_k}}{w_{i_k} + w_{j_k}} \\ 0 & \text{otherwise} \end{cases}$$

according to the BTL model where $(w_1, \cdots, w_n) \in \mathbb{R}^n_{>0}$ is an unknown vector of BTL-scores, i.e., $i_k \prec j_k$ with probability $P_{i_k j_k}$. Note $P_{ij} = 1 - P_{ji}$. Additionally define $b := \max_{i,j} w_i / w_j$. Without loss of generality we assume that $w^T \mathbf{1} = 1$, indeed scaling the weights has no effect on the comparison probabilities.

Problem. Given S, return $\hat{w}$, an estimator for w.

Consider the following matrix $Q \in \mathbb{R}^{n \times n}$, defined as

$$Q_{ij} := \begin{cases} \mu_{ij} P_{ij} & \text{if } i \neq j \\ 1 - \sum_{\ell \neq i} \mu_{i\ell} P_{i\ell} & \text{if } i = j \end{cases}. \qquad (3)$$

Observe Q_{ij} is the transition matrix of a time-reversible Markov chain, where the we transition from i to j with

[1] Without loss of generality, assume $\sum_i w_i = 1$ because P_{ij} is invariant to scaling w.

probability proportional to that of i beating j in a comparison (we refer the reader to Chapter 1 of (Norris 1998) for background on Markov Chains), i.e., it satisfies the detailed balance equations: for all $i \neq j$, we have

$$w_i Q_{ij} = \frac{\mu_{ij} w_i w_j}{w_i + w_j} = w_j Q_{ji}.$$

This implies the vector w is the stationary distribution of Q, satisfying $w^T Q = w$, i.e., w_i is the equilibrium probability of being in state i. This motivates using the stationary distribution of an empirical estimator $\hat{Q}$, with $\mathbb{E}[\hat{Q}] = Q$ as an estimator $\hat{w}$ for w. The impatient reader can skip ahead to the next section for our choice of $\hat{Q}$.

The connection between the BTL model and time-reversible Markov chains was noticed by (Negahban, Oh, and D. Shah 2016) where they proposed the RankCentrality algorithm for estimating w under a slightly different model. In their setting, they assume they have access to a (connected) graph on n vertices G, and for each edge in the graph they repeatedly query the associated pairwise comparison k times. In the specific setting of an Erdős–Rényi graph $\mathcal{G}_{n,p}$ on n vertices, they construct an estimator $\hat{w}$ and show for $d \geq 10C^2 \log n$ and $kd \geq 128C^2 b^5 \log n$, setting $p = \frac{d}{n}$ the following bound on the error rate holds with high probability:

$$\frac{\left\|\hat{w} - w\right\|_2}{\|w\|_2} \leq 8Cb^{5/2}\sqrt{\frac{\log n}{k\,d}}.$$

(where we recall $b := \max_{i,j} w_i/w_j$). Noting that the expected number of comparisons is $O(n^2 pk) = O(nkd) = O(b^5 n \log(n))$ this yields a sample complexity of $O(b^5 n \log n/\epsilon^2)$ for recovering a weight vector with relative error ϵ. Note that in this setting, for $\mathcal{G}_{n,p}$ to even be connected, it is important that p be at least on order $\log(n)/n$, and we must at least observe $O(n \log(n))$ comparisons. In the more general setting, the sample complexity depends on the spectral gap of the graph Laplacian of G ; precise dependencies have been given in (A. Agarwal, Patil, and S. Agarwal 2018; N. B. Shah, Balakrishnan, Bradley, et al. 2016)

Returning to our setting, our sampling scheme, which we refer to as *independent sampling* was proposed by (Rajkumar and S. Agarwal 2014). Observe that the independent sampling scheme is more natural in many applications, and in particular each observation is made independent of the other observations, which is not true of those in (Negahban, Oh, and D. Shah 2016). Rajkumar and Agarwal show that if $O(\frac{Cn}{\varepsilon^2 P_{\min}^2 \mu_{\min}^2} b^3 \ln\left(\frac{n^2}{\delta}\right))$ comparisons are made then with probability at least $1-\delta$ (over the random draw of m samples from which $\hat{P}$ is constructed), the score vector $\hat{w}$ produced by their version of the RankCentrality algorithm satisfies $\|\hat{w} - w\|_2 \leq \varepsilon$. The sample complexity here scales as $O(n^5 \log n)$ since $\mu_{\min}^{-1} \geq \binom{n}{2}$, with equality achieved only when μ is uniform. In the next section we propose a different estimator from the one given in (Rajkumar and S. Agarwal 2014) and we are able to give a $O(n \log n)$ sample complexity bound in the case of uniform sampling.

A crucial point to note is that both (Negahban, Oh, and D. Shah 2016) and (Rajkumar and S. Agarwal 2014) assume that the directed graph of comparisons, where an edge (i, j) represents that j beat i in at least one comparison, is strongly connected. This is because the empirical estimate $\hat{Q}$ of the Markov transition matrix needs to be ergodic, i.e., irreducible and aperiodic, which ensures that $\hat{Q}$ has a unique stationary distribution. When the number of comparisons m is small (i.e., $m < n \log(n)$ in the case of (Negahban, Oh, and D. Shah 2016)), this is usually not the case and these algorithms return a default output. In particular, in the setting mentioned in the introduction where the number of comparisons are scarce, these methods will not return a useful ranking. This is a primary motivation for the work in this paper.

3.1 WARM-UP: IMPROVED RESULTS FOR INDEPENDENT SAMPLING

In this section we improve the results given in (Rajkumar and S. Agarwal 2014) by using a different estimator of Q than the one presented there. Recall the notation of Section 3. Given a dataset of comparisons S, define

$$C_{ij} = \sum_{k=1}^{m} \Big(\mathbf{1}\{i_k = i, j_k = j, y_k = 1\} + \mathbf{1}\{i_k = j, j_k = i, y_k = 0\} \Big),$$

i.e., C_{ij} is the *number* of comparisons between i and j that j won. Additionally define the *empirical Markov transition matrix*

$$\hat{Q}_{ij} := \begin{cases} \frac{C_{ij}}{m} & \text{if } i \neq j \\ 1 - \sum_{\ell \neq i} \frac{C_{i\ell}}{m} & \text{if } i = j \end{cases}. \tag{4}$$

By construction, $Q = \mathbb{E}(\hat{Q})$ so $\hat{Q}$ is an unbiased estimator of Q. Let $\hat{w}$ be the leading left eigenvector of $\hat{Q}$. When $\hat{Q}$ is ergodic, $\hat{w}$ is the unique stationary distribution of $\hat{Q}$.

Theorem 1. *Fix $\delta \in (0,1)$ and $\varepsilon \in (0,1)$. If*

$$m \geq 64 b^3 n^{-1} \mu_{\min}^{-2} \varepsilon^{-2} (\mu_{\max} + n\mu_{\max}^2) \log \frac{2n}{\delta}$$

and the empirical Markov chain $\hat{Q}$ constructed as in (4) *is ergodic, then with probability at least $1-\delta$, we have*

$$\frac{\|\hat{w} - w\|}{\|w\|} \leq \varepsilon.$$

Proof. A complete proof can be found in the supplementary materials. We sketch an outline of the proof here.

We first prove a result on the deviation of left eigenvectors for perturbations of ergodic row stochastic matrices, Proposition 5 based on ideas from (Negahban, Oh, and D. Shah 2016). For each observation $k \in [m]$, we define a random i.i.d. matrix Q_k (in terms of i_k, j_k, and y_k) such that $\hat{Q} = I + \frac{1}{m}\sum_{k=1}^{m} Q_k$. We can therefore write $\hat{Q} - Q = \sum_k Z_k$ where each Z_k is an independent random matrix with $\mathbb{E}(Z_k) = 0$ and we can explicitly compute the matrix variance of Z_k (Lemma 8). By using matrix Bernstein inequalities given in (Tropp 2012) we can derive a central-limit type upper bound on $P(\|\hat{w} - w\| > \varepsilon)$ (Theorem 10). Solving the resulting inequality for m, we get the desired result. □

Because $\mu_{\min} = \mu_{\max} = \binom{n}{2}^{-1}$ when μ is uniform, we have given an $O\left(b^3\varepsilon^{-2}n\log(\frac{n}{\delta})\right)$ sample complexity when μ is uniform. Our argument improves upon that in (Rajkumar and S. Agarwal 2014) through improved matrix concentration results and a different (unbiased) estimator for Q.

4 REGULARIZING RANKCENTRALITY

When the number of pairwise comparison observations we have available is small, the $\hat{Q}_{ij}$ entries are poor estimators for Q_{ij}: there are $n^2 - n$ off-diagonal entries in $\hat{Q}$ and each observation only affects one off-diagonal entry leaving most entries zero. Furthermore, as described in the previous section, if the graph of pairwise comparisons (given by connecting any two points with an edge) is not strongly connected, may not guarantee that $\hat{Q}$ has a unique stationary distribution. **Motivated by this, we ask a natural question—when the number of pairwise comparisons is small; i.e., data is scarce (for example we have just observed one comparison per item) how can we still obtain a reasonable ranking?**

Intuitively, if the items $[n]$ have some inherent structure, we can hope to exploit that structure to infer pairwise comparisons. Since $Q_{ij} = \mu_{ij}P_{ij}$; i.e., a scaled probability of i beating j, even if we have never seen a comparison between i and j, it is reasonable to estimate this value by taking a weighted combination of the empirical $\hat{Q}_{ik}, 1 \le k \le n$, where the choice of weights perhaps reflect some prior knowledge on the similarity between j and k. In an extreme case—if we suspect item j and k would perform the same against item i, we may choose the weight on $\hat{Q}_{ik}$ to be large, and set the weights on all other $\hat{Q}_{ik'}, k \neq k'$ to zero.

Said more precisely, we choose a row-stochastic matrix D and use the estimator $\hat{Q}D$ whose ij-th entry is

$$[\hat{Q}D]_{ij} = \sum_{k=1}^{n} D_{kj}\hat{Q}_{ik} \tag{5}$$

How should we choose D? We want $\hat{Q}D$ to be ergodic, but it should also reflect some similarity structure between the items. This prior information could take form in many ways—for example we can imagine that associated to item i is a feature vector $x_i \in \mathbb{R}^d$ and intuitively items that are close together perform similarly on a comparison with some other element j (see Section 4.1). An extreme case of this is assuming that the items are in clusters, and items within a cluster rank similarly (or the same). Finally, we can consider forms of D that do not reflect any prior structure but do at least guarantee that $\hat{Q}D$ is ergodic—as we will show these estimators can still perform competitively with other methods (Section 4.2). To recap, our resulting regularized RankCentrality algorithm that we will discuss in the rest of this section is given below in Algorithm 1.

Algorithm 1 Regularized RankCentrality algorithm

1: **procedure** RANKCENTRALITY(n, S, D)
2: **compute** $\hat{Q}$ as in (4)
3: **return** leading left eigenvector of $\hat{Q}D$
4: **end procedure**

4.1 DIFFUSION BASED REGULARIZATION

Diffusion RankCentrality leverages additional features $x_i \in \mathbb{R}^d$ for each of the items $i \in [n]$ being ranked. We use this to compute pairwise similarities in a manner consistent with the literature (e.g., in t-SNE (Maaten and Hinton 2008) and diffusion maps formulated by (Coifman et al. 2005)) so that for a fixed i, the similarities D_{ik} are proportional to the probability density of a Gaussian centered at x_i. Let $D^{(\sigma)}_{ik}$, the similarity between item i and j, be defined as

$$D^{(\sigma)}_{ik} := \frac{\exp\left(\frac{-\|x_i - x_k\|^2}{\sigma^2}\right)}{\sum_{l=1}^{n}\exp\left(\frac{-\|x_i - x_l\|^2}{\sigma^2}\right)}, \tag{6}$$

where σ, the kernel width, is an appropriately chosen hyperparameter. The Diffusion RankCentrality algorithm, obtained by using $D^{(\sigma)}$ in Algorithm 1, returns the stationary distribution of the Markov chain $\hat{Q}D^{(\sigma)}$.

As described in equation (5), $[\hat{Q}D^{(\sigma)}]_{ij} = \sum_{k=1}^{n} D^{(\sigma)}_{kj}\hat{Q}_{ik}$, i.e., the ij entry is a weighted average of $\hat{Q}_{ik}$'s. $D^{(\sigma)}_{ij}$ is large when x_i is close to x_j and

close to 0 when they are far apart. In particular the $\hat{Q}_{jk}$ contribute more when j is close to i and less otherwise.

An alternative interpretation of this procedure is given by considering the Markov chain induced by $\hat{Q}$ and contrasting it with that of $\hat{Q}D^{(\sigma)}$. Consider starting at any item i, and repeatedly transitioning according to $\hat{Q}$. If the number of comparisons is small, there may not even be a path from i to any other item j. In addition, any additional comparison greatly affects the stationary distribution (i.e. the limiting distribution as we transition according to $\hat{Q}$) of $\hat{Q}$. Contrast this with the stationary distribution of $\hat{Q}D^{(\sigma)}$. By construction, $\hat{Q}D^{(\sigma)}$ will be dense (assuming each element has some neighbor that has a comparison). We can interpret the elements of $\hat{Q}D^{(\sigma)}$ as a Markov chain themselves: first, we make a sub-step (say from i to k) according to $\hat{Q}$, which is based only the pairwise comparison observations, and then we make a sub-step (say from k to j) with probability that inversely depends the distance of points to k. In, particular, we have imputed a series of transitions from i to other elements j, using the underlying geometry of the points along with the pairwise comparisons. This technique is similar to that found in (Dijk et al. 2018), the MAGIC algorithm used in the field of single-cell RNA sequencing, where each entry in Q is an extremely undersampled low integer count.

Example. Consider the following extreme case example. Suppose the 100 points $\{x_i\}_{i=0}^{99}$ lie in 10 tight clusters with cluster k being $\{x_{10k+1}, \cdots, x_{10k+9}\}$ and the clusters are spaced very far apart. Assume the BTL scores of items are constant within clusters; if items i and j are in the same cluster then $x_i = x_j$ and $w_i = w_j$. Set $\|x_i - x_j\| = \infty$ when i and j are in different clusters. In this case, the matrix $D^{(\sigma)}$ is block diagonal: $D^{(\sigma)}_{ij} = \frac{1}{10}$ when i and j are in the same cluster and $D^{(\sigma)}_{ij} = 0$ otherwise.

Figure 1 demonstrates the benefit of multiplying $\hat{Q}$ by $D^{(\sigma)}$. We see that a comparison between i and j does not just affect the ij entry, but those corresponding to neighbors of i and j. To visualize the effect of $D^{(\sigma)}$, we also show heatmaps of the 50-th powers of the transition matrices, $\hat{Q}$ and $\hat{Q}D^{(\sigma)}$. The checkered patterns in Q and $QD^{(\sigma)}$ are clearly visible in $(\hat{Q}D^{(\sigma)})^{50}$ while $\hat{Q}^{50}$ is still very sparse. After 50 iterations of $\hat{Q}$ vs. $\hat{Q}D^{(\sigma)}$, we see the impact of regularization, $(\hat{Q}D^{(\sigma)})^{50}$ is far less sparse than $\hat{Q}^{50}$ and reflects a block structure that is imputing comparisons for items that have been compared less often.

There are a number of different ways we could have diffused the information across the samples. We could have used $\hat{Q}D^{(\sigma)}$, $D^{(\sigma)}\hat{Q}$, or even $D^{(\sigma)}\hat{Q}D^{(\sigma)}$. In our empirical analysis, however, we found no significant difference

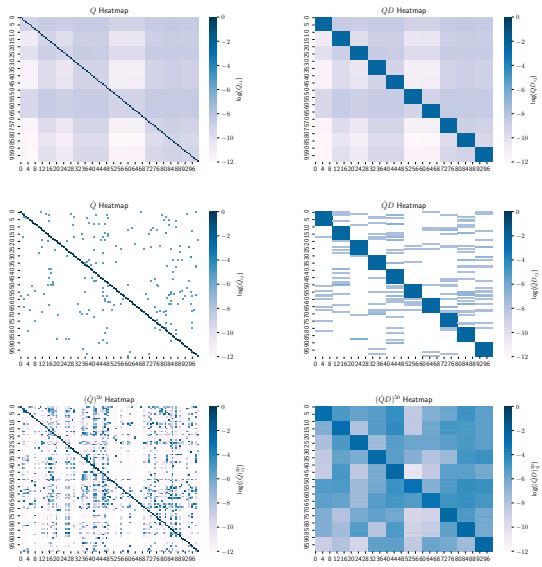

Figure 1: Demonstrating the impact of $D^{(\sigma)}$. The 100 items in this experiment lie in 10 equally sized tight clusters, where BTL scores are constant within clusters and the corresponding $D^{(\sigma)}$ matrix is block diagonal. The $\hat{Q}$ matrix was computed using 200 pairwise comparisons simulated according to the BTL model.

in the performance of the algorithm run with these possibilities.

Finally, we note that the running time of the regularized RankCentrality algorithm is dominated by the computation of the leading eigenvector. The matrices Q and D are of size $n \times n$ and we can form the matrix $M = \hat{Q}D$ in time $O(n^3)$. We then iterate in the power method with M, each iteration, requiring a matrix-vector multiply takes time $O(n^2)$. Our empirical analysis suggests that a few steps of the power method are sufficient. Furthermore, this iterative eigenvector computation on sparse matrices can be faster, than optimization procedures inherent in the MLE.

4.2 λ-REGULARIZED RANKCENTRALITY

Implicitly, D is chosen so that two properties are satisfied. Firstly, $\hat{Q}D$ will be an ergodic markov chain, and secondly, as in most regularization situations, we choose D to capture some inherent prior structural information we may have about w apriori. In this section we ignore the second motivation and instead focus on a D which just guarantees that former constraint.

In particular, given $\lambda > 0$ we consider $D_\lambda := (1-\lambda)I + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$ as a choice of regularizer in Algorithm 1. Note that $\hat{Q}D_\lambda = (1-\lambda)\hat{Q} + \frac{\lambda}{n}\mathbf{1}\mathbf{1}^T$, which ensures that $\hat{Q}D_\lambda$ is a

positive row-stochastic matrix, *which must be ergodic*. In particular, we can run Algorithm 1, regardless of the number of samples and we are guaranteed that $\hat{Q}D_\lambda$ necessarily has a unique stationary distribution. The simple nature of D_λ allows us to give a precise theoretical characterization of it's performance. In general, $\mathbb{E}\left[\hat{Q}D_\lambda\right] = QD_\lambda$, but QD_λ may not have the same left eigenvector as Q. This introduces a bias in our estimator. How can we overcome this bias? Inspecting the form of D_λ, note that if $\lambda \to 0$ as $m \to \infty$ then $D_\lambda \to I$. The following theorem characterizes the error of this procedure of any λ and shows that it is reasonable to take $\lambda = O(1/\sqrt{m})$. For notational convenience, we let $\gamma := \frac{n\mu_{\min}}{2(1+\sqrt{2})b^{3/2}}$. Note that γ is not constant—in fact it is $O(\frac{1}{n})$.

Theorem 2. *Let $\lambda \in (0, \frac{\gamma}{2})$. Choose $\delta \in (0,1)$ and $\varepsilon \in \left(2\lambda\gamma^{-1}, 1\right)$. Let $\hat{w}_\lambda$ be the output of Regularized RankCentrality run with $D = D_\lambda$. Then, with probability at least $1-\delta$,*

$$\frac{\|\hat{w}_\lambda - w\|}{\|w\|} < 2\lambda\gamma^{-1} + \sqrt{\frac{68(1-\lambda)b^3(\mu_{\max}+n\mu_{\max}^2)}{n\mu_{\min}^2 m}\log\frac{2n}{\delta}},$$

In particular, choosing $\lambda = c/\sqrt{m}$, then with probability at least $1-\delta$, we have

$$\frac{\|\hat{w} - w\|}{\|w\|} = O\left(\frac{b^3\log(2n/\delta)}{n\mu_{\min}m}\right).$$

We give a proof in the supplementary material under Corollary 14.

Our empirical experiments run with $\lambda = \eta m^{-1/2}$ for various values of η support decaying λ in this way. Figure 2 demonstrates a run of λ-Regularized RankCentrality on a setting where $w = [i]_{i=1}^{200}$ and the underlying distribution on pairwise comparisons is assumed to be uniform. We compare several choices of λ (with $\lambda = 0$ corresponding to normal RankCentrality) and the BTL MLE with an ℓ_2 regularizer[2] on the weights (implemented using logistic regression). Note that $\eta = 1/6$ seems to perform the best and even outperforms regularizing the BTL-MLE for small sample sizes where RankCentrality may still be returning a uniform distribution. For more details and experiments with different choices of w in this setting, see Appendix C in the supplementary materials.

Remark: To connect the diffusion based regularization with λ-regularization, observe that if we take $\sigma \to 0$ in the definition of D in Equation 6, then $D \to D_0 = I_n$ (when the x_i's are all distinct). The kernel width σ, therefore, determines the bias of Diffusion RankCentrality—small values of σ only introduce a small bias in the algorithm

[2]Without such a regularizer, the BTL-MLE is underdetermined when the number of comparisons is small and cannot be solved.

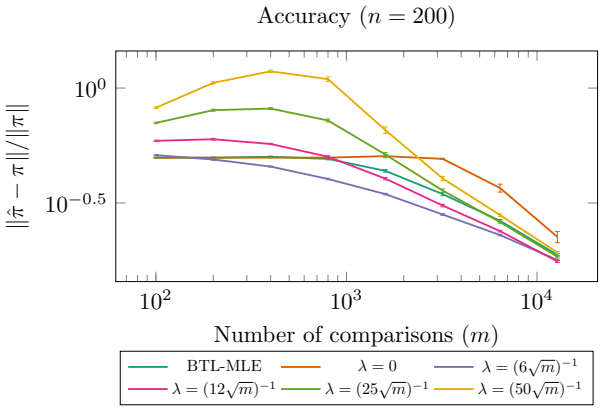

Figure 2: Comparing λ-Regularized RankCentrality with BTL-MLE and RankCentrality. Here $w = [i]_{i=1}^{200}$.

while large values of σ introduce considerable bias. Motivated by Theorem 2, to diminish this bias as m increases, we can use $(1 - \frac{1}{\sqrt{m}})I + \frac{1}{\sqrt{m}}D^{(\sigma)}$ in Diffusion RankCentrality instead of $D^{(\sigma)}$ directly. We call this *Decayed Diffusion RankCentrality*. In general, cross-validation could be used to choose the kernel width.

5 EMPIRICAL RESULTS FOR REGULARIZED RANKCENTRALITY

In this section we do a comparison of the regularized RankCentrality methods in the structured setting to standard methods for ranking on synthetic and real world datasets. The code we used along with additional plots are part of the supplementary material. Although our theoretical analyses do not make assumptions about μ, our experiments focus on the case where μ is uniform.

5.1 COMPARISON TO SCORING FUNCTIONS

As discussed in Section 2, there is a rich literature of ranking methods, though less so for ranking data that come with features. Recall, we assume for each item $i \in [n]$ there is a vector $x_i \in \mathbb{R}^d$. In past work, the goal is to learn a function $f : \mathbb{R}^d \to \mathbb{R}$, presumed to be in a specified function class $\mathcal{F}$, such that $\text{sign}(f(x_i) - f(x_j))$ predicts a comparison between item i and item j. To learn f given the dataset $S = \{(i_k, j_k, y_k)\}_{k=1}^m$, and a loss function $\ell : \mathbb{R} \times \mathbb{R} \times \{0,1\} \to \mathbb{R}$, we can learn the empirical risk minimizer $\text{argmin}_{f\in\mathcal{F}} \sum_{k=1}^n \ell(f(x_i), f(x_j), y_k)$. Two notable examples that focus on learning a scoring function that we compare to are RankSVM by (Joachims 2002) and Siamese network based approaches due to (Bromley et al. 1994).

RankSVM assumes that $\mathcal{F} = \{f : x \mapsto w^T x\}$,

i.e. linear separators through the origin and choose $\ell(f(x_i), f(x_j), y) = \min(0, 1-(f(x_i)-f(x_j))(2y-1)$. When testing RankSVM, we used it naively on the original features but also considered a kernelized version using random features, as described in (Rahimi and Recht 2008) and implemented in SkLearn, (Pedregosa et al. 2011).

Note that when the loss function is the logistic loss, $\ell(f(x_i), f(x_j), y) = \log\left(\frac{\exp(f(x_j))}{\exp(f(x_i))+\exp(f(x_j))}\right)$, we recover the MLE under the assumption that the BTL scores are given by a transformation of the features. Such an objective has been proposed several times in the literature, e.g. (Burges et al. 2005). In the extreme case $f(x_i) = \theta_i$ is the BTL-MLE.

An example of such an approach are Siamese Nets, introduced by in (Bromley et al. 1994). We implemented a Siamese network using Keras ((Chollet et al. 2015)) with two hidden dense layers, each with 20 nodes and a dropout factor of 0.1, and an output dimension of 1. Each layer in the base network used a ReLU activation. The outputs of the right network is subtracted from that of the left and a cross-entropy loss is then used.

We point out that in general both methods described above have a very different goal from what our paper proposes. Our goal is **not** to learn a scoring function, but instead to use the similarity information to inform the ranking process. In general, learning a scoring function can be expensive in terms of both computation, and samples. In addition, if the features do not actually inform the ranking very well, we want methods that will still learn a reasonable ranking—guaranteed by regularized RankCentrality as $m \to \infty$. We now demonstrate competitive performance of regularized RankCentrality even when the data is generated by a scoring function.

We constructed two synthetic datasets. We assume that the BTL-score is given by a continuous function of the features; i.e., there is an $f : \mathbb{R}^d \to \mathbb{R}$ so that the BTL score $w_i = f(x_i)$. This intuitively captures the idea that items which are close in space are close in rank. We consider a few examples of such functions f as given below.

- In Experiment A, we generated 1600 points $\{x_i\}_{i=1}^{1600}$ chosen uniformly at random from $[0, 4]^2$, we chose $\omega_1, \omega_2, \ldots, \omega_4 \in \mathbb{R}^2$ at random, each entry chosen independently from a Gaussian. To each $i \in [1600]$ we associate a score $w_i = \sum_{h=1}^{2} \exp(\cos(5\omega_h^T x_i)) + \sum_{h=3}^{4} \exp(\omega_h^T x_i/10)$.
- In Experiment B, we generated 1000 points $\{x_i\}_{i=1}^{1000} \in [0, 4]$ chosen uniformly at random and chose $\omega \in \mathbb{R}$ at random from a Gaussian. To each $i \in [1000]$ we associate a score $w_i = \exp(\cos(5\omega x_i))$.

For varying of m, we simulated m observations under the BTL-model with uniform μ and ran various algorithms that have been discussed. We recorded plotted the average Kendal-tau correlation metric (see Section D in the supplementary for details) between the ranking on the synthetic scores we generated and the true ranking on the items. The results of these experiments are summarized in Figures 3 and 4.

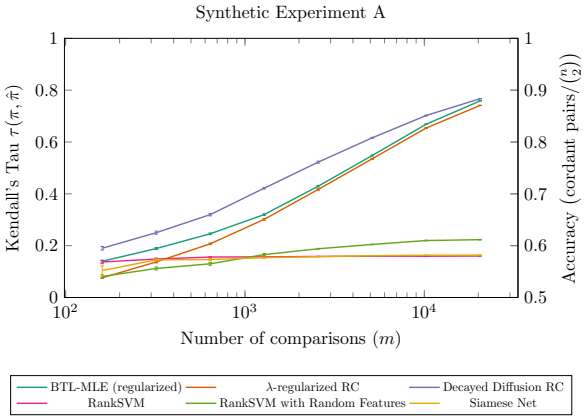

Figure 3: Comparison of algorithms in synthetic experiment A. Diffusion RankCentrality was run with kernel width $\sigma = 2^{-4}$.

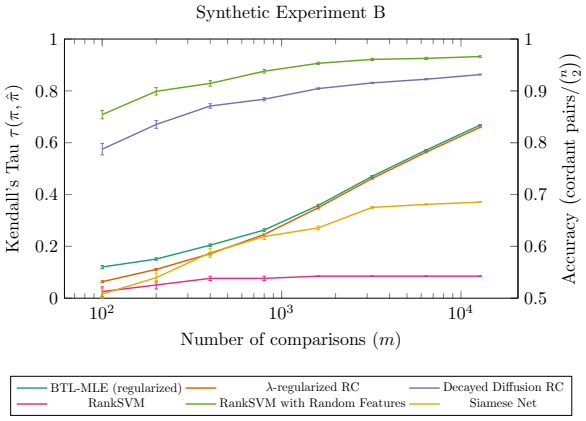

Figure 4: Comparison of algorithms in synthetic experiment B. Diffusion RankCentrality was run with kernel width $\sigma = 2^{-5}$.

In Experiment A, Diffusion RankCentrality proves to be the best method when the comparisons are scarce. The impact of Diffusion RankCentrality in Experiment B is dramatic when compared to λ-regularized RankCentrality. While it is true that RankSVM with random features far outperforms other algorithms, it should not come as a surprise given that the BTL scores w_i, as a function of x_i, come from monotonic transformations of linear combinations of the basis of the RKHS used for the im-

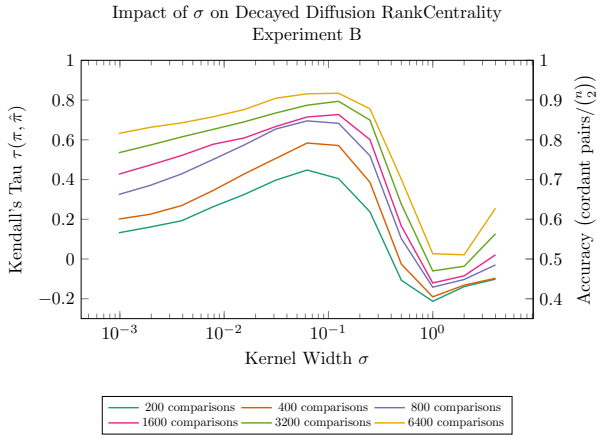

Figure 5: Impact of kernel width on performance of Diffusion RankCentrality.

plementation of random Fourier Features in scikit-learn (Pedregosa et al. 2011).

In both experiments, Diffusion RankCentrality outperforms Siamese Networks. To choose the kernel width, we ran Decayed Diffusion RankCentrality with several different choices of σ on a validation set and chose the best one (see Figure 5).

5.2 NEW YORKER CAPTION COMPETITION

It is challenging to find real-life data sets that satisfy all of the following conditions: 1) The data is structured; i.e., has image or text features associated with the items and 2) the number of items compared is moderate to large in size.

The New Yorker Caption Competition dataset consists of a cartoon and a series of associated (supposedly) funny captions submitted by readers (see (NEXTML 2019) for details on this dataset). Each week, readers vote on whether they think each caption is funny (2 points), somewhat funny(1 point) or unfunny (0 points), and the caption is assigned an average cardinal score based on these points. Included in this dataset are only two contests (#508 and #509), in which there are a large number of pairwise comparisons in addition to cardinal scores generated from user votes on a small number of items ($n = 29$ items for each contest). Each pair of items received roughly 300 comparisons and each item also received roughly 200 cardinal votes. (The associated captions and visuals of the query types are given in Figure 6, and Figure 13 in the supplementary material). Run directly on this dataset, Diffusion Rank Centrality did not show an appreciable advantage since the number of items was so small and hence similarity information provided less leverage over other methods.

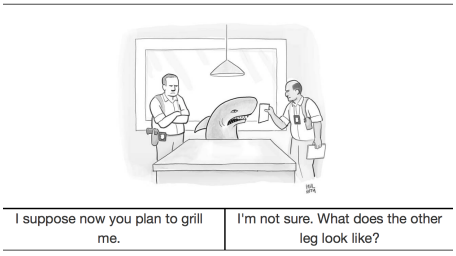

Figure 6: New Yorker Caption Competition Interface for pairwise comparisons for #508. Users were asked to click on the caption they thought was funnier.

Figure 7: A sample of the voting user interface presented to readers of the New Yorker Magazine for contest #651

5.2.1 Cardinal Scores model BTL-scores

We generate comparisons on a much larger set of captions for a different contest by transforming the cardinal data to infer pairwise comparisons. To determine this transformation, we used contest #508 for which we had 300 pairwise comparisons and 200 cardinal votes. For each pair of captions i, j in contest #508, we compute $\hat{P}_{ij}^{\text{emp}}$, the empirical probability of item i beating item j. In addition, we used the average empirical cardinal scores of items i and j denoted as $\hat{s}_i, \hat{s}_j$ we computed $\hat{P}_{ij}^{\text{card}} = \exp(\hat{s}_i)/(\exp(\hat{s}_i) + \exp(\hat{s}_j))$. In other words, we calculated the empirical probabilities implied by the cardinal scores and compared them to the empirical probabilities from the pairwise comparisons. A resulting scatterplot of the points $(\hat{P}_{ij}^{\text{emp}}, \hat{P}_{ij}^{\text{card}})$ is shown in Figure 8. Somewhat surprisingly, this plot demonstrates that a monotonic transformation of the cardinal scores seem to model an underlying pairwise probability model fairly well—implying that up to an exponential scaling transformation, the cardinal scores determine underlying BTL scores for the captions. This seems to be an interesting non-trivial result about ranking and humor that has not been previously observed.

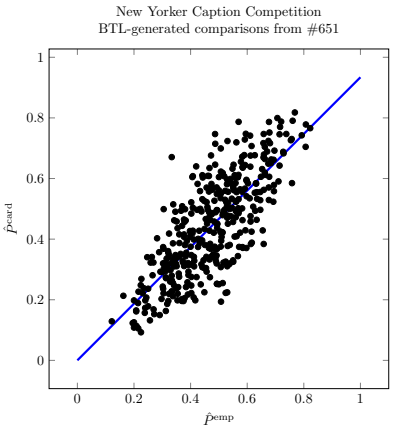

Figure 8: Scatter plot demonstrating the relationship between $\hat{P}^{\text{emp}}$ and $\hat{P}^{\text{card}}$.

5.2.2 Contest #651

Using the observations in the previous section, we chose a contest, #651, that did not have underlying pairwise comparisons but did have a large number of items all with cardinal scores. We then generated pairwise comparisons from these cardinal scores as described in Section 5.2.1. The cartoon associated to this contest is in Figure 9.

More precisely, from the captions available, we took the 400 captions (out of roughly 7000) with largest empirical average cardinal score (each caption had around 250 votes) and generated BTL weights. We used the Universal Sentence Encoder in (Cer et al. 2018) to generate 512 dimensional embeddings for each of the captions (this yields the additional structural information we need for regularization). The resulting plot contrasting the methods is shown in 7, as before the kernel width was chosen on a validation set—in addition we used $(1 - \frac{1}{\sqrt{m}})I + \frac{1}{\sqrt{m}}D^{(\sigma)}$ as the regularizer in Diffusion RankCentrality to debias the procedure.

In this setting, Diffusion RankCentrality performs extremely well, locking in a significantly better ranking almost immediately with few comparisons.

5.3 PLACE PULSE

Our final example involves comparisons arising from the Place Pulse dataset used in (Katariya et al. 2018). There were 100 images of locations in Chicago in this dataset, and a total of 5750 comparisons where MTurk workers were asked which of the two locations they thought were safer. We used ResNetV1 (He et al. 2016) to generate features for the images of each location and broke the data up into a train, test and validation set (again used to select σ and λ). Since we do not have an underlying ground truth ranking, we instead plot the test error in Figure 10.

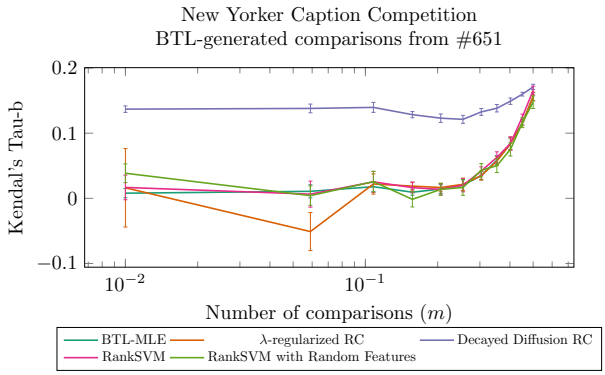

Figure 9: Test Error for various algorithms for the New Yorker Caption Competition #651 with $\sigma = .25$.

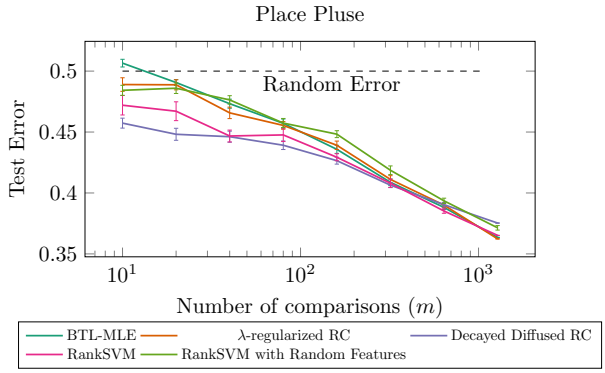

Figure 10: Performance of various algorithms from the Place Pulse dataset.

Again, Diffusion RankCentrality (a non-classification based method) performed competitively matching the performance of RankSVM.

6 CONCLUSION

In this paper we provided a way to employ structure in the RankCentrality algorithm that provides meaningful results when data is scarce. Along the way we provided a stronger sample complexity bound for a natural sampling scheme. For future work we hope to provide rigorous sample complexity bounds for diffusion based methods.

Acknowledgements

The first and third authors were supported by the MIDAS Challenge Grant from the University of Michigan. The first author had the initial idea and motivation for this work while at Agero, Inc., and would like to thank Michael Bell.

References

Agarwal, Arpit, Prathamesh Patil, and Shivani Agarwal (2018). "Accelerated spectral ranking". In: *International Conference on Machine Learning*, pp. 70–79.

Bromley, Jane et al. (1994). "Signature Verification using a "Siamese" Time Delay Neural Network". In: *Advances in Neural Information Processing Systems 6*. Ed. by J. D. Cowan, G. Tesauro, and J. Alspector. Morgan-Kaufmann, pp. 737–744.

Burges, Christopher et al. (2005). "Learning to rank using gradient descent". In: *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pp. 89–96.

Cer, Daniel et al. (2018). "Universal sentence encoder". In: *arXiv preprint arXiv:1803.11175*.

Chen, Yuxin et al. (Aug. 2019). "Spectral method and regularized MLE are both optimal for top-K ranking". In: *Ann. Statist.* 47.4, pp. 2204–2235.

Chollet, François et al. (2015). *Keras*. `https://keras.io`.

Coifman, R. R. et al. (May 2005). "Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.21, pp. 7426–7431. ISSN: 0027-8424.

Dijk, David van et al. (July 2018). "Recovering Gene Interactions from Single-Cell Data Using Data Diffusion". In: *Cell* 174.3, 716–729.e27. ISSN: 0092-8674.

He, K. et al. (June 2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Heckel, Reinhard et al. (Dec. 2019). "Active ranking from pairwise comparisons and when parametric assumptions do not help". In: *Ann. Statist.* 47.6, pp. 3099–3126.

Jamieson, Kevin G and Robert Nowak (2011). "Active Ranking using Pairwise Comparisons". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., pp. 2240–2248.

Joachims, Thorsten (2002). "Optimizing Search Engines Using Clickthrough Data". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: ACM, pp. 133–142. ISBN: 1-58113-567-X.

Katariya, Sumeet et al. (2018). "Adaptive Sampling for Coarse Ranking". In: *International Conference on Artificial Intelligence and Statistics*, pp. 1839–1848.

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). "Matrix factorization techniques for recommender systems". In: *Computer* 8, pp. 30–37.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, pp. 2579–2605.

Negahban, Sahand, Sewoong Oh, and Devavrat Shah (2016). "Rank centrality: Ranking from pairwise comparisons". In: *Operations Research* 65.1, pp. 266–287.

NEXTML (2019). *Data from the New Yorker Caption Contest*. URL: `https://github.com/nextml/caption-contest-data`.

Norris, J.R. (1998). *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. ISBN: 9781107393479.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Rahimi, Ali and Benjamin Recht (2008). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt et al. Curran Associates, Inc., pp. 1177–1184.

Rajkumar, Arun and Shivani Agarwal (2014). "A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data". In: *Proceedings of the 31st International Conference on Machine Learning*.

Shah, Nihar B, Sivaraman Balakrishnan, Joseph Bradley, et al. (2016). "Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence". In: *The Journal of Machine Learning Research* 17.1, pp. 2049–2095.

Shah, Nihar B, Sivaraman Balakrishnan, and Martin J Wainwright (2018). "Low permutation-rank matrices: Structural properties and noisy completion". In: *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 366–370.

Shah, Nihar B and Martin J Wainwright (2017). "Simple, robust and optimal ranking from pairwise comparisons". In: *The Journal of Machine Learning Research* 18.1, pp. 7246–7283.

Tropp, Joel A. (Aug. 2012). "User-Friendly Tail Bounds for Sums of Random Matrices". In: *Foundations of Computational Mathematics* 12.4, pp. 389–434. ISSN: 1615-3383.