Efficient Ranking from Pairwise Comparisons

Fabian L. Wauthier FLW@CS.BERKELEY.EDU
Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division, University of California, Berkeley, CA 94720, USA

Nebojsa Jojic JOJIC@MICROSOFT.COM
Microsoft Research, Redmond, WA 98052, USA

Abstract

The ranking of n objects based on pairwise comparisons is a core machine learning problem, arising in recommender systems, ad placement, player ranking, biological applications and others. In many practical situations the true pairwise comparisons cannot be actively measured, but a subset of all $n(n-1)/2$ comparisons is passively and noisily observed. Optimization algorithms (e.g., the SVM) could be used to predict a ranking with fixed expected Kendall tau distance, while achieving an $\Omega(n)$ lower bound on the corresponding sample complexity. However, due to their centralized structure they are difficult to extend to online or distributed settings. In this paper we show that much simpler algorithms can match the same $\Omega(n)$ lower bound in expectation. Furthermore, if an average of $O(n\log(n))$ binary comparisons are measured, then one algorithm recovers the true ranking in a uniform sense, while the other predicts the ranking more accurately near the top than the bottom. We discuss extensions to online and distributed ranking, with benefits over traditional alternatives.

1. Introduction

Ranking from binary comparisons is a ubiquitous problem in modern machine learning applications. Given a set of n objects and set of (possibly inconsistent) binary comparisons between pairs of objects (such as "player i won against player j," or "the customer bought book i instead of j"), the task is to infer a total order over objects that aggregates the given measurements. Common settings for this problem allow binary comparisons to be measured either actively (Ailon, 2012; Ailon et al., 2011; Jamieson & Nowak, 2011; Braverman & Mossel, 2009; Giesen et al., 2009), repeatedly (Negahban et al., 2012; Ammar & Shah, 2011; Feige et al., 1994), or assume that all $n(n-1)/2$ comparisons are known up to some noise (Braverman & Mossel, 2008; 2009). We believe that in many challenging applications, these assumptions are unrealistic: (1) Active measurements are often infeasible, either because measurements must be made passively (e.g., from click-through data, purchasing preferences), or because pairwise comparisons are too time consuming to measure in series (e.g., measuring protein-protein interactions). (2) Repeated measurements are not practical if comparisons are derived from the outcomes of sports games or the purchasing behavior of a customer (a customer typically wants to purchase a product only once). (3) The $O(n^2)$ growth of comparisons between n objects usually prohibits exhaustive measuring when n is large.

Since a total order can be uniquely determined by sorting distinct object "scores," it is common to formalize the problem as follows: Given a subset of (possibly noisy) binary comparisons $\bar{c}_{i,j}$ between n objects, we desire a scoring function $\hat{\Pi} : \{1,\dots,n\} \to \mathbb{R}$ so that $\bar{c}_{i,j} = 1 \iff \hat{\Pi}(i) < \hat{\Pi}(j)$ for as many examples in the training data as possible. Traditional optimization losses targeting this objective are intuitive (e.g., count the number of inversions between the training data and the scoring function,) but discontinuous and non-convex. The substantial literature on learning to rank can be specialized to this setting by learning scoring functions that only depend on the object identity. This approach suggests ways to approximately solve the optimization problem by relaxing the intractable loss to convex surrogates (Dekel et al., 2004; Freund et al., 2003; Herbrich et al., 2000; Joachims, 2006).

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

Although some of these methods (e.g., the SVM) can achieve an $\Omega(n)$ lower bound on a certain sample complexity, we feel that optimization-based approaches may be unnecessarily complex in this situation. The question arises whether simpler algorithms could be equally effective. In this paper we demonstrate that two very simple algorithms achieve the same $\Omega(n)$ lower bound without solving an explicit optimization problem. Furthermore, given slightly more measurements, we can show interesting differences between the two algorithms: The first predicts rankings with approximately uniform quality across the ranking, while the second predicts the true ranking with higher quality near the top of the ranking than the bottom. Additionally, we view the simple form of the algorithms as a significant asset which makes them much easier to extend. As a demonstration, we discuss extensions to online and distributed learning, and highlight important benefits over traditional alternatives.

The paper is organized as follows: We first introduce some notation and quality measures in Section 2. In Section 3 we discuss related research and background. Section 4 presents two simple ranking algorithms and analyzes their performance in terms of the expected Kendall tau distance as well as high probability bounds on rank displacements. In Section 5 we evaluate and validate our theoretical findings. We touch on extensions to online and distributed ranking in Section 6, before concluding with final thoughts in Section 7. The complete proofs for all propositions, lemmas and theorems are collected in the supplementary material.

2. Preliminaries

Throughout the paper we denote the true permutation we wish to recover by $\pi^* \in S_n$. We use the notation $\pi(i)$ to indicate the position of object i in permutation π. Without loss of generality, let $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. We will reveal to an algorithm a subset of binary comparisons, chosen among the $n(n-1)/2$ available pairs. Specifically, each comparison is measured independently with probability $m(n)/n$, so that on average $O(nm(n))$ measurements are made. Each comparison can be measured only once (i.e. we measure *without* replacement)[1]. The function $m(n)$ is a key quantity; we will characterize various sample complexities in terms of bounds on its growth. For some results we will find that $m(n) \in \Theta(1)$ suffices, while in others we need $m(n) \in \Theta(\log(n))$. We will always assume that $m(n) \in o(n)$. Noiseless binary comparisons are denoted by $c_{i,j} = \mathbf{1}(\pi^*(i) < \pi^*(j))$. A common observation model is to assume that each binary comparison is independently flipped with probability $1-p$, where $p > 1/2$ (Braverman & Mossel, 2008; Feige et al., 1994). To capture the overall measurement process, we introduce binary variables $s_{i,j}$ which indicate whether $c_{i,j}$ was measured, and let $\bar{c}_{i,j}$ be the (possibly noisy) measurement that was made. We will assume throughout that $s_{j,i} = s_{i,j}$ and if $s_{j,i} = s_{i,j} = 1$, then $\bar{c}_{j,i} = 1 - \bar{c}_{i,j}$.

[1] In some other analyses measurements are made independently with replacement (Radinsky & Ailon, 2011; Mitliagkas et al., 2011).

In this paper we analyze the quality of the proposed algorithms in two ways. The first counts the number of inverted binary comparisons of the predicted permutation $\hat{\pi}$ relative to π^*. That is, we use the loss

$$\mathrm{inv}(\hat{\pi}) = \sum_{\pi^*(i)<\pi^*(j)} \mathbf{1}(\hat{\pi}(j) < \hat{\pi}(i)). \tag{1}$$

This quantity is also known as the *Kendall tau distance*. Using results in Fulman (2004), one can show that if $\hat{\pi}$ is chosen uniformly at random in S_n, then $\mathrm{inv}(\hat{\pi})$ concentrates around $(1/2)(n(n-1)/2)$. To be interesting we will thus require our algorithms to have expected risk $\mathbb{E}(\mathrm{inv}(\hat{\pi})) \leq (\eta/2)(n(n-1)/2)$ for some $0 < \eta < 1$. We note that another common comparison metric is Spearman's footrule

$$\mathrm{dis}(\hat{\pi}) = \sum_{j=1}^{n} |\hat{\pi}(j) - \pi^*(j)|. \tag{2}$$

As shown in Diaconis & Graham (1977), $\mathrm{inv}(\hat{\pi})$ is related to $\mathrm{dis}(\hat{\pi})$ as $\mathrm{inv}(\hat{\pi}) \leq \mathrm{dis}(\hat{\pi}) \leq 2\mathrm{inv}(\hat{\pi})$. Our results on the expected Kendall tau distance thus directly transfer to Spearman's footrule. We also analyze the prediction $\hat{\pi}$ by how far individual objects are displaced relative to π^*. When appropriate, we will bound the largest displacement

$$\max_j |\hat{\pi}(j) - \pi^*(j)|. \tag{3}$$

However, in some cases the recovery is not uniform, warranting a detailed inspection of the set of displacements $\{|\hat{\pi}(j) - \pi^*(j)| : j = 1, \ldots, n\}$.

3. Related Research

Several threads of research aim to give various sample complexities in the active ranking setting. Ailon et al. (2012), for example, give an active algorithm which produces a permutation with small loss relative to the optimal loss (which may be zero). This result was refined by Ailon et al. (2011) to show that if the true scoring function is linear, one can find a scoring function with small loss (relative to the optimal loss) using $O(n \log^4(n))$ active queries. Braverman and Mossel (2009) give an active algorithm with

query complexity $O(n\log(n))$ for noisy binary comparisons that produces a ranking in time that is with high probability polynomial. Agarwal (2005) has developed a comprehensive theory for bipartite ranking. Here, instead of receiving binary comparisons, we receive binary labels (e.g., relevant/irrelevant) for each object, and the goal is a scoring function which orders negative before positive examples.

A number of recent papers have analyzed lower bounds for the demanding task of *exact* score recovery. Jamieson and Nowak (2011), for example, consider the case when the true scoring function reflects the Euclidean distance of object covariates from a global reference point. If objects are embedded in $\mathbb{R}^d$, then any algorithm that exactly identifies the true ranking must sample at least $O(d\log(n))$ comparisons. While this bound can be achieved by an active algorithm, any algorithm that uses only random measurements must see almost all pairwise comparisons in order to exactly predict the true ranking. Gleich and Lim (2011) suppose that the true score differences (i.e., $\Pi^*(j)-\Pi^*(i)$, or functions thereof) can be measured. Given an incomplete matrix of such measurements they use low rank matrix completion to estimate the true object scores. If the measurements are in fact score differences, their algorithm recovers the true scores with high probability exactly using between $O(n\log^2(n))$ and $O(n^2\log^2(n))$ random measurements (depending on the shape of the true scores). Although their work considers random measurements, their theory does not apply when binary comparisons are measured in lieu of score differences. Mitliagkas et al. (2011) focus on exactly recovering the preferences expressed by a population of r users. Each user's preferences are recorded by a permutation over objects, which can be queried (either actively or by random sampling) through pairwise comparisons between objects. The randomized sampling result is not helpful in our setting (where $r = 1$) since it then requires $O(n^2\log(n))$ measurements (with replacement) for exact recovery.

SVM Ranking. It is well-known that the SVM could be used to learn a linear scoring function in the setting of Section 2: For each observed comparison $\bar{c}_{i,j}$, create a feature vector $x_{i,j} = e_j - e_i$ (where e_i is a binary indicator vector with a 1 at the i-th coordinate) and associate with it the label $\bar{y}_{i,j} = 2\bar{c}_{i,j}-1$. Learning a scoring function now reduces to inferring a separating hyperplane w so that the function $\text{sign}(w^\top x_{i,j})$ best predicts the labels $\bar{y}_{i,j}$ on training data. The predicted permutation $\hat{\pi}$ follows from sorting the elements in w. Statistical learning theory shows that in the noiseless case ($p = 1$), the sample complexity for inferring a w which with high probability induces a Kendall tau distance of at most $(\eta/2)(n(n-1)/2)$ is small. Indeed, using results of Radinsky et al. (2011) one can show the following proposition, which we prove in the supplementary material

Proposition 3.1. *There is a constant d, so that for any $0 < \eta < 1$, if we noiselessly measure dn/η^2 binary comparisons, chosen uniformly at random with replacement, and $n > n_0$ is large enough, the SVM will produce a prediction $\hat{\pi}$, which satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{4}$$

The proposition highlights that the SVM needs to sample $\Theta(n/\eta^2)$ examples *with* replacement for an expected risk of at most $(\eta/2)(n(n-1)/2)$. Some algebra then reveals that this amounts to an average of $O(n)$ distinct samples. As the following proposition, a summary of results of Giesen et al. (2009), demonstrates, the sample complexity of Proposition 3.1 is tight up to constants.

Proposition 3.2. *For $\eta < 1$, any randomized, comparison-based algorithm that produces for all π^* a prediction $\hat{\pi}$ with an expected risk of*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2} \tag{5}$$

must on expectation use at least $\Omega(n)$ comparisons in the worst case.

The proposition is proved in the supplementary material for completeness. Although the SVM is effectively optimal in this setting, we feel that its direct application is overly heavy handed. The goal of this paper is to exhibit two much simpler algorithms which also achieve the above sample complexity, while being easier to extend to novel applications.

4. Two Simple Algorithms

In this section we present two simple rank estimators using the randomized data collection framework outlined in Section 2.

4.1. Balanced Rank Estimation

We begin this paper by analyzing BRE, which estimates an object's score as the relative difference of the number of items preceding and succeeding it.

Balanced Rank Estimation (BRE):
Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{\sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)}{2m(n)} \propto \sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)\,.$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Our first result concerns the expected number of inversions of $\hat{\pi}$ relative to π^*.

Theorem 4.1. *For any $0 < \eta < 1$ there is a constant $c(p,\eta) \in \Theta(1/((2p-1)^2\eta^2))$ so that if $m(n)/n \geq c(p,\eta)/n$, and $n > n_0$ is large enough, BRE satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{6}$$

To give some intuition for this theorem, we briefly sketch the proof. Since we assumed $\pi^* = (1,\ldots,n)$, the expected Kendall tau distance is

$$\mathbb{E}(\mathrm{inv}(\hat{\pi})) = \sum_{i<j} \mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right). \tag{7}$$

The score difference $\hat{\Pi}(i) - \hat{\Pi}(j)$ can be written as a sum of $2n-3$ independent random variables. By controlling their mean, variance and magnitude, if $n > n_0$ is large the following bound can be derived for $i < j$:

$$\mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right) \tag{8}$$

$$\leq \exp\left\{-\left[\frac{j-i}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\}. \tag{9}$$

Applying this to Eq. (7), we bound $\mathbb{E}(\mathrm{inv}(\hat{\pi}))$ by

$$\sum_{k=1}^{n-1}(n-k)\exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} \tag{10}$$

$$\leq \int_0^n (n-k)\exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} dk \tag{11}$$

$$\leq \frac{n}{n-1}\sqrt{\frac{128}{3}}\frac{1}{(2p-1)\sqrt{m(n)}}\binom{n}{2}. \tag{12}$$

Matching this upper bound with the target quantity $(\eta/2)(n(n-1)/2)$, we find $m(n) \in \Theta(1/((2p-1)^2\eta^2))$.

In the noiseless case ($p = 1$), Theorem 4.1 guarantees that for any $0 < \eta < 1$, BRE in expectation has the same sample complexity as the SVM in Proposition 3.1. In particular, BRE also achieves the $\Omega(n)$ lower bound of Proposition 3.2. This may seem at first surprising. However, a similar algorithm was recently shown to have favorable properties in a different context (Coppersmith et al., 2010).

More informative statements can be made if a slightly larger number measurements is available. As the following theorem shows, given an average of $\Theta(n\log(n))$ measurements, BRE predicts permutations with uniform quality across the entire permutation.

Theorem 4.2. *For any $c > 0$ and $0 < \nu < 1$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, then BRE predicts with probability at least $1 - 2n^{1-a_n\frac{3}{8}(2p-1)^2\nu^2 c}$ a permutation $\hat{\pi}$ with*

$$\max_j |\hat{\pi}(j) - \pi^*(j)| \leq \nu n, \tag{13}$$

where a_n is a sequence with $a_n \to 1$.

The crux of the argument is that the estimated scores $\hat{\Pi}(j)$ concentrate around their expectation $\tilde{\Pi}^*(j) \triangleq \mathbb{E}(\hat{\Pi}(j)) = aj/n + b$, where $a = (2p-1)$ and $b \in \mathbb{R}$ (as before we assume $\pi^* = (1,\ldots,n)$). If all scores concentrate uniformly well, they will reveal the true permutation up to local displacements. Using a similar analysis as in Theorem 4.1, our proof first establishes the following Bernstein concentration:

$$\mathbb{P}\left(\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{14}$$

$$\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)}\right\}, \tag{15}$$

to which we then apply a union bound (introducing the $\log(n)$ factor)

$$\mathbb{P}\left(\exists j : \left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{16}$$

$$\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)} + \log(n)\right\}. \tag{17}$$

Thus, the relative ordering of two objects that are far apart in the π^* (large t) should be harder to confuse than that of nearby objects (small t). Indeed, using the following intuitive lemma, the uniform concentration of scores translates into a uniform bound on displacements $|\hat{\pi}(j) - \pi^*(j)|$.

Lemma 4.3. *For any $a > 0$, and $b \in \mathbb{R}$, if $\forall j$, we have $|\hat{\Pi}(j) - (aj/n + b)| \leq t$, then we have that $\forall j$, $|\hat{\pi}(j) - \pi^*(j)| \leq 2tn/a$.*

The lemma applies to the union bound with $a = (2p-1)$. The proof is then completed by setting $t = (2p-1)\nu/2$ and simplifying Eq. (17).

The following corollary immediately follows from Theorem 4.2 and highlights for what constants c the probability in Theorem 4.2 converges.

Corollary 4.4. *For $0 < \nu < 1$, there is a constant $c = c(p, \nu)$ with $2/((2p-1)^2\nu^2) < c(p, \nu) < 3/((2p-1)^2\nu^2)$, so that for BRE $\mathbb{P}(\max_j |\hat{\pi}(j) - \pi^*(j)| \leq \nu n) \to 1$.*

4.2. Unbalanced Rank Estimation

In many situations, we are not interested in learning the entire permutation accurately but only care about the highest (or lowest) ranked objects. The well-known discounted cumulative gain (Järvelin & Kekäläinen, 2002), for example, captures this notion and has been important in the information retrieval literature. More recently, Rudin (2009) proposed p-norms for ranking losses that penalize errors near the top more severely than in the tail of the list. The approach has been taken to the ∞-norm limit by Agarwal (2011). When n grows, the number of top elements we are interested in will typically also grow; in many natural phenomena, for example, we expect more extreme examples to appear as we make more observations. Suppose then, that for some $0 < \nu < 1$ we wish to recover the placement of the first νn elements in the permutation with fairly good accuracy, but care less about the remaining $(1-\nu)n$ elements. Surprisingly, a very slight modification of the Balanced Rank Estimation Algorithm yields a method that is useful in this situation. Furthermore, it still only requires a random subset of pairwise comparisons. The new algorithm, URE, estimates an object's score by the fraction of measured items preceding it.

Unbalanced Rank Estimation (URE):
Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{1}{m(n)} \sum_{i \neq j} s_{i,j} \bar{c}^n_{i,j} \propto \sum_{i \neq j} s_{i,j} \bar{c}^n_{i,j}.$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

To begin, we first establish that this algorithm in expectation still achieves the $\Omega(n)$ lower bound given in Proposition 3.2.

Theorem 4.5. *For any $0 < \eta < 1$, there is a constant $c(p, \eta) \in \Theta(1/((2p-1)^2\eta^2))$ so that if $m(n)/n \geq c(p, \eta)/n$, URE satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2} \binom{n}{2}. \tag{18}$$

Similar to Theorem 4.1, the proof relies on a tail inequality for the difference $\hat{\Pi}(i) - \hat{\Pi}(j)$. Supposing that $\pi^* = (1, \ldots, n)$, we show in the proof that for $i < j$

$$\mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right) \tag{19}$$

$$\leq \exp\left\{ -\left[\frac{j-i}{n}\right]^2 \frac{3}{100}(2p-1)^2 m(n) \right\}. \tag{20}$$

As in Theorem 4.1 we can use this to bound the Kendall tau distance as

$$\mathbb{E}(\text{inv}(\hat{\pi})) \leq \frac{n}{n-1}\sqrt{\frac{400}{3}} \frac{1}{(2p-1)\sqrt{m(n)}} \binom{n}{2}. \tag{21}$$

Finally, equating this upper bound with $(\eta/2)(n(n-1)/2)$ allows us to solve for $m(n) \in \Theta(1/((2p-1)^2\eta^2))$.

Theorem 4.5 guarantees in the noiseless case ($p = 1$) that for any $0 < \eta < 1$, URE in expectation achieves the same $\Theta(n/\eta^2)$ sample complexity as the SVM in Proposition 3.1.

Our main interest in URE, however, is encapsulated in the following theorem which shows that predicted permutations are much more accurate near the top than the bottom if an average of $\Theta(n \log(n))$ measurements are made instead[2].

Theorem 4.6. *For any $c > 0$, and $0 < \nu < 1$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, URE predicts with probability at least*

$$1 - 2n^{1 - \frac{3}{2}\left[(2p-1)^2\nu^2/(3(1-p)+(5p-1)\nu)\right]c} \tag{22}$$

a permutation $\hat{\pi}$ with

$$|\pi^*(j) - \hat{\pi}(j)| \leq \begin{cases} 4\nu n & \text{if } \pi^*(j) < \nu n \\ 4\sqrt{\nu \pi^*(j) n} & \text{if } \pi^*(j) \geq \nu n \end{cases}. \tag{23}$$

The proof parallels that of Theorem 4.2 and shows that $\hat{\Pi}(j)$ concentrates around its expectation $\tilde{\Pi}^*(j) \triangleq \mathbb{E}(\hat{\Pi}(j)) = aj/n + b$, with $a = (2p-1)$ and $b \in \mathbb{R}$ (again, we assume $\pi^* = (1, \ldots, n)$). However, while in Theorem 4.2 the tail bound was identical for each j, here the scores $\hat{\Pi}(j)$ have variances that depend on j. To build intuition, in the noiseless case ($p = 1$), since the first element $j = 1$ in π^* has no items preceding it (i.e., $\forall i \neq j$ $\bar{c}_{i,j} = c_{i,j} = \mathbf{1}(i < j) = 0$), the estimated score $\hat{\Pi}(j)$ will always be zero and have zero variance, regardless of how many elements we measure. For remaining elements, the mean of the estimated scores will progressively increase down the permutation, as will their variance. The increase in variance brings a decrease in their predictive accuracy,

[2] Of course, a minor modification of the algorithm leads to better estimation near the bottom.

which is reflected in the theory. Specifically, one can show that

$$\mathbb{P}\left(\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{24}$$

$$\leq 2\exp\left\{-\frac{t^2 m(n)}{2\left(\frac{j}{n}p + (1-p) + \frac{t}{3}\right)}\right\}. \tag{25}$$

Before applying a union bound to the above bounds, it is convenient to first eliminate the j-dependence of the upper bounds. To do this, we define the following set of increasing deviation events

$$A_j = \begin{cases} \left\{\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > \sqrt{\nu}t\right\} & \text{if } j < \nu n \\ \left\{\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > \sqrt{\frac{j}{n}}t\right\} & \text{if } j \geq \nu n. \end{cases} \tag{26}$$

Some algebra then gives, for all j,

$$\mathbb{P}(A_j) \leq 2\exp\left\{-\frac{\sqrt{\nu}t^2 m(n)}{2\left(\sqrt{\nu}p + \frac{1}{\sqrt{\nu}}(1-p) + \frac{t}{3}\right)}\right\}, \tag{27}$$

which yields the following union bound:

$$\mathbb{P}\left(\cup_{j=1}^n A_j\right) \leq 2n\exp\left\{-\frac{\sqrt{\nu}t^2 m(n)}{2\left(\sqrt{\nu}p + \frac{1-p}{\sqrt{\nu}} + \frac{t}{3}\right)}\right\}. \tag{28}$$

As in Theorem 4.2, we turn this concentration result into a bound on the rank displacement using a lemma.

Lemma 4.7. *For $a > 0$, $0 < \gamma < a^2$ and $b \in \mathbb{R}$, if*

$$\left|\hat{\Pi}(j) - \left(\frac{aj}{n} + b\right)\right| \leq \begin{cases} \gamma/a & \text{if } j < \gamma n/a^2 \\ \sqrt{\gamma j/n} & \text{if } j \geq \gamma n/a^2 \end{cases}, \tag{29}$$

then

$$|\hat{\pi}(j) - \pi^*(j)| \leq \begin{cases} 4\gamma n/a^2 & \text{if } j < \gamma n/a^2 \\ 4\sqrt{\gamma j n}/a & \text{if } j \geq \gamma n/a^2 \end{cases}. \tag{30}$$

The proof of the lemma shows that even if a sorting algorithm breaks ties in the least favorable way, the final rank positions cannot differ too much from the true positions in π^*. The main difficulty for this argument lies in a suitable definition of the sets A_j, which translates into the preconditions used for this lemma. As before, the lemma applies with $a = (2p-1)$. The result follows if for any $0 < \nu < 1$ we set $t = a\sqrt{\nu}$ in the definition of sets A_j, $\gamma = \nu a^2$ in Lemma 4.7, simplify Eq. (28) and then substitute $\pi^*(j)$ for j where appropriate.

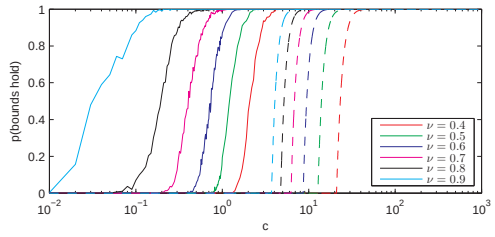
The following corollary, highlighting suitable constants c, follows immediately from Theorem 4.6.

Corollary 4.8. *For any $0 < \nu < 1$, there is a constant $c = c(p,\nu)$ with $2p/((2p-1)^2\nu) + 2(1-p)/((2p-1)^2\nu^2) \leq c(p,\nu) \leq 3p/((2p-1)^2\nu) + 2(1-p)/((2p-1)^2\nu^2)$, so that as $n \to \infty$ the displacement bounds of Theorem 4.6 hold with probability 1.*

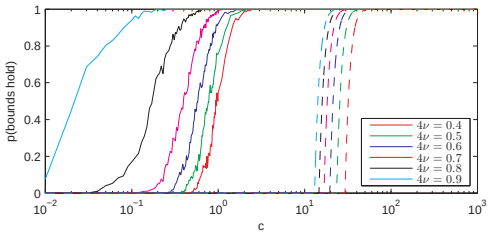
Discussion. In both Theorems 4.2 and 4.6 the size of the bins into which we correctly place objects can be decreased by increasing the number of measurements. If we consider the noiseless case ($p = 1$), Corollary 4.8 predicts that to place elements j with $\pi^*(j) < \nu n/2$ into bins half the current size, URE needs on average twice as many comparisons. To correctly place objects j with $\pi^*(j) \geq \nu n$ into bins of half the size URE needs on average four times as many measurements. From Corollary 4.4, we see that the behavior of the BRE is rather different. There, a four-fold increase is required to halve the bin sizes uniformly across the permutation. The cost of URE's improved performance near the top, however, is that for the same amount of data, the bin sizes in the tail are typically larger than those of BRE. Thus, if only the top elements are of interest, URE should be preferred. If a more uniform recovery is desired, BRE should be chosen. We will highlight this tradeoff in Section 5 with an example. An advantage in this regard is that the algorithm can be chosen *after* the data has been collected since BRE and URE work with the same type of input data. This fact could be exploited by combining the score estimators in various ways to further improve over the individual prediction results.

5. Experiments

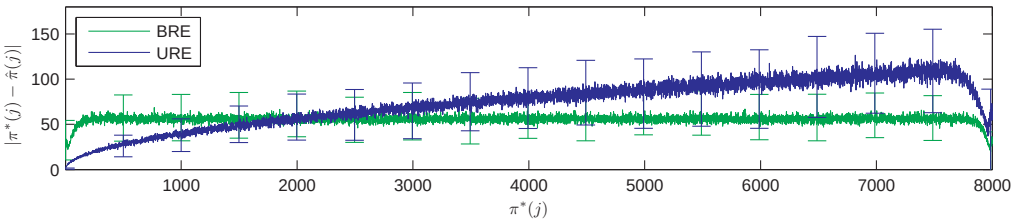
To begin, we empirically validate Theorems 4.2 and 4.6 in the noiseless case ($p = 1$). The theorems show that if each comparison is measured with probability $c\log(n)/n$, for some constant c, then the deviations $|\pi^*(j) - \hat{\pi}(j)|$ can be controlled with some probability that depends on c. In Figures 1(a) and 1(b) we show for particular choices of ν in solid the empirical probabilities that the displacement bounds of the theorems hold, as a function of the constant c. Additionally, we show the theoretical lower bounds on these probabilities, as given in Theorems 4.2 and 4.6. Notice that ν is four times smaller in Figure 1(b) than in Figure 1(a) so that Theorems 4.2 and 4.6 predict the same upper bounds on $|\pi^*(j) - \hat{\pi}(j)|$ for j s.t. $\pi^*(j) \leq \nu n$. Empirically, we see that in this case BRE requires more measurements than URE. To highlight the difference in prediction quality, we evaluated both algorithms on an 8000-object permutation. For each of 500 simulation runs, both algorithms saw *exactly* the same set of comparisons. In Figure 1(c) we show the median displacement $|\pi^*(j) - \hat{\pi}(j)|$ across the 500 runs, as a function of $\pi^*(j)$. Additionally, the error bars show 1/2 times the standard deviation of the displacements. For $\pi^*(j) < 2000$ URE predicts the correct position with higher accuracy and smaller variance than BRE. However, for large $\pi^*(j) \geq 2000$ BRE outperforms.



(a) BRE with lower bounds of Theorem 4.2.



(b) URE with lower bounds of Theorem 4.6.



(c) BRE and URE on the same set of comparisons. Different prediction characteristics are visible.

Figure 1. Empirical validation of Theorems 4.2 and 4.6. Figures (a) and (b) show for various ν in solid the empirical probabilities that the displacement bounds of Theorems 4.2 and 4.6 hold, if each comparison is measured independently with probability $c\log(n)/n$, as a function of c. To estimate these, we ran 300 noiseless simulations on permutations over 1000 objects and computed the fraction of times the bounds held. The empirical probabilities can be compared to the corresponding lower bounds produced by Theorems 4.2 and 4.6, which we plot as dashed curves. Figure (c) shows a direct comparison of our proposed algorithms. For each of 500 runs on an 8000-object permutation task, both algorithms saw *exactly* the same comparisons. Each plot shows the median displacement $|\pi^*(j) - \hat{\pi}(j)|$, as a function of $\pi^*(j)$.

6. Extensions

An important benefit of BRE/URE over active methods is that data collection can be trivially parallelized: Comparisons can be collected from independent processes, each measuring within a pre-assigned block of object pairs. Furthermore, the structure of the score estimators makes it easy to extend BRE/URE to several interesting settings. For one, we see applications in online ranking where we wish to grow rankings over n to $n+1$ objects as data streams in. Online versions of BRE/URE are easy to derive, yet lead to similar guarantees as those in Section 4. In contrast, the solutions of optimization-based methods can be non-trivial to update when the problem is slightly perturbed. Cauwenberghs and Poggio (2000), for example, show that the exact update to an SVM solution requires careful bookkeeping of dual coefficients. The simple structure of BRE/URE also makes them useful in distributed settings where costly coordination and communication among multiple processors can be avoided. We will now explore this extension.

6.1. Distributed Ranking

In many situations, the number n of objects being compared is large. For instance, online retailers can easily offer millions of products for sale among which comparisons could be made. In such situations the objects (data points) are often stored on a fixed number K of machines, so that each machine stores about $f = n/K$ data points. A consequence of this distributed storage is that the $O(nm(n))$ comparisons are likely to be collected on distinct machines. A naïve centralized ranking algorithm would collect the individual comparisons at a server for learning, incurring a communication cost of $O(nm(n))$. This cost is prohibitive if, relative to n, $m(n)$ is large[3]. Distributed, iterative SVM-type algorithms have been developed for such situations (Hazan et al., 2008; Graf et al., 2004) however, their application is typically complicated by the need for running multiple iterations which must be coordinated by locking protocols. As a result, the efficiency of these methods can rapidly deteriorate if a single machine fails. A favorable property of BRE/URE is that their simple form lends them much more naturally to distributed extensions, which can avoid locking protocols altogether. The main idea is

[3]This could be because for a particular problem size n the constant $c(p,\eta) \in \Theta(1/((2p-1)^2\eta^2))$ in Theorems 4.1 and 4.5 happens to be large, or because the probability p of correctly measuring a comparison decreases quickly as a function of n.

that the BRE/URE object scores can also be computed from partial scores rather than from individual comparisons. If the number of binary comparisons $O(nm(n))$ is large, then communicating partial scores can be much more efficient. We analyze this setting.

To compute comparisons, any algorithm must start by exchanging object encodings between the K machines. Let the data points allocated to machine k be D_k. There are $K(K-1)/2$ machine pairs $(k < l)$ that need to exchange $f = n/K$ data points from one computer to the other. Overall, this leads to $n(K-1)/2 \in O(nK) = O(n^2/f)$ data points being exchanged. Once the $O(nm(n))$ comparisons have been computed (in distributed fashion), we aggregate them into partial scores. Specifically, denote the set of binary comparisons created by a machine pair $k \leq l$ by

$$\bar{C}_{k,l} = \{\bar{c}_{i,j} : i \in D_k, j \in D_l, s_{i,j} = 1\}. \tag{31}$$

Because $\bar{c}_{j,i} = 1 - \bar{c}_{i,j}$ if $s_{j,i} = s_{i,j} = 1$, the set $\bar{C}_{l,k}$ can easily be computed from $\bar{C}_{k,l}$. In the following we will assume that $\bar{C}_{l,k}$ has been implicitly computed in this way whenever necessary. For BRE, use $\bar{C}_{k,l}$ to compute for each pair k, l the following partial scores

$$\hat{\Pi}_{k,l}(j) = \sum_{i \in D_k} s_{i,j}(2\bar{c}_{i,j} - 1) \quad \forall j \in D_l. \tag{32}$$

This amounts to a total of $K^2 f = n^2/f$ partial scores. The partial scores for URE follow a similar strategy. To complete the algorithm, the partial scores must be communicated to a central machine at cost $O(n^2/f)$. If $l(j)$ is the machine index $l \in \{1, \ldots, K\}$ so that $j \in D_l$, we combine the partial scores as

$$\hat{\Pi}(j) = \sum_{k=1}^{K} \hat{\Pi}_{k,l(j)}(j). \tag{33}$$

The overall communication time is $O(n^2/f)$. In comparison, a naïve centralized algorithm requires communication time $O(nm(n))$. If $m(n) \in O(1/((2p-1)^2\eta^2)) \gg K$ then our proposed algorithm significantly reduces the communication time. For practical applications, the number of machines K is typically less than 100. In this case the our algorithm should be a viable alternative to centralized optimization schemes with η as large as $\eta = 0.1$.

7. Conclusions

This paper analyzed two simple algorithms for ranking n objects from a random sample of binary comparisons. We showed that the algorithms in expectation achieve a lower bound on the sample complexity for predicting a ranking with fixed expected Kendall tau distance. As such, they are competitive alternatives to the SVM, which also achieves the lower bound. By giving the algorithm slightly more measurements, we showed that interesting displacement bounds between $\hat{\pi}$ and π^* can be derived.

Because the algorithms rely only on a random subset of pairwise comparisons, data collection can be trivially parallelized. The simple structure of the scoring functions makes them easy to adapt to new situations, such as online or distributed ranking. We showed that in the latter case the communication cost of a traditional centralized optimization approach can be substantially reduced if $(2p-1)^2\eta^2$ is sufficiently small.

This paper has exclusively considered scoring functions $\Pi(j)$ that only depend on the object identity. However, BRE and URE can act as a useful performance baseline even for learning parametric scoring functions, as frequently considered: If the in-sample empirical performance of such parametric ranking functions is worse than that predicted by Theorems 4.1 and 4.5, the function class may need to be redesigned or more data collected. Moreover, the two algorithms can be used as quick, general-purpose preprocessing algorithms for conventional ranking methods: A small subset of pairwise comparisons can be approximately completed using BRE or URE, irrespective of the true (possibly parametric) ranking function that generated them. This larger set of comparisons could then be useful in learning an improved parametric ranking function.

References

Agarwal, S. *A Study of the Bipartite Ranking Problem in Machine Learning*. PhD thesis, University of Illinois at Urbana-Champaign, 2005.

Agarwal, S. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *Proceedings of the SIAM International Conference on Data Mining*, 2011.

Ailon, N. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13:137–164, 2012.

Ailon, N., Begleiter, R., and Ezra, E. A new active learning scheme with applications to learning to rank from pairwise preferences. *arXiv CoRR*, abs/1110.2136, 2011.

Ammar, A. and Shah, D. Ranking: Compare, don't score. In *Proceedings of the 49th Annual Allerton*

Conference on Communication, Control and Computing (Allerton), pp. 776–783. 2011.

Braverman, M. and Mossel, E. Noisy sorting without resampling. In *Symposium on Discrete Algorithms*, pp. 268–276, 2008.

Braverman, M. and Mossel, E. Sorting from noisy information. *arXiv CoRR*, abs/0910.1191, 2009.

Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. In Leen, T.K., Dietterich, T.G., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13 (NIPS)*, pp. 409–415. MIT Press, 2000.

Coppersmith, D., Fleischer, L., and Rudra, A. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Transactions on Algorithms*, 6(3):55:1–55:13, 2010.

Dekel, O., Manning, C., and Singer, Y. Log-linear models for label ranking. In Thrun, S., Saul, L., and Schölkopf, B. (eds.), *Advances in Neural Information Processing Systems 16 (NIPS)*. MIT Press, 2004.

Diaconis, P. and Graham, R. L. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

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

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

Fulman, J. Stein's method, Jack measure, and the Metropolis algorithm. *Journal of Combinatorial Theory. Series A*, 108(2):275–296, 2004.

Giesen, J., Schuberth, E., and Stojaković, M. Approximate sorting. *Fundamenta Informaticae*, 90(1-2): 67–72, 2009.

Gleich, D. F. and Lim, L. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 60–68, 2011.

Graf, H. P., Cosatto, E., Bottou, L., Durdanovic, I., and Vapnik, V. Parallel support vector machines: The cascade SVM. In Saul, L.K., Weiss, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 17 (NIPS)*. MIT Press, 2004.

Hazan, T., Man, A., and Shashua, A. A parallel decomposition solver for SVM: Distributed dual ascend using Fenchel duality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

Herbrich, R., Graepel, T., and Obermayer, K. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pp. 115–132. MIT Press, 2000.

Jamieson, K. G. and Nowak, R. Active ranking using pairwise comparisons. In Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 24 (NIPS)*, pp. 2240–2248. MIT Press, 2011.

Järvelin, K. and Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

Joachims, T. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 217–226, 2006.

Mitliagkas, I., Gopalan, A., Caramanis, C., and Vishwanath, S. User rankings from comparisons: Learning permutations in high dimensions. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control and Computing (Allerton)*, 2011.

Negahban, S., Oh, S., and Shah, D. Iterative ranking from pair-wise comparisons. In Bartlett, P., Pereira, F., Burges, C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25 (NIPS)*, pp. 2483–2491. MIT Press, 2012.

Radinsky, K. and Ailon, N. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In King, I., Nejdl, W., and Li, H. (eds.), *Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 105–114. ACM, 2011.

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