Approximate ranking from pairwise comparisons

Reinhard Heckel
Rice University

Max Simchowitz
UC Berkeley

Kannan Ramchandran
UC Berkeley

Martin J. Wainwright
UC Berkeley

Abstract

A common problem in machine learning is to rank a set of n items based on pairwise comparisons. Here ranking refers to partitioning the items into sets of pre-specified sizes according to their scores, which includes identification of the top-k items as the most prominent special case. The score of a given item is defined as the probability that it beats a randomly chosen other item. Finding an exact ranking typically requires a prohibitively large number of comparisons, but in practice, approximate rankings are often adequate. Accordingly, we study the problem of finding approximate rankings from pairwise comparisons. We analyze an active ranking algorithm that counts the number of comparisons won, and decides whether to stop or which pair of items to compare next, based on confidence intervals computed from the data collected in previous steps. We show that this algorithm succeeds in recovering approximate rankings using a number of comparisons that is close to optimal up to logarithmic factors. We also present numerical results, showing that in practice, approximation can drastically reduce the number of comparisons required to estimate a ranking.

1 Introduction

The problem of ranking a collection of n items from noisy pairwise comparisons arises in a wide range of applications, including recommender systems for rating movies, books, or other consumer items [Pie+13; Agg16]; peer grading for ranking students in massive open online courses [Sha+13]; ranking players in tournaments; search engines; quantifying people's perception of cities from pairwise comparison of street views of the cities [Sal+13]; and online sequential survey sampling for assessing the popularity of proposals in a population of voters [SL15].

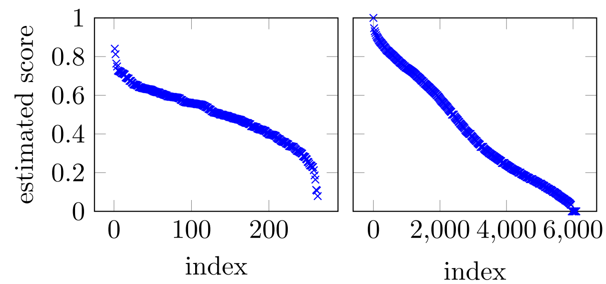
Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

In each of these applications, the aim is to obtain a statistically sound ranking from as few comparisons as possible. In this work, we investigate the power of adaptively selecting which pairs to compare based on the outcomes of previous comparisons, a setting we call *active* or *adaptive* ranking. In contrast, *passive* or *non-adaptive* ranking approaches fix the comparisons to make before any data is collected. It is well understood that one can typically learn a ranking using fewer adaptively chosen comparisons than one would need when passively choosing comparisons [Hec+16]. However, for moderately large or large collections of items–such as the ones that appear in most of the applications mentioned above–or for collections with many items of "similar quality" (to be made rigorous below), learning the *exact* ground-truth ranking may still require prohibitively many comparisons.

Motivated by these large-scale ranking problems, this work studies the problem of adaptivity obtaining *approximate* rankings. We demonstrate that learning an *approximate* ranking may still be statistically tractable even when recovering the *exact* ranking is not. Formally, we consider a collection of n items, and make comparison queries between pairs of items $i, j \in [n] := \{1, 2, \ldots, n\}$. We assume that the response to those queries are stochastic, where the probability that item i "beats" item j is given by $M_{ij} \in (0, 1)$. We assume that the outcomes of all queries are statistically independent, and assume that either item i or item j "wins" the comparison with probability 1, which means that $M_{ij} + M_{ji} = 1$ for all $i \neq j$. Our aim is to rank the items in terms of their Borda scores [DB81], defined as the probability that item i defeats an item chosen uniformly at random from $[n] \setminus \{i\}$:

$$\tau_i := \frac{1}{n-1} \sum_{j \neq i} M_{ij} \ . \tag{1}$$

Apart from their intuitive appeal, the Borda scores generalize the orderings considered in several popular comparison models, including the classical, parametric Bradley-Terry-Luce (BTL) [BT52; Luc59] and Thurstone [Thu27] models, as well as the non-parametric Strong Stochastic Transitivity (SST) model [TER69]. In all of these models, the intrinsic model-defined ordering coincides with that given by the scores $\{\tau_i\}_{i=1}^n$. Rather than learning the scores $\{\tau_i\}_{i=1}^n$ exactly, or ranking items according to their exact score, this paper considers the problem of *approximately* partitioning the items into sets of pre-specified sizes according to their respective scores. This includes finding a total ordering that is approximately correct, and the task of finding a set of k items that is close to the top-k items. For simplicity, we exclusively focus on the latter problem in this paper.



(a) PlaNYC (a) GIF

Figure 1. Estimated scores from two different domains: (a) Comparisons of the proposals in the PlaNYC survey, as reported in the paper [SL15]. (b) Scores from comparisons of Gif's according to whether they display a certain emotion (see http://www.gif.gf/).

Contributions: Our main contribution is to present and analyze a novel active ranking algorithm for estimating an *approximate* ranking of the items. The algorithm is based on adaptively estimating the scores to within sufficient resolution to deduce a ranking. We establish that with high probability, the algorithm returns a ranking which satisfies the desired approximation guarantee, and attains a distribution-dependent sample complexity which can be parameterized in terms of the scores $\{\tau_i\}_{i=1}^n$. We then prove distribution-dependent lower bounds that match our upper bound up to logarithmic factors for many problem instances. Our analysis leverages the fact that ranking in terms of the scores $\{\tau_i\}_{i=1}^n$ is related to a particular class of multi-armed bandit problems [ED+06; Bub+13; Urv+13]; this same connection has been observed in the context of finding the top item [Yue+12; Jam+15; Urv+13]. Since to the best of our knowledge, the *approximate* subset selection problem has not been studied in the bandit literature, a version of our algorithm and results are also new when specialized to the multi-armed bandit problem. Finally, we examine pathological distributions for which the complexity of approximate ranking (or approximate subset selection in the multi-armed bandit setup) seems to diverge from what one would expect. In these cases, we show that careful randomized guessing strategies can yield significant improvements in sample complexity.

Motivation for Approximate Rankings: In order to understand how approximation can drastically reduce the number of comparisons required, let us consider a motivating example. Suppose that we are interested in identifying the top-k items, and suppose for simplicity that the items are ordered, i.e., $\tau_1 > \tau_2 > \ldots > \tau_n$ (of course this ordering is not known a-priori). The paper [Hec+16] shows that in the active setting, the number of comparisons necessary and sufficient for finding the top k items is of the order

$$\sum_{i=1}^{k} \frac{1}{(\tau_i - \tau_{k+1})^2} + \sum_{i=k+1}^{n} \frac{1}{(\tau_k - \tau_i)^2}, \qquad (2)$$

up to a logarithmic factor. Thus, the sample complexity depends on the distribution of the scores; see Figure 1 how these scores are distributed in some applications. In practice, the differences between the scores often obey the scaling $\tau_i - \tau_{i+1} \approx 1/n$ on average (see Figure 1). To identify the top-k items exactly, the aforementioned optimal active scheme would require on the order of n^2 comparisons, and a minimax-optimal passive ranking scheme would even require on the order of n^3 comparisons [SW15].

Theorem 1 in this paper shows that if one does not need to extract the exact top-k items, but is instead willing to tolerate a few–say, h many–mistakes, then the number of comparisons shrinks drastically, specifically by a factor proportional to h. In particular, if we want to find a set $\mathcal{S}$ of 10% of the items ($k = 0.1n$) such that all but 10% of the elements of $\mathcal{S}_1$ are among the *true* top 10% of items ($h = 0.1k$, $k = 0.1n$), then the overall number of comparisons required would be on the order of $n^2/h = 100n$. Thus, relaxing to approximate ranking can yield speedups that are *linear* and *quadratic* in the number of items, compared to optimal exact active and exact passive schemes. Moreover, our algorithm (Algorithm 1 below) that obtains this factor-of-h speedup does not require priori information about the spacings of the $\{\tau_i\}_{i=1}^n$, but instead learns a near-optimal measurement allocation for these scores adaptively.

Related works: There is a vast literature on ranking and estimation from pairwise comparison data; however, most work focuses on finding *exact* rankings. There are a number of papers [Hun04; Neg+12; Haj+14; Sha+16a; SW15] devoted to settings in which pairs to be compared are chosen a priori, whereas here we assume that the pairs may be chosen in an active manner. Moreover, several works impose restrictions on the pairwise comparison probabilities, e.g., by assuming the Bradley-Terry-Luce (BTL) parametric model (discussed below) [Sz15; Hun04; Neg+12; Haj+14; Sha+16a]. Eriksson [Eri13] considers the problem of finding the very top items using graph-based techniques, whereas Busa-Fekete et al. [BF+13] consider the problem of finding the top-k items. Ailon [Ail11] considers the problem of linearly ordering the items so as to disagree in as few pairwise preference labels as possible. Our work is also related to the literature on multi-armed bandits, as discussed later in the paper.

2 Problem Formulation and Background

In this section, we formally state the approximate ranking problem considered in this paper.

2.1 Pairwise probabilities and scores

Given a collection of items $[n] := \{1, \ldots, n\}$, let us denote by $M_{ij} \in (0,1)$ the (unknown) probability that item i wins a comparison with item j. We let X_{ij} denote a Bernoulli random variable taking a value of 1 if i beats j and 0 otherwise, so that $M_{ij} = \mathbb{E}[X_{ij}]$. Moreover, we require that any comparison results in a winner, so that $M_{ij} + M_{ji} = 1$. For each item $i \in [n]$, recall that the score (1) defined by $\tau_i := \frac{1}{n-1}\sum_{j \in [n]\setminus\{i\}} M_{ij}$ corresponds to the probability that item i wins a comparison with an item j chosen uniformly at random from $[n] \setminus \{i\}$. We let $\pi\colon [n] \to [n]$ denote any (possibly non-unique) permutation such that $\tau_{\pi(1)} \geq \tau_{\pi(2)} \geq \ldots \geq \tau_{\pi(n)}$. In words, $\pi(i)$ denotes the item with the i^{th} largest score. Ranking corresponds to partitioning the items into disjoint sets according to its scores. For simplicity, in this paper we focus on the ranking problem of splitting $[n]$ into the top-k items and its complement $\mathcal{S}_1 := \{\pi(1), \ldots, \pi(k)\}$, $\mathcal{S}_2 := \{\pi(k+1), \ldots, \pi(n)\}$. In this work, our goal is to find an approximation to $\mathcal{S}_1$ and $\mathcal{S}_2$ in terms of the Hamming distance between two sets $\mathcal{S}, \mathcal{S}'$, defined as $\mathrm{D}_H(\mathcal{S}, \mathcal{S}') := |(\mathcal{S} \cup \mathcal{S}') \setminus (\mathcal{S} \cap \mathcal{S}')|$. Specifically, we say the ranking $\widehat{\mathcal{S}}_1, \widehat{\mathcal{S}}_2$ with $|\widehat{\mathcal{S}}_\ell| = |\mathcal{S}_\ell|$ is h-Hamming-accurate if

$$\mathrm{D}_H(\widehat{\mathcal{S}}_\ell, \mathcal{S}_\ell) \leq 2h, \quad \text{for } \ell \in \{1, 2\}.$$

For future reference, we define

$$\mathcal{C}_{M_{\min}} := \left\{ M \in (0,1)^{n\times n} \,|\, M_{ij} = 1 - M_{ji}, M_{ij} \geq M_{\min} \right\},$$

corresponding to the set of pairwise comparison matrices with pairwise comparison probabilities lower bounded by $M_{\min}$.

2.2 The active approximate ranking problem

An active ranking algorithm acts on a pairwise comparison model $M \in \mathcal{C}_0$. The goal is to obtain an approximate partition of the items into disjoint sets from active comparisons. At each time instant, the algorithm can compare two arbitrary items, which the algorithm may select based on the outcomes of previous comparisons. When comparing i and j, the algorithm obtains an independent draw of the random variable M_{ij} in response. The algorithm terminates based on an associated stopping rule, and returns an approximate ranking $\widehat{\mathcal{S}}_1, \widehat{\mathcal{S}}_2$. For a given tolerance parameter $\delta \in (0,1)$, we say a ranking algorithm $\mathcal{A}$ is (h, δ)-accurate for a pairwise comparison matrix M, if the ranking returned is h-Hamming accurate with probability at least $1 - \delta$. Moreover, we say that $\mathcal{A}$ is *uniformly* (h, δ)*-accurate* over a given set of pairwise comparison models $\mathcal{C}$ if it is δ-accurate for each $M \in \mathcal{C}$.

2.3 Relation to multi-armed bandits

The *exact* version of the ranking problem considered in this paper is related to the subset selection problem in the bandit literature [Kal+12]. Specifically, a multi-armed bandit model consists of n arms, each a random variable with unknown distribution. The subset selection problem is concerned with identifying the top arms (according to the means) by taking independent draws of the random variables. Various works [YJ11; Yue+12; Urv+13; Jam+15] have observed that, by definition of the score τ_i, comparing item i to an item chosen uniformly at random from $[n] \setminus \{i\}$ can be modeled as drawing a Bernoulli random variable with mean τ_i. Our subsequent analysis relies on this relation.

However, when viewing our problem as a multi-armed bandit problem with means $\{\tau_i\}_{i=1}^n$, we are ignoring the fact that the means are coupled, as they must be realized by some pairwise comparison matrix M. Due to $M_{ij} = 1 - M_{ji}$, this matrix must satisfy certain constraints, such as $\sum_{i=1}^n \tau_i = n/2$ and $\sum_{i=1}^j \tau_{\pi(i)} \geq \frac{1}{n-1}\frac{j(j-1)}{2}$ (e.g., see the papers [Lan53; Joe88]). Our algorithm turns out to be near-optimal, even though it does not take those constraints into account. This seems to corroborate the observation in [Sim+17] that many types of constraints surprisingly do not improve the sample complexity of bandit problems.

Finally, at least to the best of our knowledge, the problem of *approximate* subset selection has not been studied in the bandit literature, meaning that our algorithm and results are also new when specialized to the multi-armed bandit problem. However, it should be noted that other versions of approximation have been considered in the literature; for instance, Zhou et al. [Zho+14] studied the problem of selecting k arms with low aggregate regret, defined as the gap between the average reward of the optimal solution and the solution given by the algorithm.

2.4 Parametric models

In this section, we introduce a family of parametric models that are popular in the pairwise comparison literature [Sz15; Hun04; Neg+12; Haj+14; Sha+16a]. We focus on these parametric models in Section 3.3, where we show that, perhaps surprisingly, if the pairwise comparison probabilities are bounded away from zero, for most constellations of scores, these assumptions can at most provide little gains in sample complexity.

Any member of this family is defined by a strictly increasing and continuous function $\Phi\colon \mathbb{R} \to [0,1]$ obeying $\Phi(t) = 1 - \Phi(-t)$, for all $t \in \mathbb{R}$. The function Φ is assumed to be known. A pairwise comparison matrix in this family is associated to an unknown vector $w \in \mathbb{R}^n$, where each entry of w represents some quality or strength of the corresponding item. The parametric model $\mathcal{C}_{\mathrm{PAR}(\Phi)}$ associated with the function Φ is defined as:

$$\mathcal{C}_{\mathrm{PAR}(\Phi)} = \{M_{ij} = \Phi(w_i - w_j) \ \ \forall i, j \in [n], w \in \mathbb{R}^n\}.$$

Popular examples of models in this family are the Bradley-Terry-Luce (BTL) model, obtained by setting Φ equal to the sigmoid function $\Phi(t) = \frac{1}{1+e^{-t}}$, and the Thurstone model, obtained by setting Φ equal to the Gaussian CDF. Since $\tau_1 > \tau_2 > \ldots > \tau_n$ is equivalent to $w_1 > w_2 > \ldots > w_n$, the ranking induced by the scores $\{\tau_i\}_{i=1}^n$ is equivalent to that induced by w.

3 Hamming-LUCB: Algorithm and analysis

In this section, we present our approximate ranking algorithm, and an analysis proving that it is near optimal for many interesting and natural problem instances.

3.1 The Hamming-LUCB algorithm

Our algorithm is based on actively identifying sets $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ consisting of $k-h$ items and $n-k-h$ items, respectively, such that with high confidence the items in the first set have a larger score than the items in the second set. Once we have found such sets, we can arbitrarily distribute the remaining items to the sets $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ in order to obtain a Hamming-accurate ranking with high confidence.

Our algorithm identifies those sets based on adaptively estimating the scores $\{\tau_i\}_{i=1}^n$. We estimate the score of item i by comparing item i with items chosen uniformly at random from $[n] \setminus \{i\}$, which yields an unbiased estimate of τ_i. The key idea is to only estimate the scores sufficiently well so we can obtain the two sets $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ from them. This strategy decides based on the current estimates of the scores and associated confidence intervals which estimate to "update", by comparing it to a randomly chosen item. Our strategy to update the estimates of the scores is guided by the insight that the "easiest" items to distinguish are the top $k-h$ items, $\{\pi(1), \ldots, \pi(k-h)\}$, and the bottom $n-k-h$ items, $\{\pi(k+h+1), \ldots, \pi(n)\}$. Hence, our algorithm focuses on what it "thinks" are those top and bottom items.

We define a confidence bound based on an non-asymptotic version of the law of the iterated algorithm [Kau+16; Jam+14]; it is of the form $\alpha(u) \propto \sqrt{\frac{\log(\log(u)n/\delta)}{u}}$, where u is an integer corresponding to the number of comparisons, and with the constants involved explicitly chosen by setting $\alpha(u) = \sqrt{\frac{\beta(u,\delta/n)}{2u}}$, with $\beta(u,\delta') = \log(1/\delta') + 0.75\log\log(1/\delta') + 1.5\log(1+\log(u/2))$. For each item $i \in [n]$, the algorithm stores a counter T_i of the number of comparisons in which it has been involved, along with an empirical estimate of the associated score $\widehat{\tau}_i(T_i)$. For notational convenience, we adopt the shorthands $\widehat{\tau}_i = \widehat{\tau}_i(T_i)$ and $\alpha_i = \alpha(T_i)$. Within each round, we also let $(\cdot)$ denote a permutation of $[n]$ such that $\widehat{\tau}_{(1)} \geq \widehat{\tau}_{(2)} \geq \cdots \geq \widehat{\tau}_{(n)}$. We then define the indices

$$d_1 = \operatorname*{arg\,min}_{i \in \{(1),\ldots,(k-h)\}} \widehat{\tau}_i - \alpha_i,$$
$$d_2 = \operatorname*{arg\,max}_{i \in \{(k+1+h),\ldots,(n)\}} \widehat{\tau}_i + \alpha_i. \tag{3}$$

These indices are the analogues of the standard indices of the Lower-Upper Confidence Bound (LUCB) strategy from the bandit literature [Kal+12] for the top $k-h$ and bottom $n-h-k$ items. The LUCB strategy for exact top k recovery would update the scores d_1 and d_2 (for $h=0$) at each round. As mentioned before, our strategy will go after what it "thinks" are the top $k-h$ items, $\tilde{\mathcal{S}}_1 = \{(1), \ldots, (k-h)\}$, and what it "thinks" are the bottom $n-k-h$ items, $\tilde{\mathcal{S}}_2 = \{(k+1+h), \ldots, (n)\}$. Moreover, the algorithm keeps all the other items in consideration for inclusion in these sets, by keeping their confidence intervals below the confidence intervals of the items in $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ (cf. equation (4) in the algorithm below). This is crucial to ensure that the algorithm does not get stuck trying to distinguish the middle items $\{\pi(k-h+1), \ldots, \pi(k+h)\}$, which in general requires many comparisons, as their scores are typically closer. In Figure 2 we show an example run of the Hamming-LUCB algorithm, to illustrate the idea.

Algorithm 1: Hamming-LUCB

Input: Confidence parameter δ.

$\alpha(\cdot)$

Initialization: For every item $i \in [n]$, compare i to an item j chosen uniformly at random from $[n] \setminus \{i\}$, and set $\widehat{\tau}_i(1) = \mathbb{1}\{i \text{ wins}\}$, $T_i = 1$.

Do until termination:

Let $(\cdot)$ denote a permutation of $[n]$ such that $\widehat{\tau}_{(1)} \geq \widehat{\tau}_{(2)} \geq \ldots \widehat{\tau}_{(n)}$.

For d_1 and d_2 defined by equation (3), define the indices

$$b_1 = \arg\max_{i \in \{d_1, (k-h+1), \ldots, (k)\}} \alpha_i, \qquad b_2 = \arg\max_{i \in \{d_2, (k+1), \ldots, (k+h)\}} \alpha_i. \tag{4}$$

For $i \in \{b_1, b_2\}$, increment $T_i \leftarrow T_i + 1$, compare i to an item j chosen uniformity at random from $[n] \setminus \{i\}$, and update $\widehat{\tau}_i \leftarrow \frac{T_i - 1}{T_i} \widehat{\tau}_i + \frac{1}{T_i} \mathbb{1}\{i \text{ wins}\}$.

End Loop once the termination condition holds:

$$\widehat{\tau}_{d_1} - \alpha_{d_1} \geq \widehat{\tau}_{d_2} + \alpha_{d_2}. \tag{5}$$

Return the estimates of the partitions $\widehat{\mathcal{S}}_1 = \{(1), \ldots, (k)\}$ and $\widehat{\mathcal{S}}_2 = \{(k+1), \ldots, (n)\}$.

3.2 Guarantees and optimality of the Hamming-LUCB algorithm

We next establish guarantees on the number of comparisons for the Hamming-LUCB algorithm to succeed. As we show below, the number of comparisons depends on the following gaps between the scores

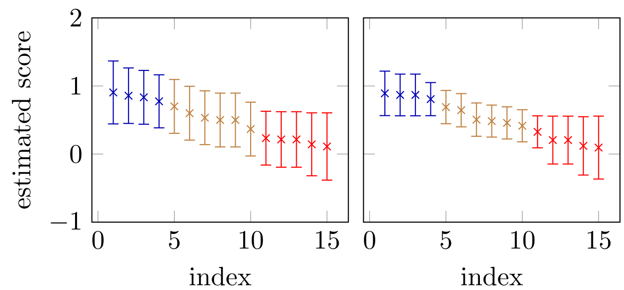
$$\Delta_{i,k+1+h} \quad \text{and} \quad \Delta_{k-h,i}, \quad \text{where} \quad \Delta_{i,j} := \tau_i - \tau_j.$$

Thus, as one might intuitively expect, the number of comparisons is typically smaller when h is larger, as the corresponding gaps typically become larger.

Theorem 1. *For any $M \in \mathcal{C}_0$, the Hamming-LUCB algorithm run with confidence parameter δ is (h, δ)-Hamming-accurate, and with probability at least $1-\delta$, makes at most $N_h^{\text{up}}(M)$ comparisons, where*

$$N_h^{\text{up}}(M) = \widetilde{O}\left(\sum_{i=1}^{k-h} \Delta_{i,k+1+h}^{-2} + \sum_{i=k+1+h}^{n} \Delta_{k-h,i}^{-2} + 2h \Delta_{k-h,k+1+h}^{-2} \right). \tag{6}$$

The notation $\widetilde{O}$ absorbs factors logarithmic in n, and doubly logarithmic in the gaps.



(a) after 200 comparisions (b) at termination

Figure 2. Visualization of a run of the Hamming-LUCB algorithm on a problem instance with scores evenly spaced in the interval $[0.1, 0.9]$, and parameters $k = 7, h = 3$. The estimates of the scores $\widehat{\tau}_i$ of the top items $\{(1), \ldots, (k-h)\}$, the middle items $\{\pi(k-h+1), \ldots, \pi(k+h)\}$, and the bottom items $\{(k+1+h), \ldots, (n)\}$, along with the confidence intervals $[\widehat{\tau}_i - \alpha_i, \widehat{\tau}_i + \alpha_i]$ are depicted in blue, brown, and red, respectively, after 200 comparisons, and at termination. Note that once the confidence intervals of the top and bottom items are separated, the algorithm terminates.

Theorem 1 proves that the Hamming-LUCB algorithm is (h, δ)-accurate, and characterizes the number of comparisons that it requires as a function of the gaps between the scores.

Comparing $N_h^{\text{up}}(M)$ to the number of comparisons necessary and sufficient for finding the top-k items, we see that the Hamming-LUCB algorithm depends on the gaps $\Delta_{i,k+1+h}$ and $\Delta_{k-h,i}$ instead of the gaps $\Delta_{i,k+1}$ and $\Delta_{k,i}$ which appear in the sample complexity for finding the top k items (cf. equation (2)). These gaps are typically significantly larger, resulting in a lower sample complexity. For example, in practice, the scores are often increasing in that $\tau_i - \tau_{i+1}$ is on average on the order of $1/n$. Thus, for sufficiently large h, several real world models belong to the class

$$\mathcal{C}_{\beta,h} := \{M \in (0,1)^{n \times n} \mid M_{ij} = 1 - M_{ji}, \tag{7}$$
$$\text{and } \tau_i - \tau_{i+h} \geq \beta h/n, \text{ for all } i\}. \tag{8}$$

See Figure 1 for plausible members of this class. For this class, the complexity of finding the top-k items with the Hamming-LUCB algorithm is on the order of $\widetilde{O}\left(n^2/(\beta^2 h)\right)$, which is by a factor of h smaller than the complexity for finding the exact top-k items.

Moreover, Hamming LUCB provides a strict improvement over the optimal sample complexity in the passive setup, for which Shah and Wainwright [SW15] establish upper bounds and minimax lower bounds which state that $O(n \log n/\Delta^2_{k-h,k+1+h})$ comparisons are necessary and sufficient to identify the top k items up to a Hamming error h with high probability.

As h increases, the upper bound depends on gaps between items with increasingly disparate position in the ranking, and thus, the upper bound on the sample complexity decreases. The following lower bound shows that, up to logarithmic factors in n, doubly logarithmic factors in the gaps, and a multiplicative scaling of h, the Hamming-LUCB algorithm is optimal.

Theorem 2. *For any $\delta \in (0, 0.14]$, let $\mathcal{A}$ denote an algorithm which is uniformly (h, δ)-accurate over $\mathcal{C}_{1/8}$. Then when $\mathcal{A}$ is run on any comparison instance $M \in \mathcal{C}_{3/8}$, the algorithm $\mathcal{A}$ must make at least $N_h^{\text{low}}(M)$ comparisons in expectation, where*

$$N_h^{\text{low}}(M) := c_{\text{low}} \log\left(\frac{1}{2\delta}\right)\left(\sum_{i=1}^{k-2h} \Delta^{-2}_{i,k+1+2h} + \sum_{i=k+1+2h}^{n} \Delta^{-2}_{k-2h,i}\right),$$

for some universal constant $c_{\text{low}} > 0$.

Note that the above lower bound *does not* depend on the gaps involving the items $k - 2h + 1, \ldots, k + 2h$. However, we can still relate the lower bound to the upper bound by (see the supplement for the simple proof)

$$N_{3h}^{\text{up}}(M) \leq \widetilde{O}(N_h^{\text{low}}(M)), \tag{9}$$

so that we see that, up to rescaling our Hamming error tolerance h, our upper and lower bounds ($N_h^{\text{up}}(M)$ and $N_h^{\text{low}}(M)$ respectively) match up to logarithmic factors. For many problem instances of interest—such as models in the class $\mathcal{C}_{\beta,h}$ in equation (8)—the sample complexity bounds $N_{3h}^{\text{up}}(M)$ and $N_h^{\text{low}}(M)$ degrade gracefully with the Hamming tolerance h, so that typically we have $N_h^{\text{up}}(M) \leq \widetilde{O}(N_h^{\text{low}}(M))$.

Observe that if $h = 0$, we recover the exact top-k recovery upper bound in equation (2), which is related to similar results for multi armed bandits [Kal+12]. We believe that by modifying the confidence intervals in Hamming LUCB as in the LUCB++ algorithm of Simchowitz et al. [Sim+17], one can sharpen the upper bound $N_h^{\text{up}}(M)$ on the sample complexity by replacing $\log n$ with $\log k$ on the terms $\Delta^{-2}_{k-h,i}$ corresponding to items $i \in \{k+h+1, \ldots, n\}$, thereby matching known lower bounds for top-k subset selection problem in the bandit literature [Sim+17; Che+17; Kal+12]. In the interest of simplicity, we defer refining these logarithmic factors to later work.

3.3 Parametric models

Even though the lower bound of $N_h^{\text{low}}(M)$ qualitatively matches the upper bound $N_h^{\text{up}}(M)$, it gives the misleading impression that an h-approximate algorithm can get away without querying the items in $\{k - 2h, \ldots, k + 2h + 1\}$. In the proof section, we use techniques from [Sim+17] and [Che+17] to establish a more refined technical lower bound showing that all items, including those with ranks not in but close to k, must be compared an "adequate" number of times. For simplicity, we state a consequence of this lower bound applied to the parametric models described in Section 2.4. In addition to showing that each item has to be compared a certain number of times, this bound also establishes that *even knowledge of the exact parametric form of the pairwise comparison probabilities* M cannot drastically improve the performance of an active ranking algorithm.

In more detail, we say that a model is parametric if there exists a strictly increasing CDF $\Phi\colon \mathbb{R} \to [0, 1]$ such that $M_{ij} = \Phi(w_i - w_j)$ for some weights $\{w_j\}$. For any pair of constants $0 < \phi_{\min} \leq \phi_{\max} < \infty$, we say that a CDF Φ is $(\phi_{\min}, \phi_{\max}, M_{\min})$-bounded, if it is differentiable, and if its derivative Φ' satisfies the bounds

$$\phi_{\min} \leq \Phi'(t) \leq \phi_{\max}, \tag{10}$$

for all $t \in [\Phi^{-1}(M_{\min}), \Phi^{-1}(1 - M_{\min})]$. Note that for the popular BTL and Thurstone models, equation (10) holds with $\phi_{\min}/\phi_{\max}$ close to one, provided that $M_{\min}$ is not too small. We say that an algorirthm $\mathcal{A}$ is *symmetric* if its distribution of comparisons commutes with permutations of the items. For any such algorithm, our main lower bound is as follows:

Theorem 3. *For a given $\delta \leq \frac{1}{2}\min(\frac{1}{k}, \frac{1}{n-k})$, let $\mathcal{A}$ be any symmetric algorithm that is uniformly (h, δ)-Hamming accurate over $\mathcal{C}_{\text{PAR}(\Phi)} \cap \mathcal{C}_{M_{\min}}$. Given any instance $M \in \mathcal{C}_{\text{PAR}(\Phi)} \cap \mathcal{C}_{M_{\min}}$, when $\mathcal{A}$ is run on the instance M, then for any integer $q \geq 1$ and any item $a \in [n]$, it must make at least*

$$\frac{M_{\min}\phi^2_{\min}}{3\phi^2_{\max}}\left(\frac{2q-1}{2h+q}\right)^2 \cdot \max_{b \in \{k-2(h+q), k+1+2(h+q)\}} \Delta^{-2}_{a,b}$$

comparisons involving item a on average.

In particular, by choosing $q = h$, we see that the total sample complexity is lower bounded by

$$\sum\nolimits_{i=1}^{k-3h} \Delta^{-2}_{i,k+1+3h} + \sum\nolimits_{i=k+1+3h}^{n} \Delta^{-2}_{k-3h,i} + 6h\Delta^{-2}_{k-3h,k+1+3h}, \tag{11}$$

which is equivalent to the upper bound $N_{3h}^{\text{up}}(M)$ achieved by the Hamming-LUCB algorithm up to logarithmic factors. The lower bound from Theorem 3 is stronger than the lower bound from Theorem 2, in that it applies to the larger class of algorithms that are only (h, δ)-accurate over the smaller class of parametric models. In fact, the parametric subclass $\mathcal{C}_{\mathrm{PAR}(\Phi)} \cap \mathcal{C}_{M_{\min}}$ is significantly smaller than the full set of pairwise comparison models $\mathcal{C}_{M_{\min}}$, in the sense that one can find matrices in $\mathcal{C}_{M_{\min}}$ that cannot be well-approximated by any parametric model [Sha+16b]. Therefore, Theorem 3 shows that, up to rescaling the Hamming error tolerance h and logarithmic factors, the Hamming-LUCB algorithm is optimal, even if we restrict ourself to algorithms that are uniformly (h, δ)-accurate *only* over a parametric subclass. Thus, in the regime where the pairwise comparison probabilities are bounded away from zero, parametric assumptions cannot substantially reduce the sample complexity of finding an *approximate* ranking; an observation that has been made previously in the paper [Hec+16] for exact rankings.

A second and equally important consequence of Theorem 3 is that each item has to be sampled a certain number of times, an intuition not captured by Theorem 2. This conclusion continues to hold for general pairwise comparison matrices, please see the supplement for a formal statement.

3.4 Random guessing

Even though our the upper and lower bounds essentially match whenever $N_{3h}^{\mathrm{up}}(M) \approx N_h^{\mathrm{up}}(M)$, there are there are pathological instances for which $N_{3h}^{\mathrm{up}}(M) \ll N_h^{\mathrm{up}}(M)$, and where the Hamming-LUCB algorithm will make considerably more comparisons than a careful random guessing strategy.

As an example, consider a problem instance parameterized by κ, ϵ, with scores given by

$$\tau_i = \begin{cases} 1/2 + \kappa, & i \in \{1, \ldots, k-h-2\} \\ 1/2 + 2\epsilon, & i = k-h-1 \\ 1/2 + \epsilon, & i \in \{k-h, \ldots, k\} \\ 1/2 - \epsilon, & i \in \{k+1, \ldots, k+1+h\} \\ 1/2 - 2\epsilon, & i = k+2+h \\ 1/2 - \kappa, & i \in \{k+h+2, \ldots, n\} \end{cases},$$

for some κ and ϵ. The upper bound (6) for the Hamming-LUCB strategy is at least on the order of h/ϵ^2, since the gap between the $(k-h)$-th and the $(k+1+h)$-th largest score is 4ϵ. However, the lower bound provided by Theorem 2 is $(n - 2(h+2))/\kappa^2$, which is independent of ϵ. Thus, by making ϵ small, the ratio of upper and lower bounds becomes arbitrarily large. Intuitively, Hamming-LUCB is wasteful because it is attempting to identify the *exact* top $k-h$ arms with too much precision. However, for this particular problem instance, the following random guessing strategy will attain our lower bound. First, we obtain estimates $\widehat{\tau}_i$ of each score τ_i by comparing item i to $T_i = c\log(n/\delta)/\kappa^2$ randomly chosen items. For each score, test whether there are $k-h-2$ items obeying $\widehat{\tau}_i \geq 1/2 + c\sqrt{\log(n)/T_i}$ and whether there are $n-(k+h+2)$ items obeying $\widehat{\tau}_i \leq 1/2 - c\sqrt{\log(n)/T_i}$. If yes, assign these items the estimates $\widehat{\mathcal{S}}_1$ and $\widehat{\mathcal{S}}_2$, respectively, and assign all remaining items uniformly at random to the sets $\widehat{\mathcal{S}}_1$ and $\widehat{\mathcal{S}}_2$, and terminate.

4 Experimental results

In this section, we provide experimental evidence that corroborates our theoretical claims that the Hamming-LUCB algorithm allows to significantly reduce the number of comparisons if one is content with an approximate ranking. We show that these gains are attained on a real-world data set. Specifically, we generate a pairwise comparison model by choosing M such that the Borda scores τ_i coincide with those found empirically in the PlaNYC survey [SL15]; see panel (b) of Figure 1 for the form of these scores. We emphasize that since Hamming LUCB depends only on the Borda scores τ_i and not on the comparison probabilities M_{ij}, these simulations provide a faithful representation of how Hamming LUCB performs on real-world data.

In Figure 3, we plot the results of running the Hamming-LUCB algorithm on the PlanNYC-pairwise comparison model in order to determine the top $k = 35$ items, for different values of h. We observed that results for other values of k were very similar. As suggested by our theory, the number of comparisons to find an approximate ranking decays in a manner inversely proportional to h.

We compare the Hamming-LUCB algorithm to another sensible active ranking strategy for obtaining an Hamming-accurate ranking. Specifically, we consider a version of the successive elimination strategy proposed in [Hec+16, Sec. 3.1] for finding an exact ranking. This strategy can be adapted to yield an Hamming-accurate ranking by changing its stopping criterium. Instead of stopping once all items have been eliminated, we stop when either $k-h$ items have been assigned to the top, or $n-k-h$ items have been assigned to the bottom. While this strategy yields an Hamming accurate ranking, its sample complexity is, up to logarithmic factors, equal to $\sum_{i=1}^{k-h} \frac{1}{(\tau_i - \tau_{k+1})^2} + \sum_{i=k+1+h}^{n} \frac{1}{(\tau_k - \tau_i)^2} + 2h(\tau_{k-h} - \tau_{k+1+h})$, which is strictly smaller than that of the Hamming-LUCB algorithm. As Figure 3 shows, this strategy requires significantly more comparisons for finding an approximate ranking, thereby validating the benefits of our approach.

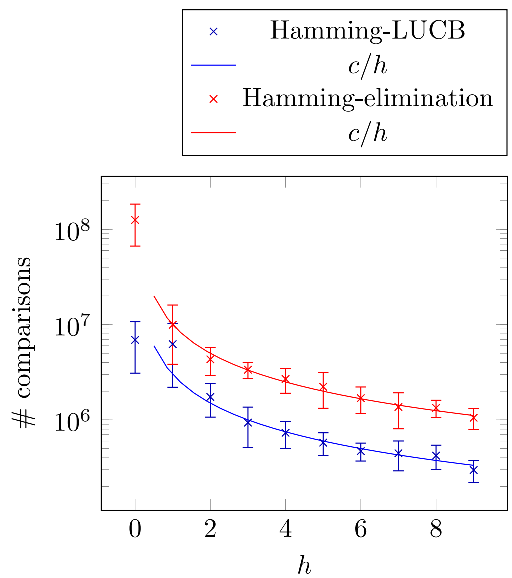


Figure 3. Sample complexity of the Hamming-LUCB algorithm and an elimination strategy run on a pairwise comparison model resembling the PlaNYC online sequential survey. Both algorithms find the top 35 proposals out of 263 proposals, up to Hamming error h. The error bars correspond to one standard deviation from the mean. The results show that the sample complexity of the Hamming-LUCB algorithm for finding an h-accurate ranking drops by a factor of about h. Moreover, the Hamming-LUCB algorithm requires significantly fewer samples than the elimination strategy.

5 Discussion

In this paper, we considered the problem of finding an Hamming-approximate ranking from pairwise comparisons. We provided an algorithm that allows to significantly reduce the sample complexity if one is content with an approximate ranking. Moreover, we showed that our algorithm is near optimal and remains near optimal when imposing common parametric assumptions. There are a number of open and practically relevant questions suggested by our work. As our work shows, it is non-trivial to adapt to approximate notions of ranking. It would be interesting to further understand how one can optimally adapt to approximate notions of ranking, by closing the gap of our bounds for pathological problem instances, and more importantly by studying other notions of approximate rankings. It would also be interesting to study algorithms that work with a limited budget of queries and quantify their approximation accuracy.

Acknowledgements

The work of RH was supported by the Swiss National Science Foundation under grant P2EZP2_159065. The work of MJW was partially supported by grants DOD ONR-N00014 and NSF-DMS-1612948.

References

[Agg16] C. C. Aggarwal. *Recommender systems: The textbook*. Springer, 2016.

[Ail11] N. Ailon. “Active learning ranking from pairwise preferences with almost optimal query complexity”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 810–818.

[BT52] R. A. Bradley and M. E. Terry. “Rank analysis of incomplete block designs: I. The method of paired comparisons”. In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[Bub+13] S. Bubeck, T. Wang, and N. Viswanathan. “Multiple identifications in multi-armed bandits”. In: *International Conference on Machine Learning*. 2013, pp. 258–265.

[BF+13] R. Busa-Fekete, B. Szorenyi, W. Cheng, P. Weng, and E. Hüllermeier. “Top-k selection based on adaptive sampling of noisy preferences”. In: *International Conference on Machine Learning*. 2013, pp. 1094–1102.

[Che+17] L. Chen, J. Li, and M. Qiao. "Nearly instance optimal sample complexity bounds for top-k arm selection". In: *arXiv preprint arXiv:1702.03605* (2017).

[DB81] J. C. De Borda. "Mémoire sur les élections au scrutin". In: *Histoire de l'Académie Royale des Sciences* (1781).

[Eri13] B. Eriksson. "Learning to top-k search using pairwise comparisons". In: *International Conference on Machine Learning*. 2013, pp. 265–273.

[ED+06] E. Even-Dar, S. Mannor, and Y. Mansour. "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems". In: *Journal on Machine Learning Research* 7 (2006), pp. 1079–1105.

[Haj+14] B. Hajek, S. Oh, and J. Xu. "Minimax-optimal inference from partial rankings". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1475–1483.

[Hec+16] R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright. "Active ranking from pairwise comparisons and when parametric assumptions don't help". In: *arXiv:1606.08842* (2016).

[Hun04] D. Hunter. "MM algorithms for generalized Bradley-Terry models". In: *Annals of Statistics* (2004), pp. 384–406.

[Jam+14] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. "lil' UCB: An optimal exploration algorithm for multi-armed bandits". In: *Conference on Learning Theory*. 2014, pp. 423–439.

[Jam+15] K. Jamieson, S. Katariya, A. Deshpande, and R. Nowak. "Sparse dueling bandits". In: *International Conference on Artificial Intelligence and Statistics*. 2015, pp. 416–424.

[Joe88] H. Joe. "Majorization, entropy and paired comparisons". In: *The Annals of Statistics* 16.2 (1988), pp. 915–925.

[Kal+12] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. "PAC subset selection in stochastic multi-armed Bandits". In: *International Conference on Machine Learning*. Vol. 655–662. 2012.

[Kau+16] E. Kaufmann, O. Cappé, and A. Garivier. "On the complexity of best arm identification in multi-armed bandit models". In: *Journal on Machine Learning Research* 17.1 (2016), pp. 1–42.

[Lan53] H. G. Landau. "On dominance relations and the structure of animal societies: III The condition for a score structure". In: *The Bulletin of Mathematical Biophysics* 15.2 (1953), pp. 143–148.

[Luc59] R. D. Luce. *Individual choice behavior: A theoretical analysis*. Wiley, 1959.

[Neg+12] S. N. Negahban, S. Oh, and D. Shah. "Iterative ranking from pair-wise comparisons". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2474–2482.

[Pie+13] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. "Tuned models of peer assessment in MOOCs". In: *International Conference on Educational Data Mining*. 2013.

[Sal+13] P. Salesses, K. Schechtner, and C. A. Hidalgo. "The collaborative image of the city: Mapping the inequality of urban perception". In: *PLOS ONE* 8.7 (2013), e68400.

[SL15] M. J. Salganik and K. E. C. Levy. "Wiki surveys: Open and quantifiable social data collection". In: *PLOS ONE* 10.5 (2015), e0123483.

[SW15] N. B. Shah and M. J. Wainwright. "Simple, robust and optimal ranking from pairwise comparisons". In: *arXiv:1512.08949* (2015).

[Sha+13] N. B. Shah, J. K. Bradley, A. Parekh, M. J. Wainwright, and K. Ramchandran. "A case for ordinal peer-evaluation in MOOCs". In: *NIPS Workshop on Data Driven Education*. 2013.

[Sha+16a] N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. "Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence". In: *Journal on Machine Learning Research* (2016).

[Sha+16b] N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. "Stochastically transitive models for pairwise comparisons: Statistical and computational issues". In: *International Conference on Machine Learning*. 2016.

[Sim+17] M. Simchowitz, K. Jamieson, and B. Recht. "The simulator: Understanding adaptive sampling in the moderate-confidence regime". In: *arXiv:1702.05186* (2017).

[Sz15] B. Szörényi, R. Busa-Fekete, A. Paul, and E. Hüllermeier. "Online rank elicitation for Plackett-Luce: A dueling bandits approach". In: *Advances in Neural Information Processing Systems*. 2015, pp. 604–612.

[Thu27] L. Thurstone. "A law of comparative judgment". In: *Psychological Review* 34.4 (1927), pp. 273–286.

[TER69] A. Tversky and J. Edward Russo. "Substitutability and similarity in binary choices". In: *Journal of Mathematical Psychology* 6.1 (1969), pp. 1–12.

[Urv+13] T. Urvoy, F. Clerot, R. Féraud, and S. Naamane. "Generic exploration and K-armed voting bandits". In: *International Conference on Machine Learning*. 2013, pp. 91–99.

[YJ11] Y. Yue and T. Joachims. "Beat the mean bandit". In: *International Conference on Machine Learning*. 2011, pp. 241–248.

[Yue+12] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. "The K-armed dueling bandits problem". In: *Journal of Computer and System Sciences* 78.5 (2012), pp. 1538–1556.

[Zho+14] Y. Zhou, X. Chen, and J. Li. "Optimal PAC multiple arm identification with applications to crowdsourcing". In: *International Conference on Machine Learning*. 2014, pp. 217–225.