Active Ranking using Pairwise Comparisons

Kevin G. Jamieson
University of Wisconsin
Madison, WI 53706, USA
kgjamieson@wisc.edu

Robert D. Nowak
University of Wisconsin
Madison, WI 53706, USA
nowak@engr.wisc.edu

Abstract

This paper examines the problem of ranking a collection of objects using pairwise comparisons (rankings of two objects). In general, the ranking of n objects can be identified by standard sorting methods using $n \log_2 n$ pairwise comparisons. We are interested in natural situations in which relationships among the objects may allow for ranking using far fewer pairwise comparisons. Specifically, we assume that the objects can be embedded into a d-dimensional Euclidean space and that the rankings reflect their relative distances from a common reference point in $\mathbb{R}^d$. We show that under this assumption the number of possible rankings grows like n^{2d} and demonstrate an algorithm that can identify a randomly selected ranking using just slightly more than $d \log n$ adaptively selected pairwise comparisons, on average. If instead the comparisons are chosen at random, then almost all pairwise comparisons must be made in order to identify any ranking. In addition, we propose a robust, error-tolerant algorithm that only requires that the pairwise comparisons are probably correct. Experimental studies with synthetic and real datasets support the conclusions of our theoretical analysis.

1 Introduction

This paper addresses the problem of ranking a set of objects based on a limited number of pairwise comparisons (rankings between pairs of the objects). A *ranking* over a set of n objects $\Theta = (\theta_1, \theta_2, \dots, \theta_n)$ is a mapping $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ that prescribes an order

$$\sigma(\Theta) := \theta_{\sigma(1)} \prec \theta_{\sigma(2)} \prec \cdots \prec \theta_{\sigma(n-1)} \prec \theta_{\sigma(n)} \tag{1}$$

where $\theta_i \prec \theta_j$ means θ_i precedes θ_j in the ranking. A ranking uniquely determines the collection of pairwise comparisons between all pairs of objects. The primary objective here is to bound the number of pairwise comparisons needed to correctly determine the ranking when the objects (and hence rankings) satisfy certain known structural constraints. Specifically, we suppose that the objects may be embedded into a low-dimensional Euclidean space such that the ranking is consistent with distances in the space. We wish to exploit such structure in order to discover the ranking using a very small number of pairwise comparisons. To the best of our knowledge, this is a previously open and unsolved problem.

There are practical and theoretical motivations for restricting our attention to pairwise rankings that are discussed in Section 2. We begin by assuming that every pairwise comparison is consistent with an unknown ranking. Each pairwise comparison can be viewed as a query: is θ_i before θ_j? Each query provides 1 bit of information about the underlying ranking. Since the number of rankings is $n!$, in general, specifying a ranking requires $\Theta(n \log n)$ bits of information. This implies that at least this many pairwise comparisons are required without additional assumptions about the ranking. In fact, this lower bound can be achieved with a standard adaptive sorting algorithm like binary sort [1]. In large-scale problems or when humans are queried for pairwise comparisons, obtaining this many pairwise comparisons may be impractical and therefore we consider situations in which the space of rankings is structured and thereby less complex.

1

A natural way to induce a structure on the space of rankings is to suppose that the objects can be embedded into a d-dimensional Euclidean space so that the distances between objects are consistent with the ranking. This may be a reasonable assumption in many applications, and for instance the audio dataset used in our experiments is believed to have a 2 or 3 dimensional embedding [2]. We further discuss motivations for this assumption in Section 2. It is not difficult to show (see Section 3) that the number of full rankings that could arise from n objects embedded in $\mathbb{R}^d$ grows like n^{2d}, and so specifying a ranking from this class requires only $O(d \log n)$ bits. The main results of the paper show that under this assumption a randomly selected ranking can be determined using $O(d \log n)$ pairwise comparisons selected in an adaptive and sequential fashion, but almost all $\binom{n}{2}$ pairwise rankings are needed if they are picked randomly rather than selectively. In other words, *actively selecting the most informative queries has a tremendous impact on the complexity of learning the correct ranking.*

1.1   Problem statement

Let σ denote the ranking to be learned. The objective is to learn the ranking by querying the reference for pairwise comparisons of the form

$$q_{i,j} := \{\theta_i \prec \theta_j\}. \tag{2}$$

The response or label of $q_{i,j}$ is binary and denoted as $y_{i,j} := \mathbf{1}\{q_{i,j}\}$ where $\mathbf{1}$ is the indicator function; ties are not allowed. The main results quantify the minimum number of queries or labels required to determine the reference's ranking, and they are based on two key assumptions.

A1 Embedding: The set of n objects are embedded in $\mathbb{R}^d$ (in general position) and we will also use $\theta_1, \ldots, \theta_n$ to refer to their (known) locations in $\mathbb{R}^d$. Every ranking σ can be specified by a *reference point* $r_\sigma \in \mathbb{R}^d$, as follows. The Euclidean distances between the reference and objects are consistent with the ranking in the following sense: if the σ ranks $\theta_i \prec \theta_j$, then $\|\theta_i - r_\sigma\| < \|\theta_j - r_\sigma\|$. Let $\Sigma_{n,d}$ denote the set of all possible rankings of the n objects that satisfy this embedding condition.

The interpretation of this assumption is that we know how the objects are related (in the embedding), which limits the space of possible rankings. The ranking to be learned, specified by the reference (e.g., preferences of a human subject), is unknown. Many have studied the problem of finding an embedding of objects from data [3, 4, 5]. This is not the focus here, but it could certainly play a supporting role in our methodology (e.g., the embedding could be determined from known similarities between the n objects, as is done in our experiments with the audio dataset). We assume the embedding is given and our interest is minimizing the number of queries needed to learn the ranking, and for this we require a second assumption.

A2 Consistency: Every pairwise comparison is consistent with the ranking to be learned. That is, if the reference ranks $\theta_i \prec \theta_j$, then θ_i must precede θ_j in the (full) ranking.

As we will discuss later in Section 3.2, these two assumptions alone are not enough to rule out pathological arrangements of objects in the embedding for which at least $\Omega(n)$ queries must be made to recover the ranking. However, because such situations are not representative of what is typically encountered, we analyze the problem in the framework of the average-case analysis [6].

Definition 1. *With each ranking $\sigma \in \Sigma_{n,d}$ we associate a probability π_σ such that $\sum_{\sigma \in \Sigma_{n,d}} \pi_\sigma = 1$. Let π denote these probabilities and write $\sigma \sim \pi$ for shorthand. The uniform distribution corresponds to $\pi_\sigma = |\Sigma_{n,d}|^{-1}$ for all $\sigma \in \Sigma_{n,d}$, and we write $\sigma \sim \mathcal{U}$ for this special case.*

Definition 2. *If $M_n(\sigma)$ denotes the number of pairwise comparisons requested by an algorithm to identify the ranking σ, then the* average query complexity *with respect to π is denoted by $\mathbb{E}_\pi[M_n]$.*

The main results are proven for the special case of $\pi = \mathcal{U}$, the uniform distribution, to make the analysis more transparent and intuitive. However the results can easily be extended to general distributions π that satisfy certain mild conditions [7]. All results henceforth, unless otherwise noted, will be given in terms of (uniform) average query complexity and we will say such results hold "on average."

Our main results can be summarized as follows. If the queries are chosen deterministically or randomly in advance of collecting the corresponding pairwise comparisons, then we show that almost all $\binom{n}{2}$ pairwise comparisons queries are needed to identify a ranking under the assumptions above. However, if the queries are selected in an adaptive and sequential fashion according to the algorithm

2

Query Selection Algorithm

input: n objects in $\mathbb{R}^d$

initialize: objects $\theta_1, \ldots, \theta_n$ in uniformly random order

for j=2,...,n
  for i=1,...,j-1
    if $q_{i,j}$ is *ambiguous*,
      request $q_{i,j}$'s label from reference;
    else
      impute $q_{i,j}$'s label from previously labeled queries.

output: ranking of n objects
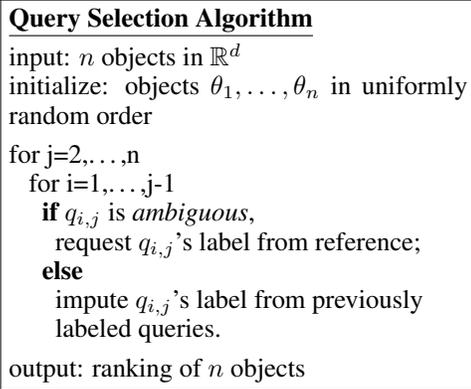
Figure 1: Sequential algorithm for selecting queries. See Figure 2 and Section 4.2 for the definition of an ambiguous query.
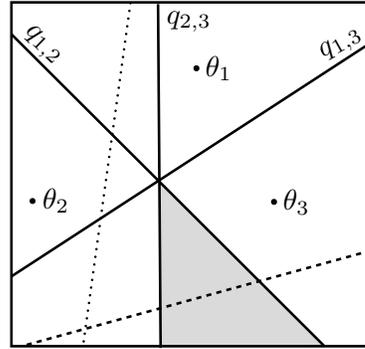
Figure 2: Objects $\theta_1, \theta_2, \theta_3$ and queries. The r_σ lies in the shaded region (consistent with the labels of $q_{1,2}, q_{1,3}, q_{2,3}$). The dotted (dashed) lines represent new queries whose labels are (are not) ambiguous given those labels.

in Figure 1, then we show that the number of pairwise rankings required to identify a ranking is no more than a constant multiple of $d \log n$, on average. The algorithm requests a query if and only if the corresponding pairwise ranking is ambiguous (see Section 4.2), meaning that it cannot be determined from previously collected pairwise comparisons and the locations of the objects in $\mathbb{R}^d$. The efficiency of the algorithm is due to the fact that *most* of the queries are unambiguous when considered in a sequential fashion. For this very same reason, picking queries in a non-adaptive or random fashion is very inefficient. It is also noteworthy that the algorithm is also computationally efficient with an overall complexity no greater than $O(n \operatorname{poly}(d) \operatorname{poly}(\log n))$ [7]. In Section 5 we present a robust version of the algorithm of Figure 1 that is tolerant to a fraction of errors in the pairwise comparison queries. In the case of *persistent errors* (see Section 5) we show that at least $O(n/\log n)$ objects can be correctly ranked in a partial ranking with high probability by requesting just $O(d \log^2 n)$ pairwise comparisons. This allows us to handle situations in which either or both of the assumptions, **A1** and **A2**, are reasonable approximations to the situation at hand, but do not hold strictly (which is the case in our experiments with the audio dataset).

Proving the main results involves an uncommon marriage of ideas from the ranking and statistical learning literatures. Geometrical interpretations of our problem derive from the seminal works of [8] in ranking and [9] in learning. From this perspective our problem bears a strong resemblance to the halfspace learning problem, with two crucial distinctions. In the ranking problem, the underlying halfspaces are not in general position and have strong dependencies with each other. These dependencies invalidate many of the typical analyses of such problems [10, 11]. One popular method of analysis in exact learning involves the use of something called the *extended teaching dimension* [12]. However, because of the possible pathological situations alluded to earlier, it is easy to show that the extended teaching dimension must be at least $\Omega(n)$ making that sort of worst-case analysis uninteresting. These differences present unique challenges to learning.

2 Motivation and related work

The problem of learning a ranking from few pairwise comparisons is motivated by what we perceive as a significant gap in the theory of ranking and permutation learning. Most work in ranking assumes a passive approach to learning; pairwise comparisons or partial rankings are collected in a random or non-adaptive fashion and then aggregated to obtain a full ranking (cf. [13, 14, 15, 16]). However, this may be quite inefficient in terms of the number of pairwise comparisons or partial rankings needed to learn the (full) ranking. This inefficiency was recently noted in the related area of social choice theory [17]. Furthermore, empirical evidence suggests that, even under complex ranking models, adaptively selecting pairwise comparisons can reduce the number needed to learn the ranking [18]. It is cause for concern since in many applications it is expensive and time-consuming to obtain pairwise comparisons. For example, psychologists and market researchers collect pairwise comparisons to gauge human preferences over a set of objects, for scientific understanding or product placement. The scope of these experiments is often very limited simply due to the time and expense required

3

to collect the data. This suggests the consideration of more selective and judicious approaches to gathering inputs for ranking. We are interested in taking advantage of underlying structure in the set of objects in order to choose more informative pairwise comparison queries. From a learning perspective, our work adds an active learning component to a problem domain that has primarily been treated from a passive learning mindset.

We focus on pairwise comparison queries for two reasons. First, pairwise comparisons admit a halfspace representation in embedding spaces which allows for a geometrical approach to learning in such structured ranking spaces. Second, pairwise comparisons are the most common form of queries in many applications, especially those involving human subjects. For example, consider the problem of finding the most highly ranked object, as illustrated by the following familiar task. Suppose a patient needs a new pair of prescription eye lenses. Faced with literally millions of possible prescriptions, the doctor will present candidate prescriptions in a sequential fashion followed by the query: better or worse? Even if certain queries are repeated to account for possible inaccurate answers, the doctor can locate an accurate prescription with just a handful of queries. This is possible presumably because the doctor understands (at least intuitively) the intrinsic space of prescriptions and can efficiently search through it using only binary responses from the patient.

We assume that the objects can be embedded in $\mathbb{R}^d$ and that the distances between objects and the reference are consistent with the ranking (Assumption **A1**). The problem of learning a general function $f : \mathbb{R}^d \to \mathbb{R}$ using just pairwise comparisons that correctly ranks the objects embedded in $\mathbb{R}^d$ has previously been studied in the passive setting [13, 14, 15, 16]. The main contributions of this paper are theoretical bounds for the specific case when $f(x) = ||x - r_\sigma||$ where $r_\sigma \in \mathbb{R}^d$ is the reference point. This is a standard model used in multidimensional unfolding and psychometrics [8, 19]. We are unaware of any existing query-complexity bounds for this problem. We do not assume a generative model is responsible for the relationship between rankings to embeddings, but one could. For example, the objects might have an embedding (in a feature space) and the ranking is generated by distances in this space. Or alternatively, structural constraints on the space of rankings could be used to generate a consistent embedding. Assumption **A1**, while arguably quite natural/reasonable in many situations, significantly constrains the set of possible rankings.

3 Geometry of rankings from pairwise comparisons

The embedding assumption **A1** gives rise to geometrical interpretations of the ranking problem, which are developed in this section. The pairwise comparison $q_{i,j}$ can be viewed as the membership query: is θ_i ranked before θ_j in the (full) ranking σ? The geometrical interpretation is that $q_{i,j}$ requests whether the reference r_σ is closer to object θ_i or object θ_j in $\mathbb{R}^d$. Consider the line connecting θ_i and θ_j in $\mathbb{R}^d$. The hyperplane that bisects this line and is orthogonal to it defines two halfspaces: one containing points closer to θ_i and the other the points closer to θ_j. Thus, $q_{i,j}$ is a membership query about which halfspace r_σ is in, and there is an equivalence between each query, each pair of objects, and the corresponding bisecting hyperplane. The set of all possible pairwise comparison queries can be represented as $\binom{n}{2}$ distinct halfspaces in $\mathbb{R}^d$. The intersections of these halfspaces partition $\mathbb{R}^d$ into a number of cells, and each one corresponds to a unique ranking of Θ. Arbitrary rankings are not possible due to the embedding assumption **A1**, and recall that the set of rankings possible under **A1** is denoted by $\Sigma_{n,d}$. The cardinality of $\Sigma_{n,d}$ is equal to the number of cells in the partition. We will refer to these cells as d-cells (to indicate they are subsets in d-dimensional space) since at times we will also refer to lower dimensional cells; e.g., $(d-1)$-cells.

3.1 Counting the number of possible rankings

The following lemma determines the cardinality of the set of rankings, $\Sigma_{n,d}$, under assumption **A1**.

Lemma 1. *[8] Assume A1-2. Let $Q(n, d)$ denote the number of d-cells defined by the hyperplane arrangement of pairwise comparisons between these objects (i.e. $Q(n, d) = |\Sigma_{n,d}|$). $Q(n, d)$ satisfies the recursion*

$$Q(n, d) = Q(n - 1, d) + (n - 1)Q(n - 1, d - 1) \,, \text{ where } Q(1, d) = 1 \text{ and } Q(n, 0) = 1. \quad (3)$$

In the hyperplane arrangement induced by the n objects in d dimensions, each hyperplane is intersected by every other and is partitioned into $Q(n-1, d-1)$ subsets or $(d-1)$-cells. The recursion,

4

above, arises by considering the addition of one object at a time. Using this lemma in a straightforward fashion, we prove the following corollary in [7].

Corollary 1. *Assume A1-2. There exist positive real numbers k_1 and k_2 such that*

$$k_1 \frac{n^{2d}}{2^d d!} < Q(n,d) < k_2 \frac{n^{2d}}{2^d d!}$$

for $n > d + 1$. If $n \leq d + 1$ then $Q(n,d) = n!$. For n sufficiently large, $k_1 = 1$ and $k_2 = 2$ suffice.

3.2 Lower bounds on query complexity

Since the cardinality of the set of possible rankings is $|\Sigma_{n,d}| = Q(n,d)$, we have a simple lower bound on the number of queries needed to determine the ranking.

Theorem 1. *Assume A1-2. To reconstruct an arbitrary ranking $\sigma \in \Sigma_{n,d}$ any algorithm will require at least $\log_2 |\Sigma_{n,d}| = \Theta(2d \log_2 n)$ pairwise comparisons.*

Proof. By Corollary 1 $|\Sigma_{n,d}| = \Theta(n^{2d})$, and so at least $2d \log n$ bits are needed to specify a ranking. Each pairwise comparison provides at most one bit. □

If each query provides a full bit of information about the ranking, then we achieve this lower bound. For example, in the one-dimensional case ($d = 1$) the objects can be ordered and binary search can be used to select pairwise comparison queries, achieving the lower bound. This is generally impossible in higher dimensions. Even in two dimensions there are placements of the objects (still in general position) that produce d-cells in the partition induced by queries that have $n - 1$ faces (i.e., bounded by $n - 1$ hyperplanes) as shown in [7]. It follows that the worst case situation may require at least $n - 1$ queries in dimensions $d \geq 2$. In light of this, we conclude that worst case bounds may be overly pessimistic indications of the typical situation, and so we instead consider the average case performance introduced in Section 1.1.

3.3 Inefficiency of random queries

The geometrical representation of the ranking problem reveals that randomly choosing pairwise comparison queries is inefficient relative to the lower bound above. To see this, suppose m queries were chosen uniformly at random from the possible $\binom{n}{2}$. The answers to m queries narrows the set of possible rankings to a d-cell in $\mathbb{R}^d$. This d-cell may consist of one or more of the d-cells in the partition induced by all queries. If it contains more than one of the partition cells, then the underlying ranking is ambiguous.

Theorem 2. *Assume A1-2. Let $N = \binom{n}{2}$. Suppose m pairwise comparison are chosen uniformly at random without replacement from the possible $\binom{n}{2}$. Then for all positive integers $N \geq m \geq d$ the probability that the m queries yield a unique ranking is $\binom{m}{d}/\binom{N}{d} \leq (\frac{em}{N})^d$.*

Proof. No fewer than d hyperplanes bound each d-cell in the partition of $\mathbb{R}^d$ induced by all possible queries. The probability of selecting d specific queries in a random draw of m is equal to

$$\binom{N-d}{m-d} / \binom{N}{m} = \binom{m}{d} / \binom{N}{d} \leq \frac{m^d}{d!} \frac{d^d}{N^d} \leq \left(\frac{m}{N}\right)^d \frac{d^d}{d!} \leq \left(\frac{em}{N}\right)^d. \qquad \square$$

Note that $\binom{m}{d}/\binom{N}{d} < 1/2$ unless $m = \Omega(n^2)$. Therefore, if the queries are randomly chosen, then we will need to ask almost all queries to guarantee that the inferred ranking is probably correct.

4   Analysis of sequential algorithm for query selection

Now consider the basic sequential process of the algorithm in Figure 1. Suppose we have ranked $k - 1$ of the n objects. Call these objects 1 through $k - 1$. This places the reference r_σ within a d-cell (defined by the labels of the comparison queries between objects $1, \ldots, k - 1$). Call this d-cell C_{k-1}. Now suppose we pick another object at random and call it object k. A comparison query between object k and one of objects $1, \ldots, k - 1$ can only be informative (i.e., ambiguous) if the associated hyperplane intersects this d-cell C_{k-1} (see Figure 2). If k is significantly larger than d, then it turns out that the cell C_{k-1} is probably quite small and the probability that one of the queries intersects C_{k-1} is very small; in fact the probability is on the order of $1/k^2$.

5

4.1 Hyperplane-point duality

Consider a hyperplane $\mathbf{h} = (h_0, h_1, \ldots, h_d)$ with $(d+1)$ parameters in $\mathbb{R}^d$ and a point $\mathbf{p} = (p_1, \ldots, p_d) \in \mathbb{R}^d$ that does not lie on the hyperplane. Checking which halfspace $\mathbf{p}$ falls in, i.e., $h_1 p_1 + h_2 p_2 + \cdots + h_d p_d + h_0 \gtrless 0$, has a dual interpretation: $\mathbf{h}$ is a point in $\mathbb{R}^{d+1}$ and $\mathbf{p}$ is a hyperplane in $\mathbb{R}^{d+1}$ passing through the origin (i.e., with d free parameters).

Recall that each possible ranking can be represented by a reference point $r_\sigma \in \mathbb{R}^d$. Our problem is to determine the ranking, or equivalently the vector of responses to the $\binom{n}{2}$ queries represented by hyperplanes in $\mathbb{R}^d$. Using the above observation, we see that our problem is equivalent to finding a labeling over $\binom{n}{2}$ points in $\mathbb{R}^{d+1}$ with as few queries as possible. We will refer to this alternative representation as the dual and the former as the primal.

4.2 Characterization of an ambiguous query

The characterization of an ambiguous query has interpretations in both the primal and dual spaces. We will now describe the interpretation in the dual which will be critical to our analysis of the sequential algorithm of Figure 1.

Definition 3. *[9] Let S be a finite subset of $\mathbb{R}^d$ and let $S^+ \subset S$ be points labeled $+1$ and $S^- = S \setminus S^+$ be the points labeled -1 and let x be any other point except the origin. If there exists two homogeneous linear separators of S^+ and S^- that assign different labels to the point x, then the label of x is said to be* ambiguous *with respect to S.*

Lemma 2. *[9, Lemma 1] The label of x is ambiguous with respect to S if and only if S^+ and S^- are homogeneously linearly separable by a $(d-1)$-dimensional subspace containing x.*

Let us consider the implications of this lemma to our scenario. Assume that we have labels for all the pairwise comparisons of $k-1$ objects. Next consider a new object called object k. In the dual, the pairwise comparison between object k and object i, for some $i \in \{1, \ldots, k-1\}$, is ambiguous if and only if there exists a hyperplane that still separates the original points and also passes through this new point. In the primal, this separating hyperplane corresponds to a point lying on the hyperplane defined by the associated pairwise comparison.

4.3 The probability that a query is ambiguous

An essential component of the sequential algorithm of Figure 1 is the initial random order of the objects; every sequence in which it could consider objects is equally probable. This allows us to state a nontrivial fact about the partial rankings of the first k objects observed in this sequence.

Lemma 3. *Assume A1-2 and $\sigma \sim \mathcal{U}$. Consider the subset $S \subset \Theta$ with $|S| = k$ that is randomly selected from Θ such that all $\binom{n}{k}$ subsets are equally probable. If $\Sigma_{k,d}$ denotes the set of possible rankings of these k objects then every $\sigma \in \Sigma_{k,d}$ is equally probable.*

Proof. Let a k-partition denote the partition of $\mathbb{R}^d$ into $Q(k, d)$ d-cells induced by k objects for $1 \le k \le n$. In the n-partition, each d-cell is weighted uniformly and is equal to $1/Q(n, d)$. If we uniformly at random select k objects from the possible n and consider the k-partition, each d-cell in the k-partition will contain one or more d-cells of the n-partition. If we select one of these d-cells from the k-partition, on average there will be $Q(n, d)/Q(k, d)$ d-cells from the n-partition contained in this cell. Therefore the probability mass in each d-cell of the k-partition is equal to the number of cells from the n-partition in this cell multiplied by the probability of each of those cells from the n-partition: $Q(n, d)/Q(k, d) \times 1/Q(n, d) = 1/Q(k, d)$, and $|\Sigma_{k,d}| = Q(k, d)$. $\square$

As described above, for $1 \le i \le k$ some of the pairwise comparisons $q_{i,k+1}$ may be ambiguous. The algorithm chooses a random sequence of the n objects in its initialization and does not use the labels of $q_{1,k+1}, \ldots, q_{j-1,k+1}, q_{j+1,k+1}, \ldots, q_{k,k+1}$ to make a determination of whether or not $q_{j,k+1}$ is ambiguous. It follows that the events of requesting the label of $q_{i,k+1}$ for $i = 1, 2, \ldots, k$ are independent and identically distributed (conditionally on the results of queries from previous steps). Therefore it makes sense to talk about the probability of requesting any one of them.

Lemma 4. *Assume A1-2 and $\sigma \sim \mathcal{U}$. Let $A(k, d, \mathcal{U})$ denote the probability of the event that the pairwise comparison $q_{i,k+1}$ is ambiguous for $i = 1, 2, \ldots, k$. Then there exists positive, real number constants a_1 and a_2 independent of k such that for $k > 2d$, $a_1 \frac{2d}{k^2} \le A(k, d, \mathcal{U}) \le a_2 \frac{2d}{k^2}$.*

6

Proof. By Lemma 2, a point in the dual (pairwise comparison) is ambiguous if and only if there exists a separating hyperplane that passes through this point. This implies that the hyperplane representation of the pairwise comparison in the primal intersects the cell containing r_σ (see Figure 2 for an illustration of this concept). Consider the partition of $\mathbb{R}^d$ generated by the hyperplanes corresponding to pairwise comparisons between objects $1, \ldots, k$. Let $P(k,d)$ denote the number of d-cells in this partition that are intersected by a hyperplane corresponding to one of the queries $q_{i,k+1}$, $i \in \{1, \ldots, k\}$. Then it is not difficult to show that $P(k,d)$ is bounded above and below by constants independent of n and k times $\frac{k^{2(d-1)}}{2^{d-1}(d-1)!}$ [7]. By Lemma 3, every d-cell in the partition induced by the k objects corresponds to an equally probable ranking of those objects. Therefore, the probability that a query is ambiguous is the number of cells intersected by the corresponding hyperplane divided by the total number of d-cells, and therefore $A(k,d,\mathcal{U}) = \frac{P(k,d)}{Q(k,d)}$. The result follows immediately from the bounds on $P(k,d)$ and Corollary 1. $\qquad\square$

Because the individual events of requesting each query are conditionally independent, the total number of queries requested by the algorithm is just $M_n = \sum_{k=1}^{n-1} \sum_{i=1}^{k} \mathbf{1}\{\text{Request } q_{i,k+1}\}$. Using the results above, it straightforward to prove the main theorem below (see [7]).

Theorem 3. *Assume A1-2 and $\sigma \sim \mathcal{U}$. Let the random variable M_n denote the number of pairwise comparisons that are requested in the algorithm of Figure 1, then*

$$\mathbb{E}_{\mathcal{U}}[M_n] \leq 2d \log_2 2d + 2da_2 \log n.$$

Furthermore, if $\sigma \sim \pi$ and $\max_{\sigma \in \Sigma_{n,d}} \pi_\sigma \leq c|\Sigma_{n,d}|^{-1}$ for some $c > 0$, then $\mathbb{E}_\pi[M_n] \leq c\mathbb{E}_{\mathcal{U}}[M_n]$.

5   Robust sequential algorithm for query selection

We now extend the algorithm of Figure 1 to situations in which the response to each query is only probably correct. If the correct label of a query $q_{i,j}$ is $y_{i,j}$, we denote the possibly incorrect response by $Y_{i,j}$. The probability that $Y_{i,j} = y_{i,j}$ is at least $1 - p$, $p < 1/2$. The robust algorithm operates in the same fashion as the algorithm in Figure 1, with the exception that when an ambiguous query is encountered several (equivalent) queries are made and a decision is based on the majority vote. This voting procedure allows us to construct a ranking (or partial ranking) that is correct with high probability by requesting just $O(d \log^2 n)$ queries where the extra $\log$ factor comes from voting.

First consider the case in which each query can be repeated to obtain multiple independent responses (votes) for each comparison query. This *random noise* model arises, for example, in social choice theory where the "reference" is a group of people, each casting a vote. The elementary proof of the next theorem is given in [7].

Theorem 4. *Assume A1-2 and $\sigma \sim \mathcal{U}$ but that each query response is a realization of an i.i.d. Bernoulli random variable $Y_{i,j}$ with $P(Y_{i,j} \neq y_{i,j}) \leq p < 1/2$. If all ambiguous queries are decided by the majority vote of R independent responses to each such query, then with probability greater than $1 - 2n \log_2(n) \exp(-\frac{1}{2}(1 - 2p)^2 R)$ this procedure correctly identifies the correct ranking and requests no more than $O(Rd \log n)$ queries on average.*

In other situations, if we ask the same query multiple times we may get the same, possibly incorrect, response each time. This *persistent noise* model is natural, for example, if the reference is a *single* human. Under this model, if two rankings differ by only a single pairwise comparison, then they cannot be distinguished with probability greater than $1 - p$. So, in general, exact recovery of the ranking cannot be guaranteed with high probability. The best we can hope for is to exactly recover a *partial ranking* of the objects (i.e. the ranking over a subset of the objects). Henceforth, we will assume the noise is persistent and aim to exactly recover a partial ranking of the objects.

The key ingredient in the persistent noise setting is the design of a voting set for each ambiguous query encountered. Suppose that at the jth object in the algorithm in Figure 1 the query $q_{i,j}$ is ambiguous. In principle, a voting set could be constructed using objects ranked between i and j. If object k is between i and j, then note that $y_{i,j} = y_{i,k} = y_{k,j}$. In practice, we cannot identify the subset of objects ranked between i and j, but it is contained within the set $T_{i,j}$, defined to be the subset of objects θ_k such that $q_{i,k}$, $q_{k,j}$, or both are ambiguous. Furthermore, Lemma 3 implies that each object in $T_{i,j}$ is ranked between i and j with probability at least $1/3$ [7]. $T_{i,j}$ will be our voting set. Note however, if objects i and j are closely ranked, then $T_{i,j}$ may be rather small, and so it is not

7
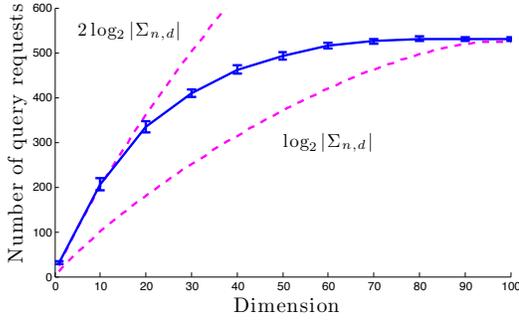
Figure 3: Mean and standard deviation of requested queries (solid) in the noiseless case for $n = 100$; $\log_2 |\Sigma_{n,d}|$ is a lower bound (dashed).

Table 1: Statistics for the algorithm robust to persistent noise of Section 5 with respect to all $\binom{n}{2}$ pairwise comparisons. Recall y is the noisy response vector, $\tilde{y}$ is the embedding's solution, and $\hat{y}$ is the output of the robust algorithm.

Dimension		2	3
% of queries requested	mean	14.5	18.5
	std	5.3	6
Average error	$d(y, \tilde{y})$	0.23	0.21
	$d(y, \hat{y})$	0.31	0.29

always possible to find a sufficiently large voting set. Therefore, we must specify a size-threshold $R \geq 1$. If the size of $T_{i,j}$ is at least R, then we decide the label for $q_{i,j}$ by voting over the responses to $\{q_{i,k}, q_{k,j} : k \in T_{i,j}\}$ and $q_{i,j}$; otherwise we pass over object j and move on to the next object in the list. This allows us to construct a probably correct ranking of the objects that are not passed over. The theorem below proves that a large portion of objects will not be passed over. At the end of the process, some objects that were passed over may then be unambiguously ranked (based on queries made after they were passed over) or they can be ranked without voting (and without guarantees). The proof of the next theorem is provided in the longer version of this paper [7].

Theorem 5. *Assume A1-2, $\sigma \sim \mathcal{U}$, and $P(Y_{i,j} \neq y_{i,j}) = p$. For any size-threshold $R \geq 1$, with probability greater than $1 - 2n \log_2(n) \exp\left(-\frac{2}{9}(1-2p)^2 R\right)$ the procedure above correctly ranks at least $n/(2R+1)$ objects and requests no more than $O(Rd \log n)$ queries on average.*

6 Empirical results

In this section we present empirical results for both the noiseless algorithm of Figure 1 and the robust algorithm of Section 5. For the noiseless algorithm, $n = 100$ points, representing the objects to be ranked, were uniformly at random simulated from the unit hypercube $[0,1]^d$ for $d = 1, 10, 20, \ldots, 100$. The reference was simulated from the same distribution. For each value of d the experiment was repeated 25 times using a new simulation of points and the reference. Because responses are noiseless, exact identification of the ranking is guaranteed. The number of requested queries is plotted in Figure 3 with the lower bound of Theorem 1 for reference. The number of requested queries never exceeds twice the lower bound which agrees with the result of Theorem 3.

The robust algorithm in Section 5 was evaluated using a symmetric similarity matrix dataset available at [20] whose (i,j)th entry, denoted $s_{i,j}$, represents the human-judged similarity between audio signals i and j for all $i \neq j \in \{1, \ldots, 100\}$. If we consider the kth row of this matrix, we can rank the other signals with respect to their similarity to the kth signal; we define $q_{i,j}^{(k)} := \{s_{k,i} > s_{k,j}\}$ and $y_{i,j}^{(k)} := \mathbf{1}\{q_{i,j}^{(k)}\}$. Since the similarities were derived from human subjects, the derived labels may be erroneous. Moreover, there is no possibility of repeating queries here and so the noise is persistent. The analysis of this dataset in [2] suggests that the relationship between signals can be well approximated by an embedding in 2 or 3 dimensions. We used *non-metric multidimensional scaling* [5] to find an embedding of the signals: $\theta_1, \ldots, \theta_{100} \in \mathbb{R}^d$ for $d = 2$ and 3. For each object θ_k, we use the embedding to derive pairwise comparison labels between all other objects as follows: $\tilde{y}_{i,j}^{(k)} := \mathbf{1}\{||\theta_k - \theta_i|| < ||\theta_k - \theta_j||\}$, which can be considered as the best approximation to the labels $y_{i,j}^{(k)}$ (defined above) in this embedding. The output of the robust sequential algorithm, which uses only a small fraction of the similarities, is denoted by $\hat{y}_{i,j}^{(k)}$. We set $R = 15$ using Theorem 5 as a rough guide. Using the popular Kendell-Tau distance $d(y^{(k)}, \hat{y}^{(k)}) = \binom{n}{2}^{-1} \sum_{i<j} \mathbf{1}\{y_{i,j}^{(k)} \neq \hat{y}_{i,j}^{(k)}\}$ [21] for each object k, we denote the average of this metric over all objects by $d(y, \hat{y})$ and report this statistic and the number of queries requested in Table 1. Because the average error of $\hat{y}$ is only 0.07 higher than that of $\tilde{y}$, this suggests that the algorithm is doing almost as well as we could hope. Also, note that $2R \, 2d \log n / \binom{n}{2}$ is equal to 11.4% and 17.1% for $d = 2$ and 3, respectively, which agrees well with the experimental values.

8

References

[1] D. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 1998.

[2] Scott Philips, James Pitton, and Les Atlas. Perceptual feature identification for active sonar echoes. In *OCEANS 2006*, 2006.

[3] B. McFee and G. Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 721–728. ACM, 2009.

[4] I. Gormley and T. Murphy. A latent space model for rank data. *Statistical Network Analysis: Models, Issues, and New Directions*, pages 90–102, 2007.

[5] M.A.A. Cox and T.F. Cox. Multidimensional scaling. *Handbook of data visualization*, pages 315–347, 2008.

[6] J.F. Traub. *Information-based complexity*. John Wiley and Sons Ltd., 2003.

[7] Kevin G. Jamieson and Robert D. Nowak. Active ranking using pairwise comparisons. *arXiv:1109.3701v1*, 2011.

[8] C.H. Coombs. A theory of data. *Psychological review*, 67(3):143–159, 1960.

[9] T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, 14(3):326–334, 1965.

[10] S. Dasgupta, A.T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *The Journal of Machine Learning Research*, 10:281–299, 2009.

[11] S. Hanneke. *Theoretical foundations of active learning*. PhD thesis, Citeseer, 2009.

[12] Tibor Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, COLT '95, pages 108–117, New York, NY, USA, 1995. ACM.

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

[14] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[15] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294. ACM, 2007.

[16] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 1, pages 97–102. IET, 1999.

[17] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. *IJCAI-11, Barcelona*, 2011.

[18] W. Chu and Z. Ghahramani. Extensions of gaussian processes for ranking: semi-supervised and active learning. *Learning to Rank*, page 29, 2005.

[19] J.F. Bennett and W.L. Hays. Multidimensional unfolding: Determining the dimensionality of ranked preference data. *Psychometrika*, 25(1):27–43, 1960.

[20] Similarity Learning. Aural Sonar dataset. [http://idl.ee.washington.edu/SimilarityLearning]. University of Washington Information Design Lab, 2011.

[21] J.I. Marden. *Analyzing and modeling rank data*. Chapman & Hall/CRC, 1995.

9