Active Ranking in Practice: General Ranking Functions with Sample Complexity Bounds

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 a companion paper in the regular NIPS 2011 program [1], we showed that if each object $x \in \mathbb{R}^d$ is assigned a score $f(x) = ||x - r||$ for some unknown $r \in \mathbb{R}^d$, then our recently proposed active ranking algorithm can recover the ranking of the scores using about $d \log n$ selectively chosen pairwise comparisons. Here we show that this same model contains all functions of the type $g(x) = w^T x$ for some unknown $w \in \mathbb{R}^d$, thus the same bound applies. We take advantage of this fact and use kernel methods to represent more general ranking functions. This extension includes popular ranking methods such as RankSVM, and we derive nontrivial query complexity bounds for active versions of such algorithms. The efficacy of the theory and method are demonstrated by applying our kernelized adaptive algorithm to two real datasets.

1 Problem statement

Given a set of n objects $\Theta := \{\theta_1, \dots, \theta_n\}$, we wish to discover how an oracle *ranks* these objects. The ranking, denoted by σ, can be thought of as a mapping $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ that prescribes an order

$$\sigma(\{\theta_i\}_{i=1}^n) := \theta_{\sigma(1)} \prec \theta_{\sigma(2)} \prec \cdots \prec \theta_{\sigma(n-1)} \prec \theta_{\sigma(n)} \tag{1}$$

where $\theta_i \prec \theta_j$ means θ_i precedes, or is preferred to, θ_j in the oracle's ranking. The ranking can be learned by querying the oracle for pairwise comparisons 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.

We define a *ranking function* to be $f : \Theta \to \mathbb{R}$ such that

$$\theta_i \prec \theta_j \iff f(\theta_i) < f(\theta_j). \tag{2}$$

We say two ranking functions f and g are *equivalent* if both ranking functions correspond to the same ranking σ. In general, there are $n!$ ways to permute n objects and we can always find an f that obeys (2) for any desired permutation. However, we assume that the oracle's ranking function belongs to a certain class denoted by $\mathcal{F}$, which may limit the set of possible rankings. Given a set of objects Θ and a ranking function class $\mathcal{F}$, we denote this constrained set of possible rankings by $\Sigma_{\Theta,\mathcal{F}}$. While $\mathcal{F}$ may be uncountably infinite, because of the equivalence of ranking functions, $\Sigma_{\Theta,\mathcal{F}}$ is a subset of $\mathbb{S}_n$ (symmetric group over n objects) and so its cardinality $|\Sigma_{\Theta,\mathcal{F}}|$ is at most $n!$.

2 Main theoretical results

We proposed an active approach to learning rankings in a companion paper in the NIPS 2011 conference [1]. In that paper, we show that if $\mathcal{F} := \{f(\theta) = ||\phi(\theta) - r||,\ r \in \mathbb{R}^d\}$ where $\phi : \Theta \to \mathbb{R}^d$ is fixed and known, then we can discover a ranking selected uniformly at random from the set $\Sigma_{\Theta,\mathcal{F}}$

by requesting just $O(2d \log n)$ pairwise comparisons, in expectation. The main contributions of this workshop paper are to extend this theoretical result to a much broader and practically relevant range of general ranking functions and to evaluate the performance of such methods on real-world datasets. Our first new contribution is the following lemma which relates the class of ranking functions above to the more common class of linear ranking functions (the proof can be found in Appendix A.1). For the remainder of this paper, assume that ϕ is fixed and known.

Lemma 1. *Let* $\phi : \Theta \to \mathbb{R}^d$. *Let* $\mathcal{F} = \{f(\theta) = ||\phi(\theta) - r||\,,\, r \in \mathbb{R}^d\}$ *and let* $\mathcal{G} = \{g(\theta) = w^T\phi(\theta)\,,\, w \in \mathbb{R}^d\}$. *Then* $\Sigma_{\Theta,\mathcal{G}} \subset \Sigma_{\Theta,\mathcal{F}}$.

One may recognize the ranking function class $\mathcal{G}$ to be the class considered in the popular RankSVM model [2]. Therefore, by the above lemma, we can study the class $\mathcal{F}$ to bound the query complexity of the ranking functions of the type $g(\theta) = w^T\phi(\theta)$. Moreover, just like kernel methods are often applied to the functions in $\mathcal{G}$, we show that the same is possible for the class $\mathcal{F}$. This extension allows the known theoretical results to be applied to very complicated ranking functions. We say a kernel $\kappa(\theta, \theta')$ is *d-dimensional* if $\kappa(\theta_i, \theta_j) = \phi(\theta_i)^T\phi(\theta_j)$ where $\phi : \Theta \to \mathbb{R}^d$ and $\phi(\theta_1), \dots, \phi(\theta_j)$ are in general position in $\mathbb{R}^d$. The proof of the next lemma is nontrivial and can be found in Appendix A.2.

Lemma 2. *Let* $\phi : \Theta \to \mathbb{R}^d$, $\kappa(\theta, \theta') := \phi(\theta)^T\phi(\theta')$ *be a d-dimensional kernel, and* $\mathcal{F}' := \{f(\theta) = \kappa(\theta, \theta) + \sum_{i=1}^n \kappa(\theta_i, \theta)\alpha_i\,,\, \alpha \in \mathbb{R}^n\}$. *Then* $\Sigma_{\Theta,\mathcal{F}'} = \Sigma_{\Theta,\mathcal{F}}$. *Moreover, if* $\mathcal{G}' := \{f(\theta) = \sum_{i=1}^n \kappa(\theta_i, \theta)\alpha_i\,,\, \alpha \in \mathbb{R}^n\}$, *then* $\Sigma_{\Theta,\mathcal{G}'} \subset \Sigma_{\Theta,\mathcal{F}}$.

Remark 1. For infinite dimensional kernels (e.g. a Gaussian kernel defined over $\mathbb{R}^d$), the ranking function class may not reduce the number of rankings from $n!$ due to the flexibility of such kernels. In such cases any algorithm must request $\Omega(n \log n)$ pairwise comparisons. However, if we constrain the ranking function class to contain only "smooth" functions by introducing a regularizer, then experimentally we have observed substantial sample complexity gains using the the algorithm in Figure 1. See Appendix A.3 for a full discussion.

We now state our main result in a theorem. The bound in the theorem pertains to the active ranking algorithm in Figure 1. The key idea of the algorithm is to sequentially pass over all possible queries (in a randomized order), requesting a query if and only if it is *ambiguous* based on previous queries. Testing for ambiguity is based on an efficient linear program, as described in [1].

Theorem 1. *Assume all requested pairwise comparisons are correct and transitive with probability 1. Let* $\kappa(\theta, \theta')$ *be a d-dimensional kernel and let* $\mathcal{F} = \{f(\theta) = \kappa(\theta, \theta) + \sum_{i=1}^n \kappa(\theta_i, \theta)\alpha_i\,,\, \alpha \in \mathbb{R}^n\}$. *For any ranking* $\sigma \in \Sigma_{\Theta,\mathcal{F}}$ *let* $\mathcal{Q}(\sigma, \mathcal{A})$ *denote the number queries required by algorithm* $\mathcal{A}$ *to discover the ranking* σ. *If* $\mathcal{A}$ *is the algorithm in Figure 1 then*

$$\mathbb{E}\left[\frac{1}{|\Sigma_{\Theta,\mathcal{F}}|} \sum_{\sigma \in \Sigma_{\Theta,\mathcal{F}}} \mathcal{Q}(\sigma, \mathcal{A})\right] \le 2cd \log n$$

where the expectation is with respect to the randomization of the algorithm and c is a constant. Furthermore $\inf_{\mathcal{A}} \frac{1}{|\Sigma_{\Theta,\mathcal{F}}|} \sum_{\sigma \in \Sigma_{\Theta,\mathcal{F}}} \mathcal{Q}(\sigma, \mathcal{A}) \ge \log_2 |\Sigma_{\Theta,\mathcal{F}}| = \Theta(d \log n)$ *where the infimum is taken over all query selection algorithms.*

The result is stated as an average over all possible rankings $\Sigma_{\Theta,\mathcal{F}}$ because while there exist contrived positionings of the points to force at least one ranking to require $\Omega(n)$ queries, this is necessarily atypical (see [1] for a discussion). The above theorem can also be interpreted as a Bayesian sort of statement: if we have a uniform prior over all possible rankings $\Sigma_{\Theta,\mathcal{F}}$ then the expected number of pairwise comparisons necessary to determine a ranking is bounded by $O(d \log n)$.

Remark 2. In practice, it is unlikely that responses to pairwise comparisons are correct with probability 1. In [1] we considered a model that assumes pairwise comparisons are flipped i.i.d. with probability $p < 1/2$ and that the errors are persistent (i.e. responses to queries do not change if repeatedly asked). The robust procedure proceeds just like the noiseless algorithm except that when an ambiguous query is encountered, a small set of related queries is requested and used to predict the ambiguous query. The robust procedure can be incorporated with the methods proposed here in a straightforward way to achieve a total sample complexity of just $O(d(1-2p)^{-2} \log^2 n)$ while making errors on only a small fraction of pairwise orderings.

The remainder of this paper proves the above results and applies the developed ideas to real datasets where significant qualitative and quantitative gains are observed.

```
Query Selection Algorithm
input: $n$ objects, ranking function class $\mathcal{F}$
initialize: objects $\theta_1, \dots, \theta_n$ 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 the oracle;
    else
      impute $q_{i,j}$'s label from previously labeled queries.
output: ranking of $n$ objects
```

Figure 1: Sequential query selection algorithm[1]. Let $q_{i,j} = \{\theta_i \prec \theta_j\}$ be a query. A query is ambiguous if its label cannot be inferred from previous responses to queries [1].

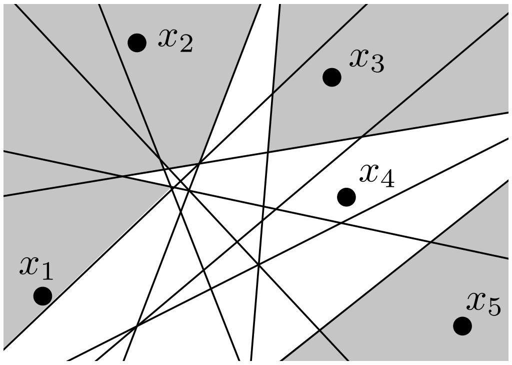


Figure 2: The d-cells (white and shaded) correspond to the rankings induced by $\mathcal{F} = \{f(x) = ||x - r||\,,\, r \in \mathbb{R}^d\}$ while the unbounded, shaded d-cells are induced by $\mathcal{G} = \{g(x) = w^T x\,,\, w \in \mathbb{R}^d\}$.

3 Related work

As we will see in the next section, the theoretical results of this paper are a straightforward, but important extension upon our recent previous work [1]. The main goal of this paper is to emphasize the practical importance of the theory and show how it performs in practice. We also wish to clearly state how our work compares to the popular RankSVM of [2]. We were not the first to consider an active strategy for collecting pairwise comparisons given structure about the objects; the potential benefits were made plain in [3, 4] using empirical studies. This problem was also studied from a theoretical perspective by [5] but in the difficult arbitrary-noise setting in which the authors bounded the query complexity by $O(dn \log^2 n)$. This bound is exponentially larger than the bound of Theorem 1 but this is likely to be unavoidable due to their general treatment of noise. Our primary interest is to characterize the query complexity of an active procedure that exploits this structure under noiseless (or bounded noise) conditions, something that was previously unknown. We believe our results to be much more satisfying from a practical perspective and that the analysis provides intuition that is absent in [5].

4 Analysis

It turns out that the space of rankings generated by the ranking function class $\mathcal{F} = \{f(x) = ||x - r||\,,\, r \in \mathbb{R}^d\}$ has a very intuitive geometric interpretation. We will start with a very simple example to provide some intuition and then generalize the results. Suppose we were not given the descriptions of the objects in $\mathbb{R}^d$ but still would like to find a ranking over the n objects by requesting pairwise comparisons (i.e. the standard sorting problem using pairwise comparisons). Let us assign each of the n objects in $\Theta = \{\theta_1, \dots, \theta_n\}$ a vector in $\mathbb{R}^n$ such that for each θ_i we assign the vector $x_i \in \mathbb{R}^n$ which has a 1 in the ith position and zeros everywhere else. Then all $n!$ rankings can be determine by some point $r \in \mathbb{R}^n$ such that $\theta_i \prec \theta_j \iff ||x_i - r|| < ||x_j - r||$. To see this, $||x_i - r|| - ||x_j - r|| = 2r^T(x_j - x_i) = 2(r^{(j)} - r^{(i)})$, where $r^{(i)}$ is the ith element of r, which means $\theta_i \prec \theta_j \iff r^{(j)} < r^{(i)}$. Therefore, if $n = 5$ and we wanted the ranking $\theta_3 \prec \theta_2 \prec \theta_4 \prec \theta_1 \prec \theta_5$, one solution for r is $(2, 4, 5, 3, 1)$. In fact, the x_i's in $\mathbb{R}^n$ only need to be in general position to guarantee the existence of such an $r \in \mathbb{R}^n$ that can determine any ranking. However, now consider an embedding of the x_i's in a d-dimensional subspace of $\mathbb{R}^n$, namely, $\mathbb{R}^d$. Does there still exist some point $r \in \mathbb{R}^d$ that can determine all $n!$ rankings? It turns out that there does not and for $d \ll n$, the number of possible rankings reduces to something like n^{2d}, a drastic reduction from $n!$. This discussion can be summarized in a lemma, thanks to Coombs [6].

Lemma 3. *[6] Let $x_1, \dots, x_n$ be a set of n points in $\mathbb{R}^d$ (in general position) describing the objects $\Theta = \{\theta_1, \dots, \theta_n\}$ where $x_i = \phi(\theta_i)$ for $i = 1, \dots, n$. For any $r_\sigma \in \mathbb{R}^d$ let σ_r define a ranking over the n points such that $\theta_i \prec \theta_j \iff ||x_i - r_\sigma|| < ||x_j - r_\sigma||$. Let $\Sigma_{\Theta,\mathcal{F}} = \cup_{r \in \mathbb{R}^d} \sigma_r$ denote the set of all possible rankings of the n objects that satisfy this ranking condition. If $Q(n, d)$ is equal to the*

[1] Code at `http://homepages.cae.wisc.edu/~jamieson/me/Active_Ranking.html`

number of distinct rankings (i.e., $Q(n,d) = |\Sigma_{\Theta,\mathcal{F}}|$), then $Q(n,d)$ satisfies the recursion

$$Q(n,d) = Q(n-1,d) + (n-1)Q(n-1,d-1) \tag{3}$$

where $Q(1,d) = 1$ and $Q(n,0) = 1$. Also, 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.

To see why the above lemma is true, one must understand the geometric interpretation of $x_i \prec x_j$ in $\mathbb{R}^d$. For ease of discussion, let $q_{i,j}$ denote the query $\{\theta_i \prec \theta_j\}$ and $y_{i,j}$ denote its label in $\{1,0\}$ denoting whether the pairwise ordering is true or false, respectively. The pairwise comparison can be viewed as the membership query: is θ_i ranked before θ_j in the ranking σ_r? The geometrical interpretation is that $q_{i,j}$ requests whether the reference r_σ is closer to object x_i or object x_j in $\mathbb{R}^d$. Consider the line connecting x_i and x_j in $\mathbb{R}^d$. The hyperplane that bisects this line and is orthogonal to it defines two halfspaces: one containing points closer to x_i and the other the points closer to x_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 (see Figure 2 for an illustration of this concept). Recall from Lemma 3 that the set of possible rankings is denoted by $\Sigma_{\Theta,\mathcal{F}}$. The cardinality of $\Sigma_{\Theta,\mathcal{F}}$ 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).

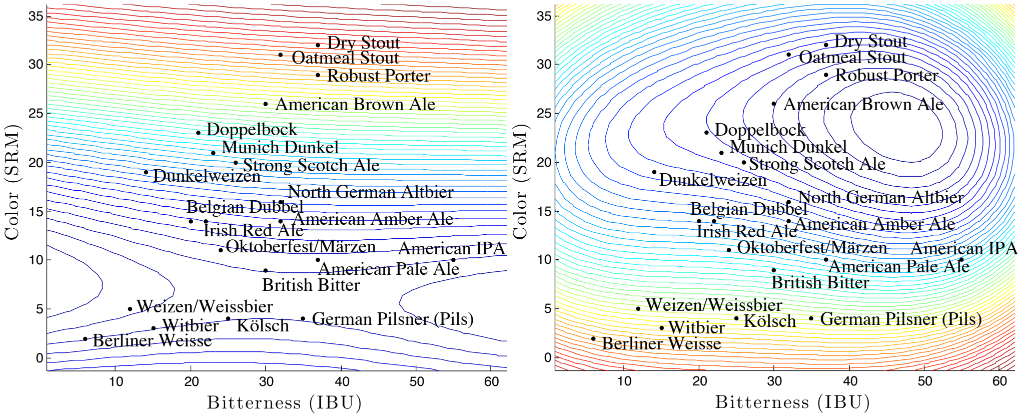
We now take a moment to describe the set $\Sigma_{\Theta,\mathcal{G}}$ that was introduced in Section 2. Recall from the above discussion that each ranking in $\Sigma_{\Theta,\mathcal{F}}$ has a one-to-one correspondence to a d-cell in $\mathbb{R}^d$. Some of these d-cells are bounded (i.e. their volumes are bounded) while others are unbounded (see Figure 2). $\Sigma_{\Theta,\mathcal{G}}$ corresponds to all those cells that are unbounded. In fact, the number of unbounded d-cells generated by n objects follows the same recursion as in Lemma 3 with the exception that $Q(n,0) = 0$ instead of 1 [6]. It is straightforward to show that $|\Sigma_{X,\mathcal{G}}| = \Theta(\frac{n^{2(d-1)}}{2^{d-1}(d-1)!})$.

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}. 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 d/k^2 (see [1] for a complete proof).

Because the individual events of requesting each query are conditionally independent, the expected number of queries requested by algorithm $\mathcal{A}$ to discover a ranking σ is just $\mathbb{E}[\mathcal{Q}(\sigma,\mathcal{A})] = \sum_{k=1}^{n-1}\sum_{i=1}^{k} P\{\text{Request } q_{i,k+1} | \text{labels to } q_{s\leq k,t\leq k}\}$. Using the results above, it straightforward to prove Theorem 1.

5 Empirical results

To demonstrate the performance of the algorithm in practice, we applied it to two real datasets. The first consists of measured quantities of 21 different styles of beer from the *beer judge certification program* (BJCP) [7]. The dataset includes several features (original gravity (OG), final gravity (FG), international bitter units (IBU), color depth (SRM), alcohol content (ABV)), but we chose to work with just two for visualization purposes: IBU and SRM. The feature vector for each beer lives in just $\mathbb{R}^2$ so this space is likely to be too simple to account for real preferences. For each pair of beers $x_i, x_j \in \mathbb{R}^2$ we applied the polynomial kernel $\kappa(x_i,x_j) = (1 + x_i^T x_j)^p$ for $p = 2$. We then ran the algorithm in Figure 1 and had the first author provide the requested pairwise comparisons. Plotted in Figure 3 are heat maps of the first author's preferences during two seasons. To learn the ranking functions, 25 and 26 pairwise comparisons were requested. Note the total number of possible comparisons is $\binom{21}{2} = 210$ and the number of comparisons needed by conventional sorting



(a) Summer

(b) Winter

Figure 3: Heat maps of the learned ranking functions (after a monotonic nonlinear transformation for visualization) of the first author's preferences over different styles of beer in Summer and Winter, respectively. The algorithm requested just 25 and 26 pairwise comparisons, respectively, to generate the ranking functions. Blue, cooler colors are preferred to red, warm colors.

algorithms is at least $\log_2 21! \approx 65$. The goal of this experiment was to show that the learned ranking functions, attained with relatively few queries, are very interpretable and can be used to predict preferences over objects that have yet to be observed or tasted.

In the second experiment we studied the tradeoff between the number of requested queries (which grows with increased complexity) and the error in the output ranking (which shrinks with increased complexity). We also study how our algorithm compares to the performance of an algorithm that selects queries uniformly at random without replacement (i.e. passive learning with a RankSVM-like algorithm). For this experiment we used a real dataset involving human-expert judgements of similarity between two audio signals that are echoes off rocks or submarines [8]. The task is simple: find the ranking of the nearest neighbors in the training set to each test signal by requesting as few comparisons from the human expert as possible. One could conceive of a system where this output could be fed into a larger system that classifies the test signal as either a rock or a submarine, and if so, what kind. The data is encoded as a symmetric similarity matrix available at [9] whose (i, j)th entry, denoted $s_{i,j}$, represents the human-judged similarity between audio signals θ_i and θ_j for all $i \neq j \in \{1, \ldots, 100\}$. Note that we do not have access to the signals θ_i and θ_j, we only have the similarity $s_{i,j}$. Ideally, the similarity matrix would be positive semi-definite (PSD) and we could define $\kappa(\theta_i, \theta_j) = s_{i,j}$. Unfortunately, this is not the case so we modified the similarity matrix in such a way that only the d largest eigenvalues for $d = 1, \ldots, 100$ are used to represent the similarity matrix. This can be thought of as finding the closest PSD matrix to the similarity matrix (see [10] for a discussion). If we consider the kth row of the similarity matrix, we can rank the other signals with respect to their similarity to the kth signal; we define $q^{(k)}_{i,j} := \{s_{k,i} > s_{k,j}\}$ and $y^{(k)}_{i,j} := \mathbf{1}\{q^{(k)}_{i,j}\}$. Because we are using a modified version of the similarity matrix for our kernel, the pairwise comparisons we request are not necessarily consistent with our model. This means that if the model is too simple (i.e., too few eigenvalues are used in the kernel) then the algorithm will likely output a very poor ranking due to its greediness. To measure the distance between any two rankings, we use a normalized version of the popular Kendell-Tau distance $d(y^{(k)}, \hat{y}^{(k)}) = \binom{n}{2}^{-1} \sum_{i<j} \mathbf{1}\{y^{(k)}_{i,j} \neq \hat{y}^{(k)}_{i,j}\}$ [11]. This metric is equal to the fraction of pairwise comparisons that the true and estimated rankings disagree on.

In the scatterplot of Figure 4 (note the log scale), each filled circle represents an average over 10 runs of the algorithm of Figure 1 for dimensions $d = 1, 5, 9, \ldots, 45$. Each run picks one of the signals uniformly at random from the total 100 and attempts to rank the other signals, as discussed above. Some of the filled circles have a number next to them indicating the dimension d used for that set of runs. Note how the number of queries is roughly proportional to the dimension, as predicted by Theorem 1. Also in Figure 4 we have plotted the results of a passive version of the algorithm with open circles that selected pairwise comparisons uniformly at random without replacement. Each

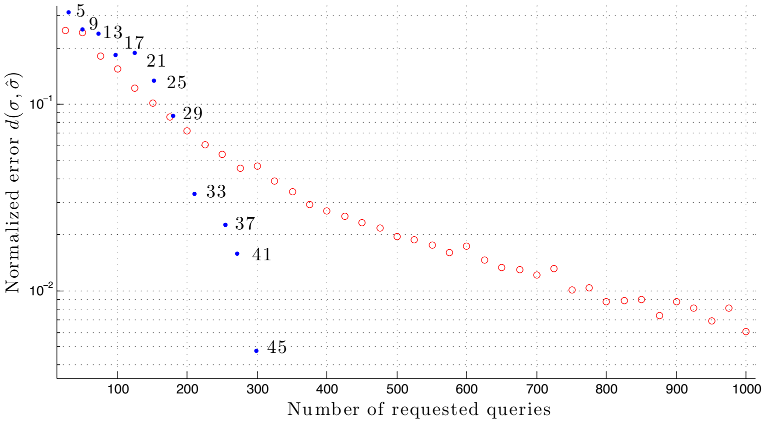


Figure 4: Each filled circle represents a randomly selected object that determines the ranking over the other objects for a certain dimension $d = \{1, 5, \ldots, 45\}$. As d increases, the model becomes more complex meaning that more queries are requested but the error decreases. Each open circle indicates the probability of error for the passive algorithm with the given number of pairwise comparisons with $d = 70$, the dimension that worked best. With $n = 99$, binary sort would require about $\log_2 n! \approx 518$ pairwise comparisons.

open circle represents the average over 100 runs using the plotted number of pairwise comparisons and $d = 70$, which is what we found to perform the best over a large range of d's. It is interesting to note that there are about 70 positive eigenvalues for the similarity matrix. We observe that if d is too small, the model does not fit the data well and the algorithm outputs a poor ranking compared to the passive strategy because of its greediness (see [12] for a discussion of this phenomenon). However, when we increase d, the complexity of the ranking function class increases and the active algorithm begins to outperform the passive algorithm by a significant amount. In fact, when $d = 41$ for the active algorithm, the passive algorithm needs more than twice the amount of pairwise comparisons to match the error rate of the active algorithm. When $d = 45$, the passive algorithm needs more than 3 times as many pairwise comparisons.

A Appendix

A.1 Proof of Lemma 1.

Proof. First note that without loss of generality, we can take $||w|| \leq 1$ because it is only the direction that matters on the finite set of objects (i.e. the size of the margin is irrelevant). To simplify notation, let $x_i = \phi(\theta_i)$ for $i = 1, \ldots, n$. Observe that $||x_i - r|| - ||x_j - r|| < 0 \iff ||r||^{-1}(x_i^T x_i - x_j^T x_j) - 2||r||^{-1} r^T (x_i - x_j) < 0$, so by making $||r||$ sufficiently large, we can make the contribution of $||r||^{-1}(x_i^T x_i - x_j^T x_j)$ as small as we'd like. This means we are trying to find a $w = -||r||^{-1} r^T$ such that $2w^T(x_i - x_j) < 0$ for all for all pairs (i, j) with $\theta_i \prec \theta_j$ with $||w|| \leq 1$. But this is equivalent to the objective function of RankSVM of [2] so if such a w exists, then the induced ranking is also in $\Sigma_{\Theta,\mathcal{F}}$. This proves $\Sigma_{\Theta,\mathcal{G}} \subset \Sigma_{\Theta,\mathcal{F}}$. □

A.2 Proof of Lemma 2.

Proof. For each $\theta_i \in \Theta$ let $x_i = \phi(\theta_i)$ and $\kappa(\theta_i, \theta_j) = x_i^T x_j$. Suppose we knew the true orientation of the pairwise comparisons for all $\binom{n}{2}$ pairs of objects and the ranking is in $\Sigma_{\Theta,\mathcal{F}}$ where $\mathcal{F} = \{f(x) = ||x - r||\,,\, r \in \mathbb{R}^d\}$. Observe that for each $f \in \mathcal{F}$, $f(x_i) - f(x_j) = ||x_i - r|| - ||x_j - r|| = x_i^T x_i - x_j^T x_j - 2r^T(x_i - x_j)$. For any $r \in \mathbb{R}^d$ satisfying $f(x_i) - f(x_j) < 0$ for all pairs (i, j) with $\theta_i \prec \theta_j$, we may assume without loss of generality that $r \in \text{span}\{(x_i - x_j),\, 1 \leq i < j \leq n\}$. In general, $\text{span}\{(x_i - x_j),\, 1 \leq i < j \leq n\} \subset \text{span}\{x_i,\, 1 \leq i \leq n\}$ so we define

$$B = \Big\{\beta \in \mathbb{R}^n : \sum_{k=1}^{n} \beta_k x_k \in \text{ span}\{(x_i - x_j),\, 1 \leq i < j \leq n\}\Big\}.$$

This allows us to write $r = \sum_{k=1}^n \beta_k x_k$ for some $\beta \in B$ and $\mathcal{F} = \{f(x) = \kappa(x,x) + \sum_{k=1}^n \beta_k \kappa(x_k, x),\, \beta \in B\}$. Define $\mathcal{H} := \{f(x) = \kappa(x,x) + \sum_{k=1}^n \alpha_k \kappa(x_k, x)\,,\, \alpha \in \mathbb{R}^n\}$. Because, in general, $B \subset \mathbb{R}^n$, it follows that $\mathcal{F} \subset \mathcal{H}$ and $\Sigma_{\Theta,\mathcal{F}} \subset \Sigma_{\Theta,\mathcal{H}}$. However, we next show that, perhaps surprisingly, $\Sigma_{\Theta,\mathcal{F}} = \Sigma_{\Theta,\mathcal{H}}$. In fact, we show that for all $h \in \mathcal{H}$, there exists an $f \in \mathcal{F}$ such that $f(x_i) < f(x_j) \iff h(x_i) < h(x_j)$ for all $i \neq j \in \{1,\dots,n\}$.

Observing that B is a subspace of $\mathbb{R}^n$, define $B_\perp$ to be the orthogonal complement of B such that $\mathbb{R}^n = B \cup B_\perp$. This implies that for any $\beta \in B_\perp$, we have $\sum_{k=1}^n \beta_k x_k^T \big(x_i - x_j\big) = 0$ for all $i \neq j \in \{1,\dots,n\}$. Fix an $h \in \mathcal{H}$ such that $h(x_i) < h(x_j)$ for every pair (i,j) with $\theta_i \prec \theta_j$ (recall $\mathcal{F} \subset \mathcal{H}$ so such an h always exists). Let α be the vector in $\mathbb{R}^n$ associated with the particular h and let $\alpha = \alpha_B + \alpha_{B_\perp}$ be a decomposition of α such that $\alpha_B \in B$ and $\alpha_{B_\perp} \in B_\perp$. Then

$$\begin{aligned}
h(x_i) - h(x_j) &= \kappa(x_i,x_i) - \kappa(x_j,x_j) + \sum_{k=1}^n \alpha_k \big(\kappa(x_k,x_i) - \kappa(x_k,x_j)\big) \\
&= \kappa(x_i,x_i) - \kappa(x_j,x_j) + \sum_{k=1}^n \alpha_k\, x_k^T \big(x_i - x_j\big) \\
&= \kappa(x_i,x_i) - \kappa(x_j,x_j) + \sum_{k=1}^n \alpha_{B,k}\, x_k^T \big(x_i - x_j\big) + \sum_{k=1}^n \alpha_{B_\perp,k}\, x_k^T \big(x_i - x_j\big) \\
&= \kappa(x_i,x_i) - \kappa(x_j,x_j) + \sum_{k=1}^n \alpha_{B,k}\, x_k^T \big(x_i - x_j\big) \\
&= f(x_i) - f(x_j)
\end{aligned}$$

for the $f \in \mathcal{F}$ with $f(x) = \kappa(x,x) + \sum_{k=1}^n \alpha_{B,k} \kappa(x_k, x)$ since, by definition, $\alpha_B \in B$. This shows that $\Sigma_{\Theta,\mathcal{H}} \subset \Sigma_{\Theta,\mathcal{F}}$, which completes the proof that $\Sigma_{\Theta,\mathcal{H}} = \Sigma_{\Theta,\mathcal{F}}$.

The second statement of the lemma follows immediately from the first statement and Lemma 1. □

A.3 A note on infinite dimensional kernels

While finite dimensional kernels offer flexibility (for example, the polynomial kernel in the empirical results) it is not uncommon to employ infinite dimensional kernels in practice, such as the popular RBF kernel $\kappa(x_i,x_j) = \exp\{-\frac{||x_i - x_j||^2}{2\sigma^2}\}$ for $X = \{x_1,\dots,x_n\} \subset \mathbb{R}^d$ and $\sigma > 0$. In this case, and any other kernel that satisfies $\kappa(x,x) = 1$ for all $x \in \mathbb{R}^d$, we can express the ranking function class as $\mathcal{F} = \{f(x) = \sum_{k=1}^n \alpha_k \kappa(x_k, x)\,,\, \alpha \in \mathbb{R}^n\}$. For the remainder of this discussion we will assume $\kappa(x,x) = 1$, which is standard property of many common infinite dimensional kernels. If $\mathbf{x} = (x_1,\dots,x_n)^T$ then for every $f \in \mathcal{F}$ there exists an $\alpha \in \mathbb{R}^d$ (and visa versa) such that $f(\mathbf{x}) = K\alpha$ where $K_{i,j} = \kappa(x_i,x_j)$ for all $i,j \in \{1,\dots,n\}$. Thus, in practice the function class is linear in n-dimensional space and not the infinite dimensional space implied by the kernel. Also note that, with the exception of trivial cases, K is full rank. Therefore for *any* $\mathbf{z} := f(\mathbf{x}) \in \mathbb{R}^n$ we can find an $\alpha \in \mathbb{R}^n$ such that $\mathbf{z} = K\alpha$, which implies $|\Sigma_{X,\mathcal{F}}| = n!$. This means that to discover a ranking in $\Sigma_{X,\mathcal{F}}$, we must request at least $\Omega(n \log n)$ queries from the oracle, on average. While at first glance infinite dimensional kernels seem hopeless, we show next that regularization may drastically reduce the number of rankings while maintaining a rich space of ranking functions.

If $K_{i,\cdot}$ denotes the ith row of the kernel matrix K, then specifying a particular ranking in $\Sigma_{X,\mathcal{F}}$ is equivalent to finding an $\alpha \in \mathbb{R}^n$ such that $(K_{i,\cdot} - K_{j,\cdot})\alpha \leq 0$ for all pairs (i,j) with $x_i \prec x_j$ in the ranking. That is, each query is associated with the hyperplane $(K_{i,\cdot} - K_{j,\cdot})$ and these $\binom{n}{2}$ hyperplanes partition $\mathbb{R}^n$ into $n!$ n-cells, each corresponding to a unique ranking. Notice that all the n-cells of $\Sigma_{X,\mathcal{F}}$ are unbounded. This implies that if we define $\mathcal{F}' = \{f(x) = \sum_{k=1}^n \alpha_k \kappa(x_k,x)\,,\, \alpha \in \mathbb{R}^n \text{ s.t. } |\sum_{k=1}^n \big(\kappa(x_k,x_i) - \kappa(x_k,x_j)\big)\alpha_k| \geq 1,\, \forall (i,j)\}$, then $\Sigma_{X,\mathcal{F}'} = \Sigma_{X,\mathcal{F}}$ because $||\alpha||$ can be arbitrarily large. In fact, the n-cells of $\Sigma_{X,\mathcal{F}'}$ are simply shrunken, or eroded, versions of the n-cells of $\Sigma_{X,\mathcal{F}}$. Now consider a slightly different function class defined as

$$\mathcal{F}(s) = \{f(x) = \sum_{k=1}^{n} \kappa(x_k, x)\alpha_k,\ \alpha \in \mathbb{R}^n \text{ s.t.}$$

$$\alpha^T K \alpha \leq s \text{ and } |\sum_{k=1}^{n} \big(\kappa(x_k, x_i) - \kappa(x_k, x_j)\big)\alpha_k| \geq 1,\ \forall (i,j)\}$$

for some $s > 0$. We observe that $\Sigma_{X,\mathcal{F}(s)}$ contains all the rankings of $\Sigma_{X,\mathcal{F}'}$ that correspond to n-cells that have non-zero intersection with the ellipse $\{\alpha \in \mathbb{R}^d : \alpha^T K \alpha \leq s\}$. Clearly, as $s \to \infty$, $\Sigma_{X,\mathcal{F}(s)} \to \Sigma_{X,\mathcal{F}}$ and as $s \to 0$, $\Sigma_{X,\mathcal{F}(s)}$ reduces to an empty set. We can also interpret $\alpha^T K \alpha \leq s$ as a means of controlling the smoothness of the function f with a smaller value of s corresponding to a smoother function. Finally, if finding an α with $\sum_{k=1}^{n} \big(\kappa(x_k, x_i) - \kappa(x_k, x_j)\big)\alpha_k \leq -1$ for all $x_i \prec x_j$ is viewed as a classification problem, one can show a one-to-one correspondence between s and the minimum acceptable margin of the linear separator $\sum_{k=1}^{n} \alpha_k \phi(x_k)$ over the points $\big(\phi(x_i) - \phi(x_j)\big)$ in the ϕ space where $\kappa(x, x') = \phi(x)^T \phi(x')$ [2, 13]. Using this last interpretation, we can use VC theory to show that $\log |\Sigma_{X,\mathcal{F}(s)}| = O(s \log n)$ by noticing that the margin is at least $2s^{-1/2}$ and $\max_{i,j} (\phi(x_i) - \phi(x_j))^T (\phi(x_i) - \phi(x_j)) \leq 2$ [14]. While we do not claim that our proposed algorithm will achieve a query complexity of $O(s \log n)$ for all data inputs and choices of s, we conjecture that the typical performance is not too much worse than this estimate. Exploring the connection between s and the query complexity is the topic of future work. It should also be noted that this regularization technique can also be applied to finite dimensional kernels.

References

[1] K. Jamieson and R. Nowak. Active ranking using pairwise comparisons. *Neural Information Processing Systems (NIPS)*, *http://homepages.cae.wisc.edu/~jamieson/activeRanking_extended.pdf*, 2011.

[2] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Neural Information Processing Systems*, pages 115–132, 1999.

[3] R.J. Arens. Learning to rank documents with support vector machines via active learning. 2009.

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

[5] N. Ailon, R. Begleiter, and E. Ezra. A new active learning scheme with applications to learning to rank from pairwise preferences. *Arxiv preprint arXiv:1110.2136*, 2011.

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

[7] Style Guidelines by Category. [http://www.bjcp.org/stylecenter.php]. Beer Judge Certification Program, 2011.

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

[9] Aural Sonar. [http://idl.ee.washington.edu/SimilarityLearning/Applications/Datasets/]. University of Washington Information Design Lab, 2011.

[10] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776, March 2009.

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

[12] Sanjoy Dasgupta. Two faces of active learning. *Theor. Comput. Sci.*, 412:1767–1781, April 2011.

[13] R. Tibshirani T. Hastie and J. Friedman. *The elements of statistical learning*. Springer, 2009.

[14] V.N. Vapnik. Statistical learning theory. 1998.