Visualizing Differences in Web Search Algorithms using the Expected Weighted Hoeffding Distance

Mingxuan Sun Guy Lebanon Kevyn Collins-Thompson
Georgia Institute of Technology Microsoft Research

ABSTRACT

We introduce a new dissimilarity function for ranked lists, the *expected weighted Hoeffding distance*, that has several advantages over current dissimilarity measures for ranked search results. First, it is easily customized for users who pay varying degrees of attention to websites at different ranks. Second, unlike existing measures such as generalized Kendall's tau, it is based on a true metric, preserving meaningful embeddings when visualization techniques like multi-dimensional scaling are applied. Third, our measure can effectively handle partial or missing rank information while retaining a probabilistic interpretation. Finally, the measure can be made computationally tractable and we give a highly efficient algorithm for computing it. We then apply our new metric with multi-dimensional scaling to visualize and explore relationships between the result sets from different search engines, showing how the weighted Hoeffding distance can distinguish important differences in search engine behavior that are not apparent with other rank-distance metrics. Such visualizations are highly effective at summarizing and analyzing insights on which search engines to use, what search strategies users can employ, and how search results evolve over time. We demonstrate our techniques using a collection of popular search engines, a representative set of queries, and frequently used query manipulation methods.

1. INTRODUCTION

Search engines return ranked lists of documents or websites in response to a query, with the precise forms of the ranked lists depending on the internal mechanisms of the engines. We consider the problems of comparing and visualizing the similarity relationships between different search algorithms. The term *search algorithm* is an intentionally vague term corresponding to a mechanism for producing ranked lists of websites in response to queries. We focus on the following three interpretations of search algorithms: (a) different search engines, (b) a single search engine subjected to different query manipulation techniques by the user, and (c) a single engine queried across different internal states.

A visualization of relationships among type-(a) search algorithms may be useful as it reveals which search engines users should or should not use. For example, two search engines that output very similar ranked lists may provide redundant information to the user. On the other hand, two search engines that output very dissimilar lists are worth examining more closely. We emphasize that we do not consider here the issue of search quality or relevance. The techniques developed in this paper allow users to understand the similarity relationships among search algorithms. Evaluating retrieval quality is a separate, well-studied problem that is beyond the scope of this paper.

Visualizing relationships among type-(b) search algorithms is useful as it indicates which query manipulation techniques are worth using. For example, a query `times square` may be manipulated by the user (prior to entering it in the search box) to `times AND square`. Such manipulations may result in different ranked lists output by the search engine. Understanding the similarities between these ranked lists may provide the user with insights into which manipulations are redundant and which are worth exploring further.

Understanding the relationships among type-(c) search algorithms is useful from the point of view of design and modification of search engines. Search engines are complex programs containing many tunable parameters, each one influencing the formation of the ranked list in a different way. For example, commercial search engines frequently update the internal index which may result in different ranked lists output in different dates (in response to the same query). It is important for both the users and the search engineers to understand how much the ranked lists differ across consecutive days and how much this difference changes as internal parameters are modified.

There are several techniques for visualizing complex data such as search algorithms. One of the most popular techniques is multidimensional scaling (MDS) [5], which transforms complex high dimensional data $s_1, \ldots, s_m$ into 2-D vectors $z_1, \ldots, z_m$ that are easily visualized by displaying them on a 2-D scatter plot. Assuming that a suitable dissimilarity measure between the high dimensional data $\overline{\rho}(s_i, s_j)$ has been identified, MDS computes the 2-D embedding of the high dimensional data $s_i \mapsto z_i \in \mathbb{R}^2$, $i = 1, \ldots, m$ that minimizes the distortion

$$R(z_1, \ldots, z_m) = \sum_{i<j} \left(\overline{\rho}(s_i, s_j) - \|z_i - z_j\|\right)^2. \qquad (1)$$

In other words, the coordinates $z_1, \ldots, z_m$ in $\mathbb{R}^2$ corresponding to $s_1, \ldots, s_m$ are selected to minimize the distortion

$$(z_1, \ldots, z_m) = \underset{z'_1, \ldots, z'_m}{\arg\min}\, R(z'_1, \ldots, z'_m). \qquad (2)$$

Variations of MDS with slightly different distortions (1) and objective functions (2) may be found in [5].

Our contributions in this paper are (1) to develop a suitable dissimilarity function for search algorithms $\overline{\rho}(s_i, s_j)$ and study its properties, (2) to examine the use of MDS in the context of visualizing search algorithms of types (a), (b), and (c), and (3) to validate it using synthetic and web data.

2. RELATED WORK

The analysis of search engine outputs is a large area of research within the IR community [2]. Most approaches evaluate rankings output by search engines against human judgement or some other ground truth [12, 16, 14, 28, 4, 27]. In this area, Carterette [6] recently enumerated the limitations of Kendall's tau for comparing system performance. He proposed an alternative method for computing rank distance that accounts for dependency between items, which is computed as the solution to a minimization problem. The resulting measure, however, has problems with interpretation and is highly dependent on the number of systems and queries being analyzed, and is used instead as input to a p-value difference estimator for comparing system performance. In general, increased industry and research interest in measuring dissimilarity between different search engines has lead to a variety of comparison tools[1].

The key for comparing and visualizing ranked lists output by search engines is to define an appropriate distance metric. A straightforward way to measure such distance is to compute the overlap of the two lists [20]. Such an approach is problematic, however, as it is invariant with respect to reordering or the ranked lists. Popular distance measures between permutations are Kendall's tau, Spearman's rho, the footrule, Ulam's distance, and Cayley's distance [9]. There are several ways to extend these permutation distances to partially ranked lists output by search engines, including Hausdorff distance [7] and expected distances [1, 22].

Substantial research on the interaction between users and search engines [13, 17] show that users' attention drops quickly from top to bottom ranks[2]. One problem with many proposed dissimilarities, however, is that they do not distinguish between disagreement at top rankings and at the bottom rankings. One exception is [11] who considers stage-wise ranking processes that generalize Kendall's tau. This generalization, however, is not unique as it depends on the order in which the different ranking stages are selected. Rank correlation coefficient such as NDCG [16] adopts inverse logarithm function as the discount rank factor. However it is not symmetric and is intended as an evaluation measure against ground truth, not a comparison measure between ranked lists. Another example is the inverse measure [3] that emphasizes disagreement at top ranks, where the weight function decays linearly with the rank.

While much previous work on visualization of search algorithms is based on document similarity, e.g. [24], there is renewed interest in visualizing and analyzing set-level differences between results from different search systems. Fagin et al. [10] and Bar-Ilan et al. [3] investigate the relationship among engines by examining the pairwise distance matrix but do not make the connection with visualization. Liggett and Buckley [21] used multi-dimensional scaling over ranking dissimilarity to examine search system variations due to the effect of query expansion, where the dissimilarity was based on Spearman's coefficient. More recently, tools like MetaCrystal [26] and the more general ConSet [19] regard search results as a set of items and visualize the common items among different engines. Temporal studies of search engines have been examined by [3] and [15] who compare search engine results across multiple time periods.

3. DISTANCES AND DISSIMILARITIES

As mentioned in Section 1, effective visualization of search algorithms using MDS depends on the quality of the dissimilarity measure $\overline{\rho}$. We describe in this section a new measure based on the expectation of the weighted Hoeffding distance on permutations and examine its properties.

We start by considering several desired properties for $\overline{\rho}(s_i, s_j)$. It should be (i) symmetric, (ii) interpretable with respect to search algorithms retrieving ranked lists of different lengths, (iii) flexible enough to model the increased attention users pay to top ranks over bottom ranks, (iv) computationally efficient, and (v) aggregate information over multiple queries in a meaningful way.

The symmetry property (i) is relatively straightforward. Property (ii) addresses an important and often overlooked issue. How should ranks in a short ranked list be compared with ranks in a long one? How should we count websites that appear only in the longer (or shorter) ranked list? Many previously proposed dissimilarity measures provide ad-hoc answers to these questions which may lower the quality of the MDS embedding. Property (iii) refers to the differences in attention users pay to websites listed in top vs. bottom ranks. The dissimilarity measure should take this into account and provide a dissimilarity similar to that experienced by users, as opposed to a rank-symmetric formula. The efficiency property (iv) can be critical for online use and we address that in Sec. 3.2. Property (v) refers to the fact that $\overline{\rho}(s_i, s_j)$ should aggregate information from multiple queries with each query contributing the "correct" amount to the final dissimilarity $\overline{\rho}(s_i, s_j)$.

Dissimilarity functions examined in previous studies satisfy some but not all of these properties (see Figure 2). In this paper we propose to define $\overline{\rho}$ using an expectation over the weighted Hoeffding distance. The expectation and the properties of the weighted Hoeffding distance provide a clear probabilistic interpretation and ensure that properties (i)-(v) are satisfied.

We start by defining the weighted Hoeffding distance which is a novel distance on permutations $d_w(\pi, \sigma)$. The weight vector w provides the flexibility necessary for satisfying (iii) while the metric property satisfies (i). We then extend it to a dissimilarity $\rho(s_i(q), s_j(q))$ over ranked lists $s_i(q), s_i(j)$ by taking expectations with respect to the sets of permutations $\mathfrak{S}(s_i(q)), \mathfrak{S}(s_j(q))$ consistent with the ranked lists. Above, we consider search algorithms s_i, s_j as functions from queries to ranked lists and $s_i(q)$ represents the ranked list retrieved by s_i in response to the query q.

The function ρ is extended to search algorithms by taking another expectation, this time with respect to queries q sampled from a representative set of queries Q. The expectations ensure that properties (ii) and (v) are satisfied. We derive an efficient closed form for the double expectation that verifies property (iv) in Sec. 3.2 and give a pseudo-code implementation.

Formally, we have

$$\begin{aligned}\overline{\rho}(s_i, s_j) &= \mathsf{E}_{q\sim Q}\{\rho(s_i(q), s_j(q))\} \\ &= \mathsf{E}_{q\sim Q}\ \mathsf{E}_{\pi\sim\mathfrak{S}(s_i(q))}\mathsf{E}_{\sigma\sim\mathfrak{S}(s_j(q))}\{d_w(\pi,\sigma)\} \qquad (3)\end{aligned}$$

where $d_w(\pi, \sigma)$ is a distance between permutations π, σ defined in Section 3.1, $\mathsf{E}_{\pi\sim\mathfrak{S}(s_i(q))}\mathsf{E}_{\sigma\sim\mathfrak{S}(s_j(q))}$ is the expectation with respect to permutations π, σ that are sampled from the sets of all permutations consistent with the ranked lists output by the two algorithms $s_i(q), s_j(q)$ (respectively), and $\mathsf{E}_{q\sim Q}$ is an expectation with respect to all queries sampled from a representative set of queries Q. In the absence of any evidence to the contrary, we assume a uniform distribution over the set of queries Q and over the sets of permutations consistent with $s_i(q), s_j(q)$. However, in other cases non-uniform distributions may be considered in order to emphasize certain queries or introduce apriori information regarding the likely ranks of certain websites.

We proceed with a description of $d_w(\pi, \sigma)$ in Section 3.1

[1] e.g.`www.bing-vs-google.com`, `www.searchrater.com`

[2] A popular discount function is the logarithm function [8] but other discount functions have been proposed as well.

and then follow up in Section 3.2 with additional details regarding the expectations in (3) and how to compute them.

3.1 Weighted Hoeffding Distance

The weighted Hoeffding distance is a distance between permutations, here considered as permutations over the n indexed websites in the internet[3]. The fact that n is extremely large should not bother us at this point as we will derive closed form expressions eliminating any online complexity terms depending on n. For simplicity we refer to the websites using the integers $\{1, \dots, n\}$.

A permutation over n websites π is a bijection from $\{1, \dots, n\}$ to itself mapping websites to ranks. That is $\pi(6)$ is the rank given to website 6 and $\pi^{-1}(2)$ is the website that is assigned second rank. A permutation is thus a full ordering over the entire web and we denote the set of all such permutations by $\mathfrak{S}_n$. We will represent a permutation by a sorted list of websites from most preferred to least, separated by vertical bars i.e. $\pi^{-1}(1)|\cdots|\pi^{-1}(n)$; for example, for $n = 5$ one permutation ranking item 3 as first and 2 as last is $3|5|1|4|2$.

Our proposed distance $d_w(\pi, \sigma)$ is a variation of the earth movers distance[4] [25] on permutations. It may also be regarded as a weighted version of the Hoeffding distance [22]. It is best described as the minimum amount of work needed to transform the permutation π to σ. Work, in this case, is the total amount of work needed to bring each item from its rank in π to its rank in σ i.e., the r-item is transported from rank $k = \pi(r)$ to $l = \sigma(r)$ (for all $r = 1, \dots, n$) requiring $w_k + \cdots + w_{l-1}$ work (assuming $k < l$) where w_k is the work required to transport an item from rank k to $k + 1$. For example, the distance $d(1|2|3, 2|1|3)$ is $w_1 + w_1$ due to the sequence of moves $1|2|3 \to |1, 2|3 \to 2|1|3$. Another example is $d(1|2|3, 3|1|2) = w_1 + w_2 + w_2 + w_1$ due to the sequence of moves $1|2|3 \to |1, 2|3 \to |1|2, 3 \to |1, 3|2 \to 3|1|2$.

Formally, the distance may be written as

$$d_w(\pi, \sigma) = \sum_{r=1}^{n} d'_w(\pi(r), \sigma(r)) \quad \text{where} \tag{4}$$

$$d'_w(u, v) = \begin{cases} \sum_{t=u}^{v-1} w_t & \text{if } u < v \\ d'_w(v, u) & \text{if } u > v \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

The weight vector $w = (w_1, \dots, w_{n-1})$ allows differentiating the work associated with moving items across top and bottom ranks. A monotonic decreasing weight vector, e.g., $w_t = t^{-q}$, $t = 1, \dots, n - 1$, $q \geq 0$ correctly captures the fact that disagreements in top ranks should matter more than disagreements in bottom ranks [13, 17, 23]. The exponent q is the corresponding rate of decay. A linear or slower rate $0 \leq q \leq 1$ may be appropriate for persistent search engine users who are not very deterred by low-ranking websites. Choosing $q \to 0$ retrieves a weighting mathematically similar to the log function weighting that is used in NDCG [16] to emphasize top ranks. A quadratic or cubic decay $q = 2, 3$ may be appropriate for users who do not pay substantial attention to bottom ranks. The weight may be modified to $w_t = \max(t^{-q} - \epsilon, 0)$, $\epsilon > 0$ to capture the fact that many users simply do not look at results beyond a certain rank. While it is possible to select an intuitive value of q, it is more desirable to select one that agrees with user studies. An MDS embedding of permutations using d_w appears in Figure 1 (see Section 4 for more details).

	(i)	(ii)	(iii)	(iv)	(v)
Kendall/Spear [9]	✓			✓	
Fligner Kendall [11]			✓		
E Kendall top k [10]	✓	✓		✓	✓
E Spearman [21]	✓	✓		✓	✓
InverseMeasure [3]	✓		✓	✓	✓
NDCG [16]			✓	✓	✓
E Weighted Hoeffding	✓	✓	✓	✓	✓

Figure 2: Summary of how different dissimilarities satisfy properties (i)-(v) in Section 3.

Proposition 1. *Assuming w is a positive vector, the weighted Hoeffding distance* (4) *is a metric.*

Proof. Non-negativity $d_w(\pi, \sigma) \geq 0$ and symmetry $d_w(\pi, \sigma) = d_w(\sigma, \pi)$ are trivial to show. Similarly it is easy to see that $d_w(\pi, \sigma) = 0$ iff $\pi = \sigma$. The triangle inequality holds as

$$\begin{aligned} d_w(\pi, \sigma) + d_w(\sigma, \varphi) &= \sum_{r=1}^{n} d'_w(\pi(r), \sigma(r)) + d'_w(\sigma(r), \varphi(r)) \\ &\geq \sum_{r=1}^{n} d'_w(\pi(r), \varphi(r)) = d_w(\pi, \varphi) \qquad (6) \end{aligned}$$

where the inequality (6) holds due to the positivity of w. □

The weighted Hoeffding distance has several nice properties that make it more appropriate for our purposes than other permutation measures. First, it allows customization to different users who pay varying degrees of attention to websites in different ranks (typically higher attention is paid to higher ranks). Standard permutation distances such as Kendall's tau, Spearman's rho, the footrule, Ulam's distance and Cayley's distance treat all ranks uniformly [9]. Second, it is a true metric in contrast to the generalized Kendall's tau [11]. Third, its clear interpretation allows explicit specification of the weight vector based on user studies. Finally, it is computationally tractable to compute the weighted Hoeffding distance as well as its expectation over partially ranked lists corresponding to ρ in (8). Figure 2 summarizes the advantages of our distance over other dissimilarities.

3.2 Ranked Lists and Expectations

In this section we describe how the ranked lists retrieved by search algorithms relate to permutations and provide more information regarding the expectations in (3) and their computation. A ranked list output by a search algorithm forms an ordered list $\langle i_1, \dots, i_k \rangle$ of a subset of the websites $\{i_1, \dots, i_k\} \subset \{1, \dots, n\}$. Different search strategies may result in lists of different sizes but in general k is much smaller than n. In addition to the notation $\langle i_1, \dots, i_k \rangle$ we also denote it using the bar notation as

$$\begin{aligned} &i_1|i_2|\cdots|i_k|i_{k+1}, \dots, i_n \quad \text{where} \\ &\qquad \{i_{k+1}, \dots, i_n\} = \{1, \dots, n\} \setminus \{i_1, i_2, \cdots, i_k\} \qquad (7) \end{aligned}$$

indicating that the unranked items $\{1, \dots, n\} \setminus \{i_1, \dots, i_k\}$ are ranked after the k items. Partial rankings (7) are not

[3] There are several ways to define the number of indexed websites in the internet. In any case, this number is very large and is growing continuously. We avoid its dynamic nature and consider it as a fixed number.

[4] The earth mover distance between two non-negative valued function is the minimum amount of work needed to transform one to the other, when the functions are viewed as representing spatial distributions of earth or rubble.

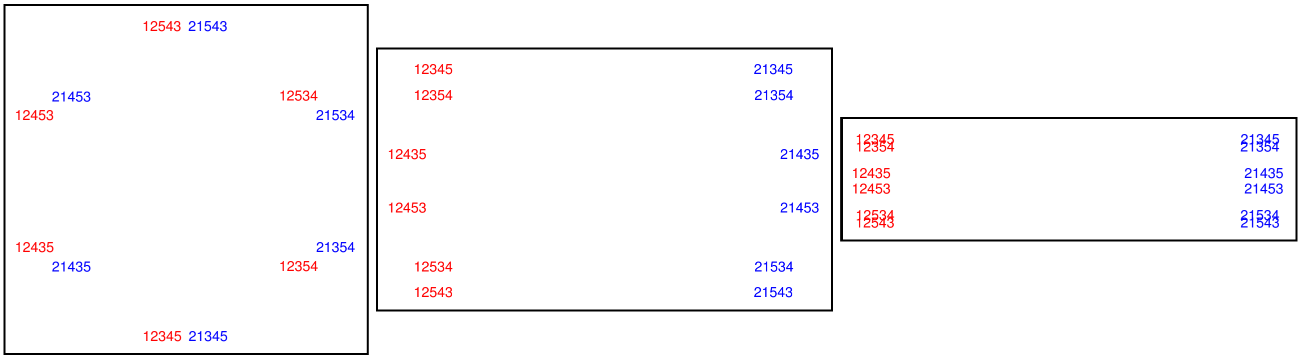

Figure 1: MDS embedding of permutations over $n = 5$ websites. The embeddings were computed using the weighted Hoeffding distance with uniform weight function $w_t = 1$ (left), linear weight function $w_t = 1/t$ (middle) and quadratic weight function $w_t = 1/t^2$ (right). The permutations starting with 1 and 2 (colored in red) and the permutations starting with 2 and 1 (colored in blue) become more spatially disparate as the rate of weight decay increases. This represents the increased importance assigned to agreement in top ranks as we move from uniform to linear and quadratic decay.

identical to permutations as there is no known preference among the unranked items $\{1, \dots, n\} \setminus \{i_1, \dots, i_k\}$. We therefore omit vertical lines between these items and list them separated by commas i.e., $3|2|1, 4$ is equivalent to the ranked list $\langle 3, 2\rangle$ which prefers 3 over 2 and ranks 1 and 4 last without clear preference between them.

It is natural to identify a ranked list $\langle i_1, \dots, i_k\rangle$ as a full permutation of the web that is unknown except for the fact that it agrees with the website ranking in $\langle i_1, \dots, i_k\rangle$. Denoting the set of permutations whose website ordering does not contradict $\langle i_1, \dots, i_k\rangle$ as $\mathfrak{S}(\langle i_1, \dots, i_k\rangle)$, we have that $\langle i_1, \dots, i_k\rangle$ corresponds to a random draw from $\mathfrak{S}(\langle i_1, \dots, i_k\rangle)$. Assuming lack of additional knowledge, we consider all permutations in $\mathfrak{S}(\langle i_1, \dots, i_k\rangle)$ as equally likely resulting in

$$\rho(\langle i_1, \dots, i_k\rangle, \langle j_1, \dots, j_l\rangle) \overset{\text{def}}{=} \mathsf{E}_{\pi\sim\mathfrak{S}(\langle i_1,\dots,i_k\rangle),\sigma\sim\mathfrak{S}(\langle j_1,\dots,j_l\rangle)} d(\pi, \sigma)$$
$$= \frac{1}{(n-k)!(n-l)!} \sum_{\pi\in\mathfrak{S}(\langle i_1,\dots,i_k\rangle)} \sum_{\sigma\in\mathfrak{S}(\langle j_1,\dots,j_l\rangle)} d(\pi, \sigma). \quad (8)$$

For example, consider the case of $n = 5$ with two search strategies returning the following ranked lists $\langle 3, 1, 4\rangle = 3|1|4|2, 5$ and $\langle 1, 5\rangle = 1|5|2, 3, 4$. The expected distance is

$$\rho(3|1|4|2, 5, 1|5|2, 3, 4) = \frac{1}{2\cdot 6}(d(3|1|4|2|5, 1|5|2|3|4)$$
$$+ d(3|1|4|5|2, 1|5|2|3|4) + \cdots + d(3|1|4|2|5, 1|5|4|3|2)$$
$$+ d(3|1|4|5|2, 1|5|4|3|2)). \quad (9)$$

Expression (8) provides a natural mechanism to incorporate information from partially ranked lists. It is difficult to compare directly two ranked lists $\langle i_1, \dots, i_k\rangle$, $\langle j_1, \dots, j_l\rangle$ of different sizes. However, the permutations in $\mathfrak{S}(\langle i_1, \dots, i_k\rangle)$ and $\mathfrak{S}(\langle j_1, \dots, j_k\rangle)$ are directly comparable to each other as they are permutations over the same set of websites. The expectation (8) aggregates information over such directly comparable events to provide a single interpretable and coherent dissimilarity measure. Figure 3 displays the MDS embedding for partial rankings using the expected distance ρ in (8) for several different weight vectors (see Section 4 for more details).

The expectation defining ρ in (8) appears to require insurmountable computation as it includes summations over $(n-k)!(n-l)!$ elements with n being the size of the web. However, using techniques similar to the ones developed in [22] we are able to derive the following closed form.

Proposition 2. *The following closed form applies to the expected distance over the weighted Hoeffding distance* (4)*.*

$$\rho(\langle i_1, \dots, i_k\rangle, \langle j_1, \dots, j_l\rangle) = \sum_{r=1}^{n} \bar{d}(r) \quad \textit{where} \quad (10)$$

$$\bar{d}(r) = \begin{cases} d'_w(u, v) & r \in A \cap B \\ \frac{1}{n-l}\sum_{t=l+1}^{n} d'_w(t, u) & r \in A \cap B^c \\ \frac{1}{n-k}\sum_{t=k+1}^{n} d'_w(t, v) & r \in A^c \cap B \\ \frac{1}{n-k}\frac{1}{n-l}\sum_{t=k+1}^{n}\sum_{s=l+1}^{n} d'_w(t, s) & \textit{otherwise} \end{cases}.$$

Above, $A = \{i_1, \dots, i_k\}$, $B = \{j_1, \dots, j_l\}$, and $u \in \{1, \dots, k\}$, $v \in \{1, \dots, l\}$ are the respective ranks of r in $\{i_1, \dots, i_k\}$ and $\{j_1, \dots, j_l\}$ (it they exist).

Proof. A careful examination of (4) reveals that it may be written in matrix notation:

$$d(\pi, \sigma) = \text{tr}(A_\pi \triangle A_\sigma^T) \quad (11)$$

where tr is the trace operator, $\triangle$ is the $n \times n$ distance matrix with elements $\triangle_{uv} = d'_w(u, v)$, and A_π, A_σ are permutation matrices corresponding to the permutations π and σ i.e., $[A_\pi]_{uv} = 1$ iff $\pi(u) = v$. Using equation (11), we have

$$\rho(\langle i_1, \dots, i_k\rangle, \langle j_1, \dots, j_l\rangle)$$
$$= \frac{\sum_{\pi\in\mathfrak{S}(\langle i_1,\dots,i_k\rangle)}\sum_{\sigma\in\mathfrak{S}(\langle j_1,\dots,j_k\rangle)} \text{tr}(A_\pi \triangle A_\sigma^T)}{(n-k)!(n-l)!}$$
$$= \text{tr}(\hat{M}_{\langle i\rangle} \triangle (\hat{M}_{\langle j\rangle})^T) \quad (12)$$

where

$$\hat{M}_{\langle i\rangle} = \frac{\sum_{\pi\in\mathfrak{S}(\langle i_1,\dots,i_k\rangle)} A_\pi}{(n-k)!}, \quad \hat{M}_{\langle j\rangle} = \frac{\sum_{\sigma\in\mathfrak{S}(\langle j_1,\dots,j_l\rangle)} A_\sigma}{(n-l)!}. \quad (13)$$

Note that the marginal matrices $\hat{M}_{\langle i\rangle}$, $\hat{M}_{\langle j\rangle}$ have a probabilistic interpretation as their u, v entries represent the probability that item u is ranked at v. Combining (12) with Lemma 1 below completes the proof. □

Lemma 1. *Let $\hat{M}$ be the marginal matrix for a top-k ranked list $\langle i_1, \dots, i_k\rangle$ with a total of n items as in* (13)*. If $r \in \{i_1, \dots, i_k\}$ and $r = i_s$ for some $s = 1, \dots, k$, then $\hat{M}_{rj} = \delta_{js}$ where $\delta_{ab} = 1$ if $a = b$ and 0 otherwise. If $r \notin \{i_1, \dots, i_k\}$ then $\hat{M}_{rj} = 0$ for $j = 1, \dots, k$ and $1/(n-k)$ otherwise.*

Proof. For a top-k ranking $\langle i_1, \dots, i_k\rangle$ out of n items, the size of the set $\mathfrak{S}(\langle i_1, \dots, i_k\rangle)$ is $(n-k)!$. Each of the

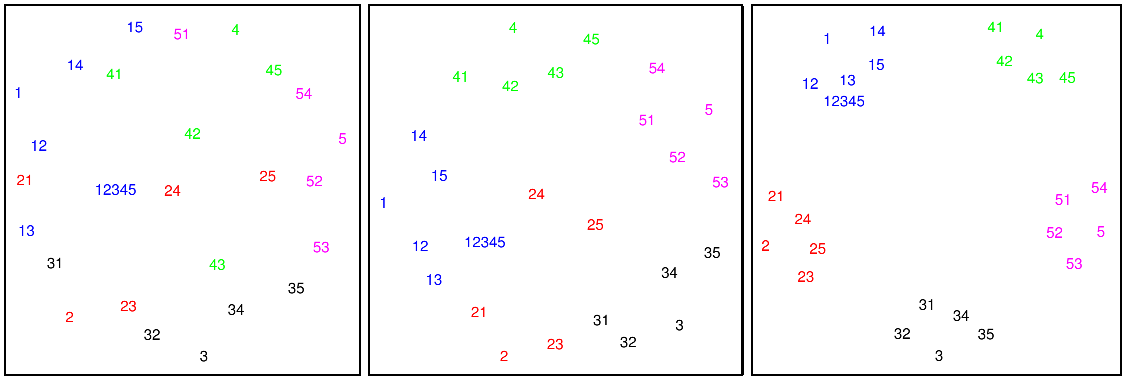

Figure 3: MDS embedding of ranked lists of varying lengths (k varies) over a total of $n = 5$ websites. The embeddings were computed using the expected weighted Hoeffding's distance (8) with uniform weight function $w_t = 1$ (left), linear weight function $w_t = 1/t$ (middle) and quadratic weight function $w_t = 1/t^2$ (right). We observe the same phenomenon that we saw in Figure 1 for permutations. The expected distance (8) separates ranked lists agreeing in their top rankings (denoted by different colors) better as the weights decay faster.

permutations compatible with it has exactly the same top-k ranks. If $r \in \{i_1, \dots, i_k\}$ and $r = i_s$ for some $s = 1, \dots, k$ then the number of permutations compatible with $\langle i_1, \dots, i_k \rangle$ that assign rank s to the item is $(n-k)!$. Similarly, the number of consistent permutations assigning rank other than s to the item is 0. As a result we have $\hat{M}_{rs} = \frac{(n-k)!}{(n-k)!} = 1$ and $\hat{M}_{rj} = 0$ for $j \neq s$. If $r \notin \{i_1, \dots, i_k\}$, the number of permutations consistent with the ranked list that assign rank $j \in \{k+1, \dots, n\}$ to the item is $(n-k-1)!$. Similarly, the number of permutations that assign rank $j \in \{1, \dots, k\}$ to the item is 0. As a result $\hat{M}_{rj} = 0$ for $j = 1, \dots, k$, and $\hat{M}_{rj} = \frac{(n-k-1)!}{(n-k)!} = \frac{1}{n-k}$ for $j = k+1, \dots, n$. □

The expected distance (8) may be computed very efficiently, assuming that some combinatorial numbers are precomputed offline. Bounding k, l by a certain number $k, l \leq m \ll n$ we have that the online complexity is $O(k+l)$ and the offline complexity is $O(n+m^2)$. The next proposition, makes this precise. A pseudo-code description of the distance computation algorithm is given as Algorithm 3.1.

Proposition 3. *Let $\langle i_1, \dots, i_k \rangle$ and $\langle j_1, \dots, j_l \rangle$ be top-k and top-l ranks on a total n items with $k, l \leq m \ll n$. Assuming that $d_w(u, v)$ is computable in constant time and space complexity (as is the case for many polynomial decaying weight vectors w) the online space and time complexity is $O(k+l)$. The offline space complexity is $O(m^2)$ and the offline time complexity is $O(n+m^2)$.*

Proof. From equation 10, the offline pre-computation requires computing $D_{m\times m}$, $D^{E1}_{m\times m}$ and $D^{E2}_{m\times m}$, where $D_{uv} = d'_w(u, v)$, $D^{E1}_{kv} = \frac{1}{n-k}\sum_{t=k+1}^{n} d'_w(t, v)$, and $D^{E2}_{kl} = \frac{1}{n-k}\frac{1}{n-l}\sum_{t=k+1}^{n}\sum_{s=l+1}^{n} d'_w(t, s)$. The space complexity for computing these matrices is $O(m^2)$. The time complexity to compute $D_{m\times m}$ is $O(m^2)$. Exploiting features of cumulative sums and the matrix $D_{m\times m}$ it can be shown that computing $D^{E1}_{m\times m}$ requires $O(n+m^2)$ time. Similarly, computing $D^{E2}_{m\times m}$ requires $O(n+m^2)$ time. As a result, the total offline complexity is $O(m^2)$ space and $O(n+m^2)$ time. Given the three precomputed matrices, computing the expected distance for two partially ranked lists $\langle i_1, \dots, i_k \rangle$ and $\langle j_1, \dots, j_l \rangle$ requires $O(k+l)$ time and space. The reasons are that given two list, the time to identify overlapping items from two lists of size k and l is $O(k+l)$ and that for items ranked by at least one engine, we need to use the look-up table no more than $k+l$ times and another extra look-up for items never ranked in both. □

4. SIMULATION STUDY

To evaluate the proposed framework we conducted two sets of analyses. The first analysis uses synthetic data to examine properties of the embedding and compare it to alternative methods under controlled settings. The second set of experiments includes real world search engine data. Its goal is to demonstrate the benefit and applicability of the framework in the context of a real world visualization problem. We explore the first set of experiments in this section and the second set in Section 5.

We start by examining the embedding of permutations over $n = 5$ websites. A small number is chosen intentionally for illustrative purposes. We consider two sets of permutations. The first set contains all permutations ranking item 1 first and item 2 second. The second set contains all permutations ranking item 2 first and item 1 second. Figure 1 displays the MDS embedding of these two sets of permutations based on the weighted Hoeffding's distance with constant weights $w_t = 1$ (left), linear weight $w_t = t^{-1}$ (middle) and quadratic weight $w_t = t^{-2}$ (right). The first set of permutations are colored red and the second set blue.

The uniform weight MDS embedding does not pay particular attention to differing websites in the top ranks and so the red and blue permutations are interspersed together. This is also the embedding obtained by using the Kendall's tau distance as in [9, 10, 18]. Moving to linearly decaying and quadratic decaying weights increases the separation between these two groups dramatically. The differences in websites occupying top ranks are emphasized while differences in websites occupying bottom ranks are de-emphasized. This demonstrates the ineffectiveness of using Kendall's tau distance or uniform weight Hoeffding distance in the context of search engines. The precise form of the weight - linear, quadratic, or higher decay rate depends on the degree to which a user pays more attention to higher ranked websites than to lower ranked websites.

The second simulated experiment is similar to the first, but it contains partially ranked lists as opposed to permutations. We form five groups - each one containing partially ranked lists ranking a particular website at the top. Figure 3 displays the MDS embedding of these five sets of

```
Off-line:
1. Specify $n$, the number of total items and $m$ the list length bound.
2. Precompute matrices $D_{m\times m}$, $D^{E1}_{m\times m}$ and $D^{E2}_{m\times m}$ (Section 3).
On-line:
3. Call Expected-Weighted-Hoeffding($\pi$, $\sigma$) for lists $\pi$ and $\sigma$

Expected-Weighted-Hoeffding($\pi, \sigma$)
  $k_1 \leftarrow size(\pi)$
  $k_2 \leftarrow size(\sigma)$
  $[\pi_{mark}, \sigma_{mark}] =$ Mark-Rank($\pi, \sigma$);
  $sum \leftarrow 0$;
  for $i \leftarrow 1$ to $k_1$
  do
      if $\pi_{mark}[i] > 0$
        then $sum \leftarrow sum + D[i, \pi_{mark}[i]]$
        else
              $sum \leftarrow sum + D^{E1}[k_2, i]$

  $count \leftarrow 0$
  for $i \leftarrow 1$ to $k_2$
  do
      if $\sigma_{mark}[i] = 0$
        then $sum \leftarrow sum + D^{E1}[k_1, i]$
              $count \leftarrow count + 1$

  $sum \leftarrow sum + (n - k_1 - count) \cdot D^{E2}[k_1, k_2]$;
  return $sum$;

Mark-Rank($a, b$)
  $k_1 = size(a)$
  $k_2 = size(b)$
  $a_{mark} = zeros(1 \ldots k_1), b_{mark} = zeros(1 \ldots k_2)$
  for $i \leftarrow 1$ to $k_1$
  do for $j \leftarrow 1$ to $k_2$
      do if $a[i] = b[j]$
            then $a_{mark}[i] = j$
                  $b_{mark}[j] = i$

  return $[a_{mark}, b_{mark}]$
```

Algorithm 3.1: Algorithm to compute expected weighted Hoeffding distance between two ranked lists π and σ. The online complexity of the above algorithm is $O(k_1 k_2)$. A slightly more complex algorithm can achieve online complexity $O(k_1 + k_2)$ as described in Proposition 3.

permutations based on the expected weighted Hoeffding distance $\rho(\langle i_1, \ldots, i_k\rangle, \langle j_1, \ldots, j_l\rangle)$ in (8) using constant weights $w_t = 1$ (left), linear weights $w_t = t^{-1}$ (middle) and quadratic weights $w_t = t^{-2}$ (right). Ranked lists in each of the different groups are displayed in different colors.

We observe a similar conclusion with the expected distance over partially ranked lists as we did with the distances over permutations. The five groups are relatively interspersed for uniform weights and get increasingly separated as the rate of weight decay increases. This represents the fact that as the decay rate increases, disagreements in top ranks are emphasized over disagreement at bottom ranks.

We also conducted some comparisons between the weighted Hoeffding distance and alternative distance measures. Table 1 shows how one recently proposed measure, the inverse measure [3], lacks discriminative power, assigning the same dissimilarity to very different ranked lists. Kendall's tau and the other distances proposed in [9, 10] lack the ability to distinguish disagreement in top ranks and bottom ranks. In particular, Kendall's tau is identical to our weighted Hoeffding distance with uniform weights (see Figures 1-3 for a demonstration of its inadequacy). NDCG [16] and other precision recall measures rely on comparing a ranked list to a ground truth of relevant and not-relevant websites. As such they are not symmetric and are not appropriate for computing MDS embedding based on the matrix of pairwise distances.

d to 1|2|3|4|5	InverseMeasure	$w_t = t^{-1}$	$w_t = t^{-2}$
2	0.8374	0.6500	0.7539
3	0.8374	0.7786	0.8589
4	0.8374	0.8357	0.8901
5	0.8374	0.8571	0.8988
1|3	0.2481	0.3048	0.2049
1|4	0.2481	0.3810	0.2464
1|5	0.2481	0.4095	0.2581

Table 1: A comparison of the inverse measure [3] with the weighted Hoeffding distance indicates that the inverse measure lacks discriminative power as it assigns the same dissimilarity to very different ranked lists ($n = 5$).

d to 1|2|3|4|5	$n = 5$	$n = 10$	$n = 10^3$	$n = 10^5$	$n = 10^7$
1|2|3|5|4|	0.0117	0.0176	0.0670	0.0698	0.0699
2|1|3|4|5|	0.7464	0.6755	0.6660	0.6683	0.6683
1|4|2|	0.1268	0.1362	0.1950	0.1980	0.1981
1|	0.1064	0.1592	0.2656	0.2692	0.2692
2|1	0.7726	0.7283	0.7515	0.7543	0.7543
5|	0.9395	0.9280	0.9820	0.9851	0.9852
5|4|3|2|1|	1.0000	0.9025	0.8727	0.8748	0.8748

Table 2: A comparison of weighted Hoeffding distance with cubic weight decay $w_t = t^{-3}$ reveals that increasing n beyond a certain size does not alter the distances between partially ranked lists. This indicates lack of sensitivity to the precise value of n as well as computational speedup resulting from replacing n by $n' \ll n$.

Table 2 shows a comparison of weighted Hoeffding distances with $w_t = t^{-3}$ for different sizes of the web n. It reveals that increasing n beyond a certain size does not alter the distances between partially ranked lists. This indicates a lack of sensitivity to the precise value of n as well as computational speedup resulting from replacing n by $n' \ll n$.

5. SEARCH ENGINE EXPERIMENTS

We discuss in this section three experiments conducted on real world search engine data. In the first experiment we visualize the similarities and differences between nine different search engines: `altavista.com`, `alltheweb.com`, `ask.com`, `google.com`, `lycos.com`, `live.com`, `yahoo.com`, `aol.com`, and `dogpile.com`. We collected 50 popular queries online in each of six different categories: company names, questions[5], sports, tourism[6], university names, and celebrity names. These queries form a representative sample of queries Q within each category over which we average the expected distance ρ according to (3). Figure 4 shows several queries for each one of the topic categories. We visualize search result sets within each of the query categories in order to examine whether the discovered similarity patterns are specific to a query category, or are generalizable across many different query types.

5.1 Search Engines Similarities

Figure 5 displays the MDS embedding of each of the nine engines for the six query categories, based on the expected weighted Hoeffding distance (3) with linear weight decay. The ρ quantity was averaged over the 50 representative queries from that category. Each search engine is represented as a circle whose center is the 2-D coordinates obtained from the MDS embedding.

[5] `queries from http://answers.yahoo.com`

[6] `queries from http://en.wikipedia.org/wiki/Tourism`

Categories	Queries
Tourism	Times Square, Sydney Opera House, Eiffel Tower, Niagara Falls, Disneyland, British Museum, Giza Pyramids
Celebrity Names	Michael Bolton, Michael Jackson, Jackie Chan, Harrison Ford, Halle Berry, Whoopi Goldberg, Robert Zemeckis
Sports	Football, Acrobatics, Karate, Pole Vault, Butterfly Stroke, Scuba Diving,Table Tennis, Beach Volleyball, Marathon
University Names	Georgia Institute of Technology, University of Florida, Virginia Tech, University of California Berkeley
Company	Goldman Sachs, Facebook, Honda, Cisco Systems, Nordstrom, CarMax, Wallmart, American Express, Microsoft
Questions	How are flying buttresses constructed, Does toothpaste expire, How are winners selected for the Nobel Prize
Temporal Queries	AIG Bonuses, G20 major economies, Timothy Geitner, Immigration Policy, NCAA Tournament Schedule

Figure 4: Selected queries from each of the 6 query categories, and from the set used for examining temporal variations.

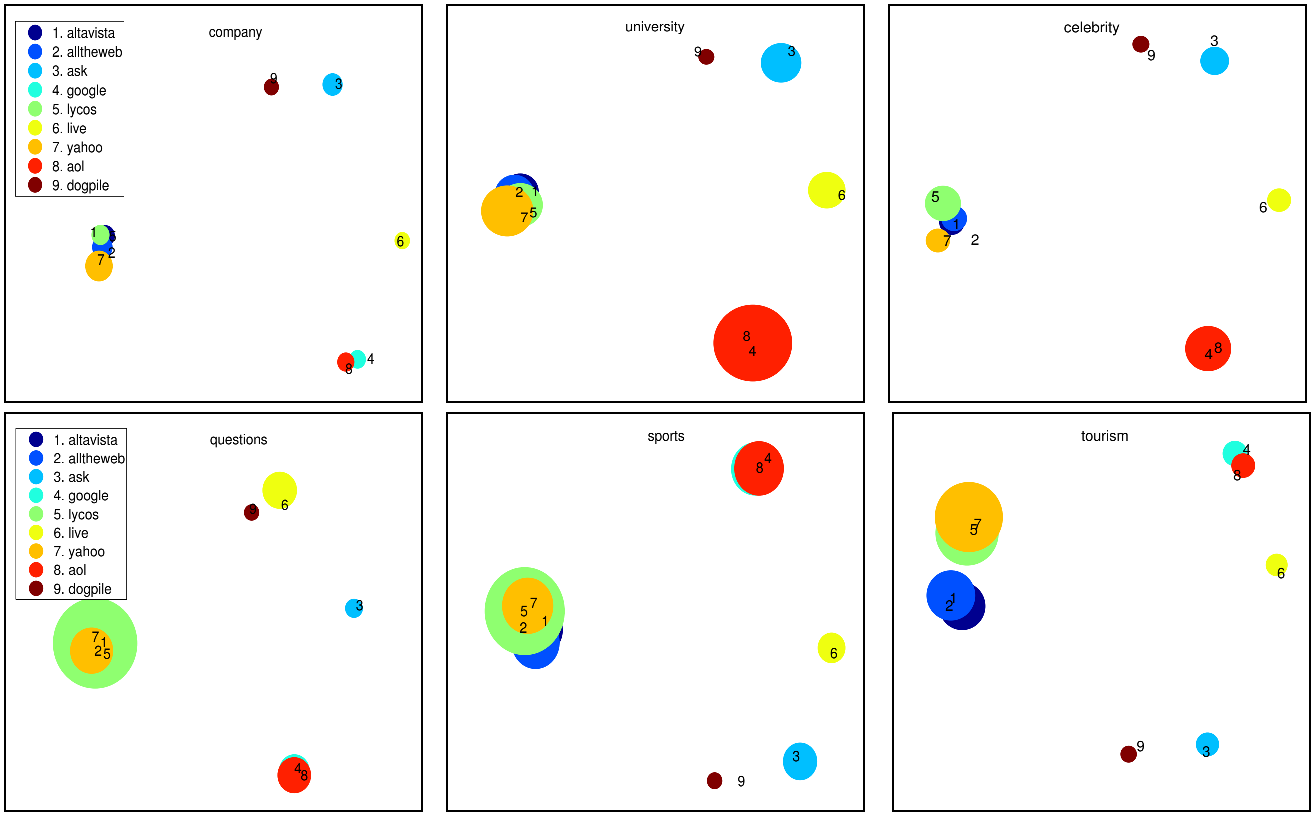

Figure 5: MDS embedding of search engine results over 6 sets of representative queries: company names, university names, celebrity names, questions, sports, and tourism. The MDS was based on the expected weighted Hoeffding distance with linear weighting $w_t = t^{-1}$ over the top 100 sites. Circle sizes indicate position variance with respect to within category queries.

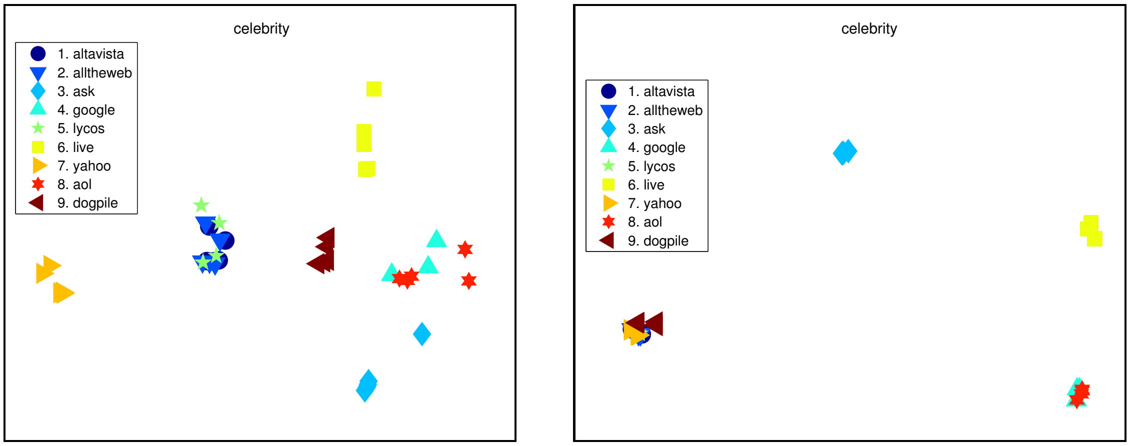

Figure 6: MDS embedding of search engine results over the query category celebrity with different query manipulations. The MDS was computed based on the expected distance (3) with (8) corresponding to the weighted Hoeffding distance with quadratic decaying weights (left panel) and Kendall top k distance [10] (right panel). Each marker represents a combination of one of the 9 search engines and one of the 5 query manipulation techniques. By comparison, the embedding of the Kendall top-k distance (right) lacks discriminative power and results in a loss of information.

The radii of the circles in Figure 5 were scaled proportionally to the *positional stability* of the search engine. More precisely, we scaled the radius of the circle corresponding to the i-th search engine proportionally to its distance variance over the 50 queries

$$\text{stability}(i) \stackrel{\text{def}}{=} \sum_{j:j\neq i} \mathsf{Var}_{q\sim Q}\{\rho(s_i(q), s_j(q))\}. \tag{14}$$

Scaling the circles according to (14) provides a visual indication of how much will the position change if one or more queries are deleted or added to Q. This can also be interpreted as the degree of uncertainty regarding the precise location of the search engines due to the averaging over Q.

Examining Figure 5 reveals several interesting facts. To begin with there are five distinct clusters. The first and largest one contains the engines altavista, alltheweb, lycos, and yahoo (indicated by the numeric codes 1,2,5,7). These four search engines are clustered together very tightly in all 6 query categories. The second cluster is composed of google and AOL (numeric codes 4, 8) who also appear in very close proximity across all 6 query categories. The remaining three clusters contain individual engines: live, dogpile, and ask.

The clusters in the embedding do in fact mirror the technology relationships that have evolved in the search engine industry. FAST, the company behind alltheweb, bought Lycos and was subsequently bought by Overture who also bought Altavista[7]. Overture was subsequently bought by the fourth member of the cluster, Yahoo. All four search engines in the first cluster have close proximity in the embedding and yet are dissimilar from the remaining competitors. The second cluster, for Google and AOL, reflects the fact that AOL now relies heavily on Google's web search technology, leading to extremely similar ranked lists.

The remaining engines are quite distinct. Dogpile is a meta-search engine which incorporates the input of the other major search engines. We see that dogpile's results are roughly equidistant from both Yahoo and Google clusters for all query categories. Figure 5 also shows that dogpile is more similar to the two remaining engines - Live and Ask. Apparently, dogpile emphasizes pages highly-ranked by Live and Ask in its meta search more than Google and AOL and more than Yahoo, Lycos, Altavista, and alltheweb.

5.2 Query Manipulations

In the second experiment we used search engine data to examine the sensitivity of the search engines to four commonly used query manipulation techniques. Assuming that the queries contained several words $w_1 w_2 \cdots w_l$ with $l > 1$, the query manipulation techniques that we considered were

$$\begin{array}{llll}
(a) & w_1 w_2 \cdots w_l & \Rightarrow & w_1 w_2 \cdots w_l \\
(b) & w_1 w_2 \cdots w_l & \Rightarrow & w_1 + w_2 + \cdots + w_l \\
(c) & w_1 w_2 \cdots w_l & \Rightarrow & \text{``}w_1 w_2 \cdots w_l\text{''} \\
(d) & w_1 w_2 \cdots w_l & \Rightarrow & w_1 \text{ and } w_2 \text{ and } \cdots \text{ and } w_l \\
(e) & w_1 w_2 \cdots w_l & \Rightarrow & w_1 \text{ or } w_2 \text{ or } \cdots \text{ or } w_l
\end{array}$$

with the first technique (a) being the identity i.e. no query manipulation. The embeddings of queries in the query category celebrity are displayed in Figure 6. The left panel displays the MDS embedding based on our expected weighted Hoeffding distance with quadratic decaying weight. As a comparison, the right panel shows the MDS based on Kendall's top k distance as described by Fagin et al. [10]. Each marker in the figure represents the MDS embedding of a particular engine using a particular query manipulation technique which brings the total number of markers to $9 \cdot 5 = 45$.

Engine/Distance	(b) +	(c) " "	(d) and	(e) or
Ask	0.5308	**0.6903**	0.6424	0.6447
Live	0.5625	**0.5639**	0.5006	0.5374
Google	0.3553	0.4117	**0.5584**	0.5500
Yahoo	0.4281	0.4647	0.5777	**0.5918**

Figure 7: The expected distance of different query manipulations from the original query for different search engines.

Comparing the left and right panels shows that visualizing using Kendall's top k distance [10] lacks discriminative power. The points in the right panel fall almost on top of each other limiting their use for visualization purposes. In contrast, the points in the left panel (weighted Hoeffding distance) differentiate among not only different engines but also different types of query manipulations.

In particular, it shows that most search engines produce two different clusters of results corresponding to two sets of query manipulation techniques: transformations {(a),(b), (c)} in one cluster and transformations {(d),(e)} in the other cluster. Live and ask form an exception to that rule forming clusters {(a),(d)}, {(b),(c)}, {(e)} (live) and {(a),(b),(d),(e)}, {(c)} (ask). Figure 7 shows the query manipulations that produce ranked lists most distinct from the original query: (c) for ask and live, (d) for google, and (e) for yahoo.

5.3 Temporal Variation

In the third experiment, we visualize search result sets created by replicating 7 out of the 9 search engines over 7 consecutive days resulting in $7 \cdot 7 = 49$ search result sets. The search engines were queried on a daily basis during 3/25/2009 - 3/31/2009 and the returned results were embedded in 2-D for visualization. In contrast to the previous two experiments, we used a separate query category which was specifically aimed at capturing time sensitive matters. For example, we ignored tourism queries such as Eiffel Tower due to their time insensitive nature and instead used queries such as Timothy Geitner or AIG bonuses which dominated the news in March 2009. See Fig. 4 for more examples.

The embedded rankings are displayed in Figure 8 (top). The embedding reveals that the yahoo cluster (yahoo, altavista, alltheweb, and lycos) shows a high degree of temporal variability, and in particular a sharp spatial shift on the third day from the bottom region to the top left region. This could be interpreted either as a change in the index, reflecting the dynamic nature of the Web, or an internal change in the retrieval algorithms underlying the engines. The other engines were more stable as their ranked lists changed very little with the temporal variation. Note that as the queries were time sensitive this should not be interpreted as a measure of robustness, but rather as a stability measure for the internal index and ranking mechanisms. Interestingly, Vaughan [27] also reports that Altavista shows temporal jumps with Google being more stable over a set of queries in 2004.

Figure 8 (bottom) shows the expected distance between the yahoo search results across the seven consecutive days and a reference point (2nd and 6th day). As expected, for each one of the two plots, the deviation to the reference date increases monotonically with the temporal difference. The slope of the curve represents the degree of temporal change $\Delta(s_t, s_{t+\tau})$ between yahoo at time t and at time $t + \tau$ as a

[7] `http://google.blogspace.com/archives/000845.html`

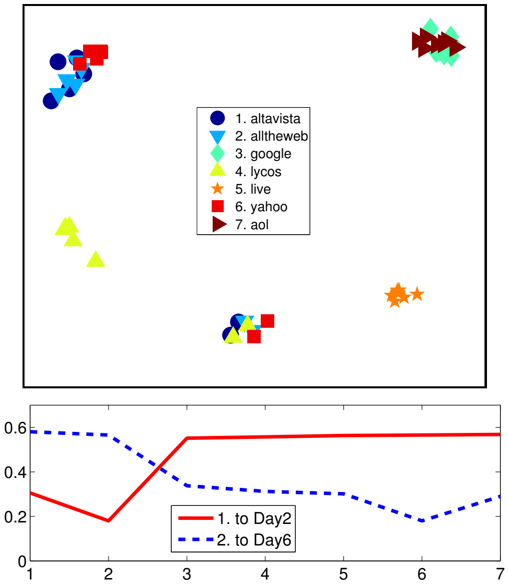

Figure 8: Top: MDS embedding of search engine results over seven days for a set of queries based on temporal events. The MDS embedding was based on the expected Hoeffding distance with linear weighting $w_t = 1/t$ over the top 50 sites. Bottom: The dissimilarity of Yahoo results over seven days with respective to a reference day for a set of queries based on temporal events.

function of τ with respect to the reference point t.

5.4 Validation of the MDS Embedding

A central assumption of this study is that MDS embeddings can give a faithful representation of the distances in the original higher-dimensional space. Thus, we now provide a validation of the MDS embedding as a visualization tool, diagnosing whether MDS is providing a reasonable embedding and which MDS variant should be preferable.

The most common tool for validating the MDS embedding is Shepard's plot (Figure 9, left) which displays a scatter plot contrasting the original dissimilarities (on the x axis) and the corresponding distances after the embedding (on the y axis). Points on the diagonal represent zero distortion and a curve that deviates substantially from the diagonal represents substantial distortion. The Shepard's plot in Figure 9 (left) corresponds to the metric MDS using the standard stress criterion as described in (2). The plot displays low distortion with a tendency to undervalue dissimilarities in the range $[0, 0.5]$ and to overvalue dissimilarities in the range $[0.5, 0.9]$. Such a systematic discrepancy between the way small and large distances are captured is undesirable.

An alternative is non-metric MDS which achieves an embedding by transforming the original dissimilarities into alternative quantities called disparities using a monotonic increasing mapping which are then approximated by the embedding distances [5]. Doing so preserves the relative ordering of the original dissimilarities and thus (assuming the embedding distances approximate well the disparities) accurately represent the spatial relationship between the points. Figure 9 (middle) displays the Shepard's plot for the same data embedded using non-metric stress MDS with the disparities displayed as a red line. Figure 9 (right) displays the embedded distances as a function of the disparities revealing no systematic tendency to overestimate or underestimate as did the metric MDS. Thus, despite the fact that its numeric distortion is higher, the non-metric MDS is a viable alternative to the metric MDS, and is what was used to generate the figures in this paper.

6. DISCUSSION

In this paper we present a framework for visualizing relationships between search algorithms. The framework starts by deriving an expected distance based on the earth mover's distance on permutations and then extends it to partially ranked lists by taking expectations over all permutations consistent with the ranked lists. The expected distance ρ is then averaged over representative queries $q \in Q$ to obtain a dissimilarity measure for use in multidimensional scaling embedding. The expected distance has several nice properties including being computationally efficient, customizable through a selection of the weight vector w, and interpretable.

We explore the validity of the framework using a simulation study which indicates that the weighted Hoeffding distance is more appropriate than Kendall's tau and more discriminative than the inverse measure. It is also more appropriate for MDS embedding than non-symmetric precision-recall measures such as NDCG. We also demonstrate the robustness of the proposed distance with respect to the choice of n and its efficient computation with complexity that is linear in the sizes of the ranked lists (assuming some quantities are precomputed offline). Experiments on search engine data reveal several interesting clusters which are corroborated by examining recent news stories about the web search industry. We demonstrate how to visualize the positional stability by scaling the MDS markers proportionally to the total distance variance and how to visualize sensitivity of search engines to popular query manipulation techniques. We also use the visualization framework to examine how the search results vary over consecutive days.

Search engines use complex proprietary algorithms containing many parameters that are automatically tuned based on human provided ground truth information. As a result, it is difficult for search engineers to have a detailed understanding of how precisely their search engine works. For external users the problem is even worse as they are not privy to the internal algorithmic details. Our framework provides visual assistance in understanding the relationship between a search algorithm's ranked results and its dependence on internal and external parameters. Such visualization may lead to better designed search engines as the engineers improve their understanding of how search engines depend on the internal parameters. It may also improve the search experience as users understand better the relationship between different engines and their dependency on query manipulation techniques and external parameters such as time.

7. REFERENCES

[1] M. Alvo and P. Cabilio. Rank correlation methods for missing data. *The Canadian Journal of Statistics*, 23(4):345–358, 1995.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] J. Bar-Ilan, K. Keenoy, E. Yaari, and M. Levene. User rankings of search engine results. *Journal of American Society for Information Science and Technology*, 58(9):1254–1266, 2007.

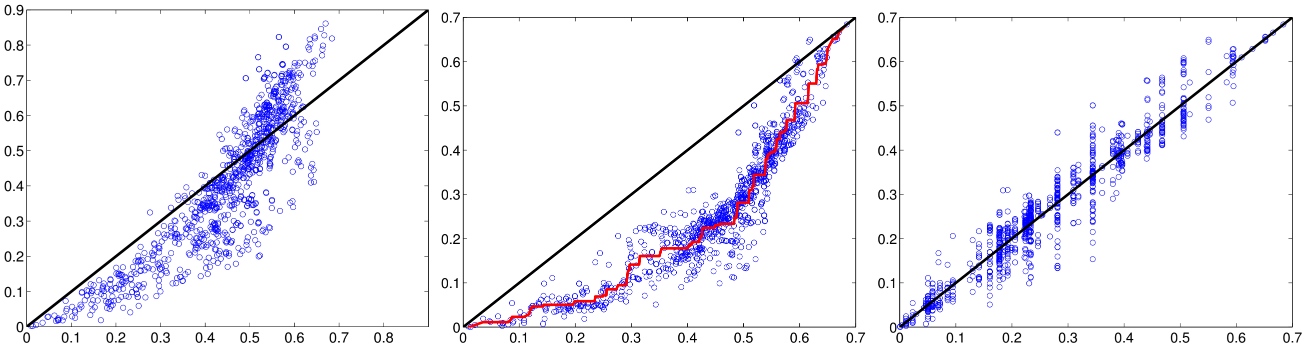

Figure 9: Shepard plots (embedding distances as a function of the original dissimilarities) for 2D MDS embeddings corresponding to the weighted hoeffding distance of search engine results over query category celebrity with different query manipulations. The metric stress MDS [5] (left panel) produces an embedding that has a lower overall distortion than the non-metric stress MDS [5] (middle panel). The non-metric stress MDS, however, achieves an embedding by transforming the original dissimilarities into alternative quantities called disparities using a monotonic increasing mapping (red line in middle panel). Doing so preserves the relative ordering of the distances thus accurately reflecting the spatial relationships between the points. The right panel displays the embedded distances as a function of the disparities. The displayed spread is symmetric and with little outliers making the non-metric MDS a viable alternative to the metric MDS. See [5] for more details.

[4] M. M. S. Beg. A subjective measure of web search quality. *Information Sciences*, 169(3-4):365–381, 2005.

[5] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling*. Springer, 2nd edition, 2005.

[6] B. Carterette. On rank correlation and the distance between rankings. In *Proc. of the 32nd ACM SIGIR Conference*, 2009.

[7] D. E. Critchlow. *Metric Methods for Analyzing Partially Ranked Data*. Lecture Notes in Statistics, volume 34, Springer, 1985.

[8] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 2009.

[9] P. Diaconis. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, 1988.

[10] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. In *Proc. of ACM SODA*, 2003.

[11] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society B*, 43:359–369, 1986.

[12] M. Gordon and P. Pathak. Finding information on the world wide web: the retrieval effectiveness of search engines. *Information processing and management*, 35(2):141–180, 1999.

[13] L.A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proc. of the ACM-SIGIR conference*, pages 478–479, 2004.

[14] D. Hawking, N. Craswell, P. Bailey, and K. Griffihs. Measuring search engine quality. *Information Retrieval*, 4(1):33–59, 2001.

[15] B. J. Jansen, A. Spink, and J. Pedersen. A temporal comparison of altavista web searching. *Journal of the American Society for Information Science and Technology*, 56(6):559–570, 2005.

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

[17] T. Joachims, L.A. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of the ACM-SIGIR conference*, pages 154–161, 2005.

[18] P. Kidwell, G. Lebanon, and W. S. Cleveland. Visualizing incomplete and partially ranked data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1356–1363, 2008.

[19] B. Kim, B. Lee, and J. Seo. Visualizing set concordance with permutation matrices and fan diagrams. In *Interacting with Computers*, 2007.

[20] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proc. of the 18th ACM SIGIR conference*, 1995.

[21] W. Liggett and C. Buckley. Query expansion seen through return order of relevant documents. Technical report, NIST, 2001.

[22] J. I. Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

[23] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems*, 27, 2008.

[24] M. E. Rorvig. Images of similarity: A visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets. *JASIS*, 50(8):639–651, 1999.

[25] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2), 2000.

[26] A. Spoerri. Metacrystal: A visual interface for meta searching. In *Proceedings of ACM CHI*, 2004.

[27] L. Vaughan. New measurements for search engine evaluation proposed and tested. *Information processing and management*, 40(4):677–691, 2004.

[28] E. Yilmaz, J. A. Aslam, and S. Robertson. A new rank correlation coefficient for information retrieval. In *Proc. of the 31st ACM SIGIR conference*, 2008.