

Rank Aggregation for Similar Items

D. Sculley *

Tufts University, Medford, MA, USA

dsculley@cs.tufts.edu

Abstract

The problem of combining the ranked preferences of many experts has gained renewed importance in data mining, and information retrieval applications. Effective rank aggregation becomes difficult in real-world situations in which the rankings are noisy, incomplete, or even disjoint. We address these difficulties by extending several standard methods of rank aggregation to consider similarity between items in the various ranked lists, in addition to their rankings. We establish evaluation criteria for these algorithms by extending previous definitions of distance between ranked lists to include the role of similarity between items. Experimental results show that this approach improves the performance of several standard rank aggregation methods, particularly when used with noisy, incomplete rankings.

1 Introduction.

Rank aggregation, the process of combining ordered preferences of many experts, is a useful tool for modern data mining. Rank aggregation can be thought of as the unsupervised analog to regression, in which the goal is to find an aggregate ranking that minimizes the distance to each of the given ranked lists. Rank aggregation has also been proposed as an effective method for nearest-neighbor ranking of categorical data [4], and gives a robust approach to the problem of combining the opinions of diverse experts in classification tasks.

We address the problem of aggregating noisy, incomplete ranked lists through the novel addition of *similarity* information. The intuition is that similar items should be ranked similarly, given an appropriate similarity measure for the data. This information can help resolve noisy rankings, and increase the effectiveness of aggregating incomplete or even disjoint rankings.

As a motivating example, consider the following toy problem, consisting of ranked lists from two experts:

Expert 1: A, B, C Expert 2: C', D, E

In these lists, the items **C** and **C'** are highly similar, but are not exact matches for one another. Traditionally,

the two lists are considered disjoint, and standard methods of rank aggregation will interleave the rankings in one of the following ways:

Aggregation 1: A, C', B, D, C, E

Aggregation 2: C', A, D, B, E, C

Each of these aggregations is unsatisfactory, as highly similar items **C** and **C'** are given divergent rankings.

The methods proposed in this paper, described in Section 4, capitalize on the additional information provided by a defined similarity measure. Consider the following alternative, based on rank aggregation with similarity:

Aggregation 3: A, B, C', C, D, E

Aggregation 3 agrees with our intuition that **C'** should be ranked behind both **A** and **B**, while **C** should be ranked ahead of both **D** and **E**. In Section 3, we propose an extension to the Kendall Tau metric that formalizes this benefit, and give experimental results supporting these methods in Section 5.

2 Definitions

Before moving further, we formalize the notions of *rankings* and *similarity*.

2.1 Ranked lists. We start with a universe U of items, each with a unique identifier $i \in U$. A ranked list r of items $i_1 \dots i_n$ drawn from a universe U of items is an ordered subset $S \subseteq U$ with each $i \in S$, such that $r = [i_1 \geq i_2 \geq \dots \geq i_n]$. (Here, we draw heavily on the notation of [3].) For each item $i \in S$, $r(i)$ shows the ranking of item i in the ranked list r . Note that the optimal ranking is 1, rankings are always positive, and higher rank shows lower preference in the list. The number of items in r is given by $|r|$, and we assume that the items in U are each assigned a unique identifier. We refer to the set of items in r by S_r , although we will occasionally refer to items in a list r with the shorthand notation $i \in r$, which will be clear by context. The notation r^n refers to the item i in r such that $r(i) = n$.

*Work performed at Yahoo!, Inc., in Spring of 2006.

There are several special cases of ranked lists to consider. A *complete ranking* is a list r that contains a ranking for every item $i \in U$. A *partial ranking* is a list r that contains rankings for only a subset of items $S \subset U$. A *Top- k* list is a list r with $|r| = k$, and all other items $i \notin r$ are assumed to be ranked below every item in r by the given expert [5].

A *projection* operator is used in computing distances between partial and complete rankings [3]. When T is a subset of U , and r is a (possibly partial) ranking of items $S_r \subseteq U$, then the projection $r|_T$ contains only items $S \cap T$ while keeping the relative orderings from r invariant. The notation $r|_{r_2}$ denotes the projection of r onto the set of objects in S_{r_2} .

2.2 Similarity functions. A pairwise *similarity* function $s(\cdot, \cdot)$ between items in U must satisfy the following requirements. First, the function must be non-negative for all pairwise comparisons: $\forall i, j \in U : s(i, j) \geq 0$. Second, the function must be symmetric: $\forall i, j \in U : s(i, j) = s(j, i)$. Third, the similarity between an item and itself must be greater than zero: $\forall i \in U : s(i, i) > 0$. Finally, the maximum similarity between an item and any other item may not be greater than the similarity between the item and itself: $\forall i, j \in U : s(i, i) \geq s(i, j)$. It will often be the case that a *normalized* similarity function is useful. In these cases, $\forall i \in U : s(i, i) = 1$. One method of normalizing similarity functions is: $s_{norm}(i, j) = \frac{s(i, j)}{\sqrt{s(i, i)s(j, j)}}$.

These flexible similarity requirements allow the use of a variety of similarity functions, including kernels from pattern recognition and machine learning [14].

Our distance measures between partial ranked lists using similarity information will require a *lambda-similarity projection*. This projection is defined from a list r of items $S \subseteq U$ onto a set $T \subseteq U$, and uses a given similarity function $s(\cdot, \cdot)$. The projection yields a new list $r_{\lambda|T}$ that contains a rank-invariant ordering of all the items $j \in T$ for which there exists an item $i \in S$ such that $s(i, j) > \lambda$. Here, λ represents a minimum similarity threshold that i and j must meet for the similarity to be considered meaningful.

Notation in hand, we now formalize our task by establishing a method of measuring disagreement among ranked lists, using similarity information.

3 Measuring Disagreement Among Rankings

On an informal level, we desire that any method of rank aggregation should seek to produce an aggregate ranking with minimal total disagreement among the input lists. In this section, we extend the *Kendall-Tau distance* to include the case where we have a meaningful

similarity measure between items in the lists. This will serve to define formally the problem of rank aggregation given similarity information. (Note that an analogous extension of the Spearman Footrule distance for ranked lists with similarity information appears in [13].)

3.1 Distance Measures with Similarity Previous methods of measuring distances between ranked lists, such as the Kendall Tau distance, have a drawback for measuring distance between ranked lists of similar items when those lists are incomplete. In this case, these methods ignore the rankings of items that do not appear in both lists [3]. This makes sense when we do not have similarity information – ignoring disjoint items may be the best solution. However, when we do have meaningful similarity information for ranked items, ignoring the effect of similar (but not identical) ranked items may cause us to inappropriately prefer one aggregation over another, or to be unable to distinguish between two aggregations. An example of these problems was shown in the introduction. To address these issues, we extend the standard distance measures to take similarity information into account.

Kendall Tau Similarity Distance. We now extend the Kendall tau distance to include item similarity. We will require the specification of a similarity measure $s(\cdot, \cdot)$ on the items in U . Recall that the Kendall tau distance examines the number of pairwise re-orderings between the two lists [7]. Extending this measure raises the possible case that two items i and j in list r_1 may have many similar items $i'_1..i'_n$ and $j'_1..j'_m$ in r_2 , and these similar items may be ordered in arbitrarily complex ways. Thus, we first define an *aggregate similarity position* function, $g(\cdot, \cdot, \cdot)$ and an *aggregate similarity list* r_g to resolve these complexities.

DEFINITION 3.1. *The aggregate similarity position of item i with respect to list r , under similarity function $s(\cdot, \cdot)$, is defined as*

$$g(i, r, s(\cdot, \cdot)) = \frac{\sum_{j \in r} s(i, j)r(j)}{\sum_{j \in r} s(i, j)}$$

An aggregate similarity list r_g is composed from lists r_1, r_2 , and similarity function $s(\cdot, \cdot)$, such that for every element $i \in r_1, r_g(i) = g(i, r_2, s(\cdot, \cdot))$ when $g(i, r_2, s(\cdot, \cdot))$ is defined. When $g(i, r_2, s(\cdot, \cdot))$ is not defined, i is not in r_g . (Furthermore, note that rank values in r_g are not necessarily integers.) Such a list r_g is returned from the function $r_g(r_1, r_2, s(\cdot, \cdot))$.

DEFINITION 3.2. *The Kendall tau similarity distance between two (possibly partial) ranked lists r_1 and r_2 on*

items in U , given a pairwise similarity function $s(\cdot, \cdot)$ on items in U , is defined as

$$K_{sim}(r_1, r_2, s(\cdot, \cdot)) = \frac{1}{2} \{K(r_1, r_g(r_1, r_2, s(\cdot, \cdot))) + K(r_2, r_g(r_2, r_1, s(\cdot, \cdot)))\}$$

Thus, $K_{sim}(r_1, r_2)$ is the average Kendall tau distance between r_1 and its aggregate similarity list drawn from r_2 , and the Kendall tau distance between r_2 and its aggregate similarity list drawn from r_1 .

In this paper, we will assess the total distance between an aggregate ranking σ and a set R of k expert rankings $R = \{r_1 \dots r_k\}$. In this case, define U as the union of all items in the lists in R , and if σ is a complete ranking on U then the *induced distance* is computed via lambda-similarity projections of σ onto the various lists in R : $K_{sim}(\sigma, R) = \frac{1}{k} \sum_{j=1}^k K_{sim}(\sigma_{\lambda|r_j}, r_j)$. The Kendall tau distance may be scaled by the maximum possible disagreement, $\binom{|U|}{2}$, which normalizes these scores to the range $[0,1]$.

The original Kendall tau distance is recovered from the Kendall tau similarity distance by using the uniqueness function $s_0(\cdot, \cdot)$. Note that both $K_{sim}(\cdot, \cdot, s(\cdot, \cdot))$ and $K(\cdot, \cdot)$ are non-metric for partial rankings, as either will compute zero distance between any list and the empty list. The evaluation of the Kendall tau similarity metric requires $O(|r_1||r_2|)$ similarity computations, in addition to the two order $n \log n$ time evaluations of the Kendall tau distance.

Minimizing Distances Selecting a distance measure on ranked lists, our goal in rank aggregation is to find an aggregate list σ that minimizes the total distance from σ to every given list $r \in R$, as computed by the chosen distance measure.

Minimizing the Kendall tau distance results in what is called a *Kemeny optimal ranking*, which has well studied desirable properties, such as the Condorcet criterion which states that the top ranked item in aggregation should be the item that is preferred in a majority of pairwise comparisons [15]. Unfortunately, it has been shown that optimizing a Kemeny optimal ranking from as few as four lists is NP-hard [3]. In our setting of rank aggregation in the presence of similarity information, we see that minimizing the the Kendall-tau similarity distance is also NP-hard, as optimizing this distance requires optimizing the Kendall-tau distance on aggregate similarity lists.

4 Extending Rank Aggregation Methods

Rank aggregation is a difficult problem even without the presence of similarity information, and has been

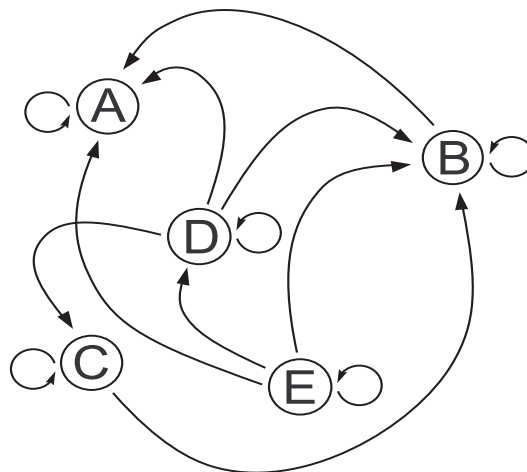


Figure 1: **Rank Aggregation with Markov Chains.** This Markov Chain represents the aggregation of lists (B, C, D), (A, D, E), and (A, B, E). Exact probabilities of transitions are determined by the choice of mapping method. Self-transitions represent the probability of staying at the node during a step.

studied formally since the eighteenth century. There is no universally accepted best method. Moreover, rank aggregation among similar items is, to our knowledge, previously unstudied. In this paper, we examine a family of Markov Chain rank aggregation methods which cover a range of standard approaches. Extensions to other methods, including positional and median-rank methods, appear in [13].

4.1 Markov Chain Rank Aggregation Methods

Markov Chain methods for ranked list aggregation represent the items in the various lists as nodes in a graph, with transition probabilities from node to node defined by the relative rankings of the items in the various lists. The aggregate rankings of the lists are found by computing (or approximating) the stationary distribution on the Markov chain – that is, by determining which nodes would be visited most often in a random walk on the graph. We augment these methods to include the case of similar items by adding *epsilon transitions* between nodes, based on similarity between the corresponding items.

Markov Chain Rank Aggregation. When transition probabilities are defined by relative rankings, the stationary distribution on a Markov chain implies an

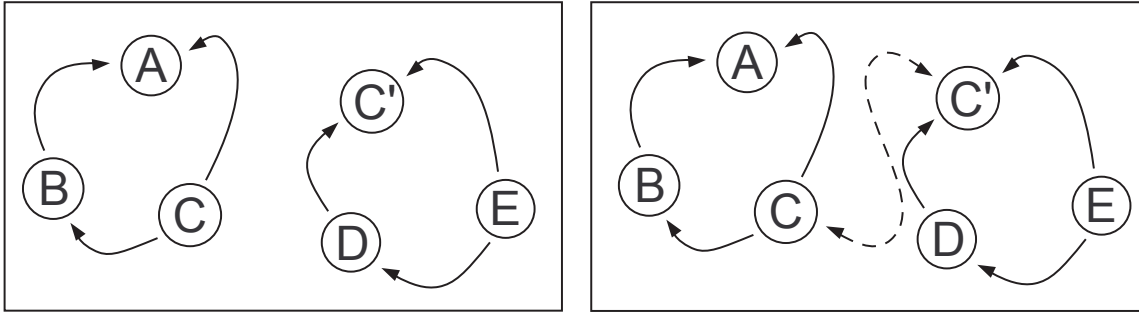


Figure 2: **Markov Chain Rank Aggregation with Similarity.** On the left, the Markov Chains are not able to merge the lists **A, B, C** and **C', D, E**, because the lists are completely disjoint. On the right, adding the dashed line of the similarity transition between **C** and **C'** enables a smooth ranking across all items. Note that epsilon transitions and self-transitions have been removed for clarity.

ordering on the nodes in M (see Figure 1). This observation led Dwork et al. to propose a general algorithm for rank aggregation, composed of three steps [3]:

1. Map the set of ranked lists R to a single Markov chain M , with one node per item in U .
2. Compute (or approximate) the stationary distribution π on M .
3. Rank the items in U from $1..n$ based on π . That is, the node with the highest score in π is given rank 1, and so on down to the node with the lowest score in π which is given rank n .

The key to this method is to define the appropriate mapping from the set of ranked lists R to a Markov Chain M . Dwork et al. proposed, analyzed, and tested four mapping schemes, which they dubbed MC1 through MC4 [3]. Although none of these methods necessarily produces a Kemeny optimal ranking, these methods have proven effective in practice [3].

4.2 Markov Chain Rankings with Similarity.

Although effective in many situations, the Markov Chain methods fail when the input lists are disjoint. (See Figure 2.) We address this issue with the addition of similarity transitions, which can link together otherwise isolated Markov Chain islands.

Similarity Transitions. Similarity transitions are defined from node to node based on the similarity measured between nodes. Thus, the ranking of an item will depend not only on those items it is ranked higher or lower than, but will also depend on the rankings of items that are similar.

A similarity transition from $S_k = i$, with respect to a given similarity function $s(\cdot, \cdot)$ is executed by choosing

an item j from U randomly from a weighted distribution in which for all $j \in U$

$$Pr(i \rightarrow j) = \frac{s(i, j)}{\sum_{l \in U} s(i, l)}$$

Note that if the similarity function is the uniqueness function $s_0(\cdot, \cdot)$, then a similarity transition will always result in $S_{k+1} = S_k$.

Furthermore, note that we also include small, uniform probability *epsilon transitions* from every node to every other node. These eliminate the possibility of *sink nodes* in the Markov chain, and ensure a smooth, complete ranking on all the items in U . This smoothing technique appears in a number of Markov chain based ranking methods, including the Google PageRank algorithm [10].

Formally, an epsilon transition is executed at S_k by choosing an item j from U uniformly at random, and setting $S_{k+1} = j$.

The prior mappings MC1 through MC4 [3] are generalized to consider similarity information, resulting in the following mappings MCS1 through MCS4. Note that each of these mappings include two parameters: ϵ determines the probability of an epsilon transition, while γ determines the probability of a similarity transition.

MCS1: At $S_k = i$, transition to S_{k+1} as follows. Create a multiset A of all nodes j such that there is a list $r \in R$ such that $r(j) \leq r(i)$, and choose a candidate node j by selecting uniformly at random from the nodes in A . Let $S_{k+1} = j$ if $i \neq j$. Otherwise, execute a similarity transition with probability γ . If no similarity transition is executed, then execute an epsilon transition

with probability ϵ . If no epsilon transition is executed, let $S_{k+1} = i$.

This method pays a relatively small amount of attention to similarity information, and gives more weight to the relative rankings.

MCS2: At $S_k = i$, transition to S_{k+1} as follows. Choose a list r uniformly at random from the subset of the (possibly partial) lists $r \in R$ for which $r(i)$ is defined. Choose a node for S_{k+1} by choosing uniformly at random from the items j such that $r(j) \leq r(i)$. Let $S_{k+1} = j$ if $i \neq j$. Otherwise, execute a similarity transition with probability γ . If no similarity transition is executed, then execute an epsilon transition with probability ϵ . If no epsilon transition is executed, let $S_{k+1} = i$.

Dwork et al. show that **MC2** is a generalization of the geometric mean variant of Borda's method and argue that this method is most sensitive of the four to statistically significant minority opinions [3].

MCS3: At $S_k = i$, transition to S_{k+1} as follows. Choose a list r uniformly at random from the set of lists $r \in R$ such that $r(i)$ is defined. Choose an item j from r uniformly at random. If $r(j) < r(i)$, $S_{k+1} = j$, otherwise execute a similarity transition with probability γ . If no similarity transition is executed, then execute an epsilon transition with probability ϵ . If no epsilon transition is executed, let $S_{k+1} = i$.

MCS4: At $S_k = i$, transition to S_{k+1} as follows. Choose an item j uniformly at random from U . if $r(j) < r(i)$ for a majority of the lists that ranked both i and j , $S_{k+1} = j$, otherwise execute a similarity transition with probability γ . If no similarity transition is executed, then execute an epsilon transition with probability ϵ . If no epsilon transition is executed, let $S_{k+1} = i$.

Dwork et al. show that this method is a generalization of Copeland's method of ranking items based on the number of pairwise majority contests won [3].

5 Experiments

In this section, we report the results on data from a real world application drawn from keyword expansion for sponsored search, and find that the addition of similarity information to rank aggregation is a significant benefit in this environment of noisy, incomplete rankings. In [13], additional experiments are performed supporting this claim, using synthetic data and including comparisons to several other methods.

The following parameters were used in our various aggregation methods, all of which were set by testing on small, independent tuning sets. The value of λ in λ -similarity projections was zero. For the Markov

Chain methods, $\epsilon = .01$, and $\gamma = 1$, and the stationary distribution was approximated by the power method.

5.1 Real World Data: Keywords Expansion.

We test our methods experimentally with data drawn from a real-world application: keywords expansion for sponsored search. In sponsored search, a service provided by many Internet search engines such as Yahoo!, an advertiser places bids on a set of keywords. When a user searches for that keyword, the search engine may show the advertiser's listing in a premium area, and the advertiser will pay the bid price to the search engine if the user clicks on the listing.

It is advantageous to both advertisers and search engines to make sure that the advertiser is bidding on as large a list of relevant keywords as possible – this reduces unused inventory and optimizes both market efficiency and user traffic. The goal of *keywords expansion* is to automatically suggest additional relevant keywords to a given advertiser. There are many methods for generating such suggestions, including the use of taxonomies, mining the text of the advertiser's website, and analyzing aggregate user behavior and keyword use. Each of these methods has incomplete coverage of the total keyword space and is subject to noise. The various methods of generating suggestions use different scoring mechanisms, on different scales, and perhaps with different levels of confidence. This is a perfect application for rank aggregation with similarity information, which can meaningfully join ranked lists of non-overlapping, similar items – even on partial lists in the presence of moderate noise.

We tested our methods on ten data sets, each with three ranked lists of keywords generated by distinct methods of keyword suggestion. These lists were highly disjoint, and contained noise. The similarity measure used on the keywords was a simple n -gram string matching kernel [8], with $n = 2$. Other similarity measures, such as those based on edit distance, could also have been used.

The results in Table 5 show that in this experiment, rank aggregation with similarity consistently outperformed non-similarity aggregation methods, as measured with the scaled induced Kendall tau similarity distance. This experiment serves as a real world example of noisy rankings, with a mixture of characteristics of partial and top- k lists, which benefits from the addition of similarity information to rank aggregation.

6 Discussion

The use of similarity information in rank aggregation has a wide range of potential applications. It could be used in spam reduction [3], importance ranking of

Table 1: **Aggregating Keyword Rankings.** Results for rank aggregation on keywords expansion data for scaled Footrule similarity distance. Results were consistent with those of the scaled Kendall similarity distance, except that BMS had relatively better performance on that measure, and MCS4 had relatively worse performance.

SET ID:	0	1	2	3	4	5	6	7	8	9
MCS4	0.197	0.209	0.198	0.189	0.213	0.176	0.192	0.196	0.208	0.186
MC4	0.242	0.257	0.255	0.245	0.268	0.208	0.266	0.256	0.242	0.242
MCS3	0.209	0.227	0.218	0.203	0.226	0.181	0.212	0.201	0.216	0.203
MC3	0.242	0.256	0.254	0.240	0.268	0.207	0.267	0.257	0.242	0.242
MCS2	0.221	0.242	0.239	0.219	0.253	0.188	0.245	0.226	0.223	0.226
MC2	0.240	0.253	0.253	0.246	0.271	0.207	0.279	0.260	0.242	0.242
MCS1	0.224	0.246	0.240	0.218	0.255	0.191	0.256	0.238	0.223	0.224
MC1	0.241	0.253	0.253	0.248	0.269	0.208	0.283	0.264	0.241	0.241

web pages [10], or to improve the quality of results in meta-search [3]. It is also possible to imagine using rank aggregation with similarity in the field of social choice, and designing elections in which party affiliation is used as a similarity measure between candidates. In the fields of data mining and machine learning, rank aggregation with similarity has its strongest use as a method of unsupervised learning, in which items are implicitly clustered as an effective aggregate ranking is found. Future work will lie in further exploring ways in which these goals can be simultaneously achieved.

7 Acknowledgments

This work was performed in the Data Mining and Research group of Yahoo!, Inc., under the direction of Pavel Berkhin, Rajesh Parekh, and Scott J. Gaffney. Thanks are given to Anselm Blumer, Carla E. Brodley, and Ravi Kumar for their comments on this work.

References

- [1] M.-J. Condorcet. Essai sur l'application de l'analyse a la probabilité des décisions rendues a la pluralité des voix, (1785).
- [2] P. Diaconis and R. L. Graham. Spearman's Footrule as a Measure of Disarray *Journal of the Royal Statistical Society, Series B (Methodological)* (1977), 39:2, pp. 262–268
- [3] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web, *WWW '01: Proceedings of the 10th international conference on World Wide Web*, (2001), pp. 613–622.
- [4] R. Fagin, R. Kumar and D. Sivakumar. Efficient similarity search and classification via rank aggregation, *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (2003), pp. 301–312.
- [5] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top-k lists. *Proceedings of the 2003 ACM-SIAM Symposium on Discrete Algorithms*(2003), pp. 28–36..
- [6] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar and E. Vee. Comparing and aggregating rankings with ties, *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (2004), pp. 47–58.
- [7] M. G. Kendall. *A New Measure of Rank Correlation*, *Biometrika* (1938), 30:1/2, pp. 81–93.
- [8] C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: a string kernel for svm protein classification, *Proceedings of the Pacific Symposium on Biocomputing* (2002), pages 564–575.
- [9] S. Nene and S. Nayar. A Simple Algorithm for Nearest Neighbor Search in High Dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1997), 19:9, pp. 989–1003.
- [10] L. Page, S. Brin, R. Motwani, T. Winograd. The PageRank citation ranking: Bringing order to the Web, Technical report (1998), Stanford University, Stanford, CA, USA.
- [11] D. G. Sarri. The mathematics of voting: democratic symmetry. *Economist*. p. 83, March 4, 2000.
- [12] B. Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press (2001).
- [13] D. Sculley. Rank Aggregation for Similar Items. Technical Report TR-2007-1 (2007), Tufts University, Medford, MA, USA.
- [14] J. Shawe-Taylor and N. Cristianini *Kernel Methods for Pattern Analysis* Cambridge University Press (2004).
- [15] H. P. Young and A. Levenglick. A Consistent Extension of Condorcet's Election Principle, *SIAM Journal on Applied Mathematics* (1978), 35:2, pp. 285–300.