An Iterative Distance-Based Model for Unsupervised Weighted Rank Aggregation

Leonidas Akritidis
Data Structuring & Engineering Lab
Electrical and Computer Engineering
University of Thessaly
Volos, Greece
leoakr@e-ce.uth.gr

Athanasios Fevgas
Data Structuring & Engineering Lab
Electrical and Computer Engineering
University of Thessaly
Volos, Greece
fevgas@e-ce.uth.gr

Panayiotis Bozanis
Data Structuring & Engineering Lab
Electrical and Computer Engineering
University of Thessaly
Volos, Greece
pbozanis@e-ce.uth.gr

ABSTRACT

Rank aggregation is a popular problem that combines different ranked lists from various sources (frequently called voters or judges), and generates a single aggregated list with improved ranking of its items. In this context, a portion of the existing methods attempt to address the problem by treating all voters equally. Nevertheless, several related works proved that the careful and effective assignment of different weights to each voter leads to enhanced performance. In this article, we introduce an unsupervised algorithm for learning the weights of the voters for a specific topic or query. The proposed method is based on the fact that if a voter has submitted numerous elements which have been placed in high positions in the aggregated list, then this voter should be treated as an expert, compared to the voters whose suggestions appear in lower places or do not appear at all. The algorithm iteratively computes the distance of each input list with the aggregated list and modifies the weights of the voters until all weights converge. The effectiveness of the proposed method is experimentally demonstrated by aggregating input lists from six TREC conferences.

CCS CONCEPTS

• **Information systems → Rank aggregation**; • **Theory of computation → Unsupervised learning and clustering**.

KEYWORDS

rank aggregation, weighted rank aggregation, unsupervised rank aggregation, distance-based, unsupervised learning

ACM Reference Format:
Leonidas Akritidis, Athanasios Fevgas, and Panayiotis Bozanis. 2019. An Iterative Distance-Based Model for Unsupervised Weighted Rank Aggregation. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19), October 14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3350546.3352547

WI '19, October 14–17, 2019, Thessaloniki, Greece

ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352547

1 INTRODUCTION

Nowadays, there is an increasing number of applications which provide aggregated information originating from multiple sources. In particular, these systems initially collect multiple lists of ranked elements, such as preferences, suggestions, behaviors, etc., from their users. In the sequel, they combine the provided input and construct a unified "golden" (or winning) list which has its items ranked according to several criteria. Examples of such applications include social recommendation systems, voting interfaces for various entities (e.g. products, services, hotels, articles, etc.), social choice platforms, collaborative filtering algorithms, and so on.

The objective of rank aggregation methods is the determination of an ideal ranking of elements, based on the preferences of diverse sources of information, frequently called *voters*, or *judges*. Although quite old, rank aggregation still attracts the attention of the researchers due to its adoption by a wide variety of expert and intelligent systems. Examples include voting and recommendation platforms [5, 6, 10, 16], metasearching [1, 2, 8, 12], spam detection [8], bioinformatics [7, 11, 13], NLP [14, 15], and others.

The majority of the relevant algorithms consider that all voters are of equal importance and treat them equivalently [3, 8, 12]. On the other hand, several studies introduce unsupervised methods that learn the importance of the voters and use this information to improve the quality of the aggregated list. For instance, in [13] the authors propose a Monte Carlo cross-entropy algorithm based on a consensus ranking which minimizes the distance from all individual lists. This objective transforms rank aggregation into a standard optimization problem, broadly known as Kemeny optimal aggregation. The method of [4] learns a static, query-independent weight vector for the voters and uses this vector to aggregation.

In this paper, we introduce an unsupervised algorithm for the determination of the degree of expertise of the voters in rank aggregation problems. It is not a stand-alone method, but, instead, it attaches itself to other existing non weighted approaches, converting them to weighted methods. It is based on the rationale that a voter, who submits a list that is close to the final aggregate list L, should be considered as expert and treated differently than a voter whom suggestions are ranked low in L.

More specifically, after the processing of the input lists of the voters and the construction of the aggregated list L, the distances between each input list and L are computed. Then, a kernel function assigns higher weights to the expert voters and the aggregation method is reapplied in a weighted fashion. This procedure is applied iteratively until the weights converge to their final value and the final aggregated list is stabilized.

2 PRELIMINARY

The analysis begins with the introduction of a set V of $n = |V|$ voters. Given a query q, each voter $v \in V$ submits a ranked list $R^v = \{r_1^v, \dots, r_k^v\}$ of k elements. An item $r_i^v \in R^v$ is retrieved from a universe U of elements and has two attributes; namely, the voter v who submitted it, indicated in the superscript, and its position (or ranking) i in R^v, denoted in the subscript. In case the lists of all voters contain all the elements of U under different orderings, then they are called *full lists* or *permutations*; otherwise, they are named *partial* or *top-k* lists. Notice that the proposed method is applicable to both permutations and top-k lists.

A rank aggregation method $\mathcal{T}$ accepts n input lists and fuses them into a single output list $L = \{r_1, r_2, \dots\}, r_i \in U$. In the simplest case, $\mathcal{T}$ treats all voters equally and their submitted lists are processed in an unbiased manner. Then, L is simply given by:

$$L = \mathcal{T}\big(R^{v_1}, R^{v_2}, \dots, R^{v_n}\big) \tag{1}$$

On the other hand, in weighted scenarios, each voter v is assigned a weight w_v that reflects her/his degree of expertise or importance. In this case, $\mathcal{T}$ accepts both the input lists R^v and the weights of all voters w_v. Consequently, the output list L is derived as follows:

$$L = \mathcal{T}\big(w_{v_1}, w_{v_2}, \dots, w_{v_n}, R^{v_1}, R^{v_2}, \dots, R^{v_n}\big) \tag{2}$$

Moreover, the proposed method requires the computation of the distance $d(R^v, L)$ of each input list R^v from the aggregated list L. There exist several such distance metrics, with the Spearman's footrule and Kendall's τ being the most popular among them. However, these distances apply to full lists only. If the individual lists R^v are not full but top-k ones, then the distance between two ranked lists can be calculated by employing the *scaled* versions of these metrics. For instance, the scaled footrule distance is given by:

$$d_{F'}(R^v, L) = \sum_{j=1}^{k} \left| \frac{j}{k} - \frac{l_j}{|L|} \right| \tag{3}$$

These measures have been used extensively by multiple works related to the distance-based rank aggregation. However, they are insensitive to the "locality" of the differences between two lists; that is, they do not take into account how far from the top of the aggregated list these disagreements are found. For this reason, we introduce a variant of the scaled footrule distance, which we name *locality-sensitive scaled footrule distance*, defined as follows:

$$d_{F'}(R^v, L) = \sum_{j=1}^{k} \left| \frac{j}{k} - \frac{l_j}{|L|} \right| \log \frac{|L|}{l_j} \tag{4}$$

This metric does not only compute the scaled distance between two elements, but it also modifies this distance by multiplying it with a factor that embodies the position l_j of the element r_j^v in L.

3 UNSUPERVISED WEIGHT LEARNING

The key idea is to assign to each voter v an importance score (or weight) w_v that depends on the distance of her/his ranked list R^v from the aggregated list L. The rationale is that if the suggested elements of v appear in high positions in L, then these elements are apparently valuable. Consequently, v should be considered as an important source of information and must be assigned a higher weight compared to other voters whose selections have been ranked lower in L.

This logic is served by the following approach: Initially, we apply a traditional rank aggregation method $\mathcal{T}$ that treats all voters equally, according to Eq. 1. Let us consider that L_0 is the output of this process. Then, we compute the distance $d(R^v, L_0)$ of each input list R^v from L_0, and we modify the weights of each voter with respect to $d(R^v, L_0)$. Finally, we reapply $\mathcal{T}$ in the weighted fashion indicated by Eq. 2, and we obtain another list L_1.

More formally, the modification of the weights is determined by a kernel function f of the distance $d(R^v, L)$, as follows:

$$w_{v,1} = w_{v,0} + f\big(d(R^v, L_0)\big) \tag{5}$$

where $w_{v,1}$ and $w_{v,0}$ represent the new and the initial weight of v, respectively. The initial weights $w_{v,0}$ can be set to any arbitrary fixed value (e.g., $w_{v,0} = 1$); in our implementation, we used $w_{v,0} = 1/n$, where n is the total number of voters.

After the new weight values have been computed for all voters, the rank aggregation method $\mathcal{T}$ is reapplied, leading to a new aggregated list L_1. Notice that Eq. 5 is designed to favor the expert voters, since it grants higher importance to those who selected multiple important items; that is, items that have been highly ranked in both R^v and L_0.

This approach can be generalized and applied in an iterative fashion. According to it, Eq. 5 can be rewritten as follows:

$$w_{v,i} = w_{v,i-1} + f\big(d(R^v, L_{i-1})\big), \quad i \in \mathbb{N}^* \tag{6}$$

The kernel function f must satisfy the following requirements:

- Provided that $d(R^v, L) \geq d(R^{v'}, L)$, it must hold that $f\big(d(R^v, L)\big) \leq f\big(d(R^{v'}, L)\big)$, and vice versa. That is, the higher the distance between an input list R^v and the aggregated list L is, the lower the value returned by f should be, and vice versa. Consequently, Eq. 6 will assign higher weights to the voters whose submitted lists have smaller distances from L.
- f should be upper bounded; that is, the weights of the voters must converge after a number of iterations have been executed. Otherwise, L will eventually become equal to the input list R^v which was initially closest to it.

The simple exponential function satisfies both requirements:

$$f\big(d(R^v, L)\big) = \exp\big(i \cdot d(R^v, L)\big) \tag{7}$$

where i represents the i-th iteration; its presence in the exponent renders f more stable and accelerates its convergence. By plugging Eq. 7 to Eq. 6 we obtain the final form for the weights:

$$w_{v,i} = w_{v,i-1} + \exp\big(i \cdot d(R^v, L_{i-1})\big), \quad i \in \mathbb{N}^* \tag{8}$$

or, as an expression of partial sums:

$$w_{v,i} = w_{v,0} + \sum_{j=1}^{i} \exp\big(j \cdot d(R^v, L_{j-1})\big), \quad i \in \mathbb{N}^* \tag{9}$$

In Algorithm 1 we present the steps of the proposed method. The variables w'_v and w_v are used to store the values of the previous and the current voter weight, respectively. Initially, all voters are assigned the same weight $w'_v = 1/|V|, \forall v \in V$. Next, the initial aggregated list L is created by applying any non weighted rank aggregation method $\mathcal{T}$. The variable i is an iteration counter that gets updated before each iteration. Moreover, *allconverged* is a

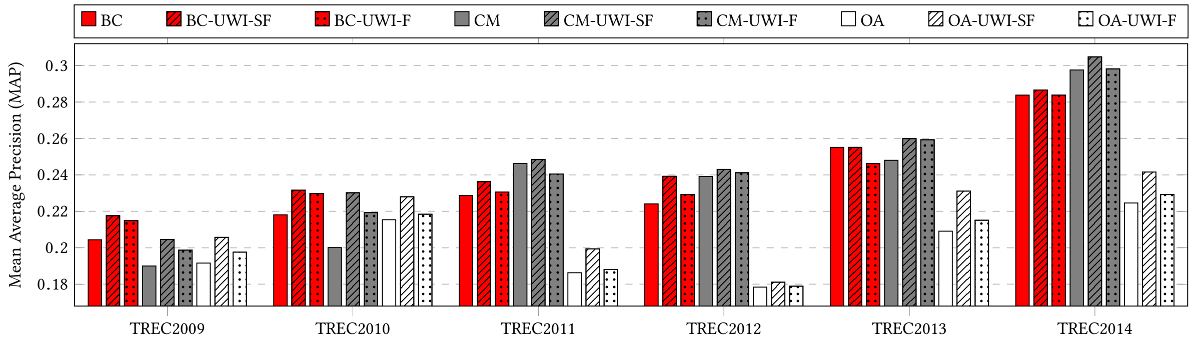

Figure 1: Performance comparison of various rank aggregation methods on the Adhoc datasets of TREC 2009–2014.

boolean variable that determines whether *all* the voter weights have converged, whereas *prec* is the precision of the difference $w_v - w'_v$ under which we assume that convergence has been achieved.

The algorithm includes two loops. The outer one performs the iterations and its exit condition is activated only when *allconverged* becomes true. In other words, the outer loop ends when no change in L is possible any more. Before this loop begins, *allconverged* is set to false and it is immediately switched to true before each iteration. *allconverged* will remain true if and only if all the weights of the voters change by a margin which is smaller than $prec$.

The inner loop iterates through the voters and updates their weights according to our previous presentation. For each voter $v \in V$, the distance $d(R^v, L)$ is computed and the new weight w_v is calculated according to Eq. 8. When the new weights of all voters have been computed, the weighted version of the rank aggregation method $\mathcal{T}$ is employed and a new aggregated list L is constructed according to Eq. 2. The current iteration ends by overwriting the previous voter weights with the new ones.

In overall, Algorithm 1 needs $n_{it}|V|$ distance and $(n_{it} + 1)$ aggregated list computations, n_{it} the number of iterations. As it is exhibited in Section 4, n_{it} on average takes values that can be characterized from small to moderate.

Before we close this presentation, we briefly refer to the manner the voter weights are used by $\mathcal{T}$. In case $\mathcal{T}$ is a positional method (e.g. Borda Count), its weighted version merely replaces the plain ranking j of an item r_j^v by the product $j \cdot w_v$. On the other hand, in order-based methods, the weights can be used to modify the number of wins of each item.

Algorithm 1: Distance-based weighted rank aggregation

initialize an empty aggregated list L;
for *each voter* $v \in V$ **do**
    $w'_v \leftarrow 1/|V|$;
end
$L \leftarrow \mathcal{T}(R^{v_1}, R^{v_2}, \ldots, R^{v_n})$ (Eq. 1);
$i \leftarrow 0$; *allconverged* $\leftarrow$ false;
while *not allconverged* **do**
    $i \leftarrow i + 1$; *allconverged* $\leftarrow$ true;
    for *each voter* $v \in V$ **do**
        Compute $d(R^v, L)$ (Eq. 3, or 4);
        Compute normalized $d(R^v, L)$;
        Set $w_v \leftarrow w'_v + \exp\left(i \cdot d(R^v, L)\right)$ (Eq. 8);
        if $w_v - w'_v > prec$ **then**
            allconverged $\leftarrow$ false;
        end
    end
    $L \leftarrow \mathcal{T}\left(w_{v_1}, \ldots, w_{v_n}, R^{v_1}, \ldots, R^{v_n}\right)$ (Eq. 2);
    for *each voter* $v \in V$ **do**
        $w'_v \leftarrow w_v$;
    end
end

4 EXPERIMENTAL EVALUATION

In this section, we report the results of the experimental evaluation of the introduced method. The datasets we used originate from six Adhoc tasks of TREC (Text REtrieval Conference –http://trec.nist.gov) organized between 2009 and 2014. The evaluation was conducted by employing the corresponding relevance judges.

Three of the most popular rank aggregation methods were employed with the aim of evaluating the proposed model: (i) *Borda Count (BC)*, (ii) *Condorcet method (CM)*, and (iii) the *Outranking approach (OA)* of [9]. These are the baseline methods. The application of Algorithm 1 to each one of them leads to *BC-UWI*, *CM-UWI*, and *OA-UWI*, respectively, where the suffix *UWI* stands for the term *Unsupervised Weighted Iterative*. We shall collectively refer to these methods as *UWI extensions*. We tested two flavors of the UWI extensions: (i) by using our proposed locality sensitive scaled footrule distance of Eq. 4 (leading to the *UWI-SF* methods), and (ii) by employing the standard scaled footrule distance of Eq. 3 (*UWI-F* methods). In all cases, the *prec* parameter was set equal to 0.1.

Figure 1 illustrates the performance of the UWI extensions against their baseline methods on the six aforementioned TREC datasets. The retrieval effectiveness was measured by using Mean Average Precision (MAP), a popular measure that is broadly used to evaluate IR systems and algorithms.

The best performing baseline method for the Adhoc task of TREC 2009 was Borda Count, since its MAP was roughly 0.204. The application of our model in BC-UWI-SF led to a considerable increase in performance, since its MAP was 0.217, enhanced by

Table 1: Performance comparison of various rank aggregation methods with different evaluation metrics on the Adhoc datasets of the Web Tracks of TREC 2009–2014.

Dataset	Method	MAP	P@5	P@10	P@20	P@100	nDCG@10	nDCG@20
TREC 2009	Borda Count	0.2044	0.5040	0.4980	0.4620	0.2964	0.3972	0.3761
	Borda Count-UWI-SF	**0.2162**	0.5440	0.5020	**0.4750**	**0.3056**	**0.3996**	**0.3773**
	Condorcet method	0.1900	0.4240	0.4380	0.4410	0.2874	0.3436	0.3450
	Condorcet method-UWI-SF	0.1987	0.4680	0.4760	0.4450	0.2976	0.3752	0.3543
	Outranking approach	0.1916	0.5440	0.5060	0.4450	0.2888	0.3691	0.3423
	Outranking approach-UWI-SF	0.2057	**0.5640**	**0.5160**	0.4510	0.2918	0.3725	0.3543
TREC 2010	Borda Count	0.2181	0.4960	**0.4580**	0.4080	0.2766	0.2596	0.2572
	Borda Count-UWI-SF	**0.2316**	**0.5000**	**0.4580**	0.4130	0.2902	**0.2599**	**0.2578**
	Condorcet method	0.2001	0.4320	0.3980	0.3900	0.2814	0.2116	0.2290
	Condorcet method-UWI-SF	0.2193	0.4160	0.4060	0.3920	0.2932	0.2206	0.2383
	Outranking approach	0.2154	0.4760	0.4560	0.4180	0.2818	0.2514	0.2546
	Outranking approach-UWI-SF	0.2280	0.4930	0.4560	**0.4240**	**0.2972**	0.2564	0.2560
TREC 2011	Borda Count	0.2287	0.4080	0.3600	0.3360	0.2326	0.2417	0.2392
	Borda Count-UWI-SF	0.2363	0.4400	0.3620	0.3300	0.2354	0.2373	0.2387
	Condorcet method	0.2463	**0.4520**	**0.3760**	**0.3610**	0.2490	0.2713	0.2697
	Condorcet method-UWI-SF	**0.2484**	**0.4520**	**0.3760**	0.3520	**0.2524**	**0.2751**	**0.2730**
	Outranking approach	0.1863	0.4120	0.3460	0.3010	0.1932	0.2380	0.2176
	Outranking approach-UWI-SF	0.1994	0.4240	0.3460	0.3010	0.2074	0.2457	0.2260
TREC 2012	Borda Count	0.2241	0.4000	0.3800	0.3420	0.2366	0.1605	0.1632
	Borda Count-UWI-SF	0.2392	0.4360	0.3860	0.3490	0.2398	0.1688	0.1683
	Condorcet procedure	0.2391	**0.4720**	0.4520	0.3960	0.2446	0.2066	0.2046
	Condorcet method-UWI-SF	**0.2430**	0.4680	**0.4560**	**0.4030**	**0.2458**	**0.2112**	**0.2063**
	Outranking approach	0.1784	0.3120	0.2960	0.2880	0.2034	0.1329	0.1484
	Outranking approach-UWI-SF	0.1811	0.3240	0.3120	0.2980	0.2056	0.1345	0.1501
TREC 2013	Borda Count	0.2551	0.4880	0.4620	0.4190	0.2724	0.2509	0.2677
	Borda Count-UWI-SF	0.2551	0.5000	**0.4680**	**0.4260**	0.2730	**0.2559**	**0.2733**
	Condorcet method	0.2480	0.4600	0.4220	0.4090	0.2684	0.2385	0.2637
	Condorcet method-UWI-SF	**0.2599**	0.4720	0.4560	0.4200	**0.2816**	0.2506	0.2691
	Outranking approach	0.2091	0.4960	0.4420	0.3870	0.2438	0.2310	0.2425
	Outranking approach-UWI-SF	0.2311	**0.5120**	0.4560	0.3900	0.2604	0.2409	0.2517
TREC 2014	Borda Count	0.2838	0.5960	0.5820	0.5460	0.3382	0.2815	0.3020
	Borda Count-UWI-SF	0.2866	0.5940	0.5800	0.5520	0.3456	0.2912	0.3159
	Condorcet method	0.2976	**0.6000**	**0.5800**	**0.5720**	0.3712	0.2880	0.3177
	Condorcet method-UWI-SF	**0.3048**	**0.6000**	**0.5880**	**0.5720**	**0.3728**	**0.2928**	**0.3207**
	Outranking approach	0.2246	0.5160	0.5080	0.4530	0.2978	0.2385	0.2487
	Outranking approach-UWI-SF	0.2416	0.5160	0.5080	0.4550	0.3112	0.2512	0.2619

about 6.8%. The respective gain with the scaled footrule distance of Eq. 3 (BC-UWI-F) also improved the baseline method, but by a smaller margin of 5.4%.

On the other hand, OA and CM were less effective and achieved 0.192 and 0.190, respectively. The UWI-SF extensions of OA and CM also exhibited improved effectiveness compared to the baseline methods. In particular, the gains were 7.4% (MAP=0.206) and 4.6% (MAP=0.199), respectively.

The results for TREC 2010 were even better than those of TREC 2009. More specifically, BC-UWI-SF and BC-UWI-F outperformed BC by 6.4% and 5.5%, respectively; the MAPs we measured for these three methods were about 0.232, 0.230, and 0.218, respectively. The performance of OA was close to that of BC this time, since its MAP was equal to 0.215. On the contrary, CM was considerably less effective and was outperformed by the baseline BC by a margin of 9%. Nonetheless, both OA and BC were benefited by our model, since the corresponding UWI-SF extensions improved their performance by roughly 5.8% and 9.6%, respectively.

Regarding the Adhoc tasks of TREC 2011, 2012, and 2014, the situation was reversed and the Condorcet method has been the best performing method, leaving Borda Count in the second place. The Outranking approach was substantially outperformed in these tasks. More specifically, CM achieved Mean Average Precisions equal to 0.246, 0.239, and 0.298 for the three tasks, respectively, and outperformed BC by 7.6%, 6.7%, and 4.8%, respectively. On the other hand, OA was 32-35% less effective. In every case, all three methods were improved by our model, even by a small margin. The highest gains were observed for BC in the Adhoc task of TREC 2012 (+6.7% for UWI-SF/+2.8% for UWI-F) and OA in TREC 2011 (+7% for UWI-SF/+1.1% for UWI-F). Notice that, in all cases, our proposed distance function of Eq. 4 led to higher performance compared to the standard scaled footrule distance.

Table 1 contains additional evaluation metrics which could not be plotted due to space restrictions. For the same reason, we only present the measurements for UWI-SF. In particular, we report the mean Precision values for the first 5, 10, 20, and 100 results, whereas the last two columns show the values of the normalized discounted cumulative gain (nDCG) at positions 10 and 20. The values of MAP, which are illustrated in Fig. 1, are shown in the third column.

Notice that in all experiments, the winning method was either an UWI-SF extension, or there was a tie between an UWI-SF extension and a baseline method.

Regarding P@5, BC was improved by a remarkable margin of 7.8%, 7.9% and 9% in the datasets of TREC 2009, 2011, and 2012, respectively. In the rest datasets, the benefits were smaller and were limited below 3%. In the other two baseline methods, CM and OA, the greatest gains in P@5 were 10.4% and 3.8% and were measured in the datasets of TREC 2009 and 2012, respectively.

Performance gains of similar magnitude were also observed with respect to the other three Mean precision values, i.e., P@10, P@20, and P@100. We only provide some indicative results which demonstrate the beneficial effect of the proposed model. In the dataset of TREC 2013, CM-UWI-SF outperformed CM by 8%. In TREC 2012, OA-UWI-SF won its baseline counterpart by 3.5%, and in TREC 2011, OA-UWI-SF dominated over OA by a margin of 7.4%.

Similarly to all previous evaluation metrics, the measurements of nDCG also reflected the usefulness of our approach. Regarding nDCG@10, significant improvements were observed for CM in the Adhoc tasks of TREC 2009 (+9.1%), 2010 (+4.3%), and 2013 (+5%). For Borda Count and the Outranking approach, the greatest gains in performance in terms of nDCG@10 were achieved in the TREC 2013 dataset (i.e., 3.5%), and TREC 2014, respectively (5.3%).

Table 2 contains the required number of iterations for each UWI extension before the voters' weights converge to their final value. As stated earlier, these results have been obtained for $prec = 0.1$, and indicate that, although the UWI-SF extensions perform better than the UWI-F ones, they require slightly more iterations. Moreover, the OA-UWI extensions converge faster than the others, whereas the ones of BC require the most iterations.

Table 2: Average number of iterations per query before weights convergence for various UWI extensions

Dataset	BC-UWI		CM-UWI		OA-UWI	
	-SF	-F	-SF	-F	-SF	-F
TREC 2009	14.10	12.48	7.26	5.62	7.32	6.14
TREC 2010	14.70	12.96	6.98	5.56	6.90	5.72
TREC 2011	12.22	10.24	7.26	5.72	6.50	5.62
TREC 2012	9.50	8.78	6.48	5.52	5.80	5.22
TREC 2013	15.12	11.90	8.42	6.32	7.26	5.82
TREC 2014	11.70	10.38	7.28	5.50	6.26	5.30

5 CONCLUSIONS

In this paper we introduced an unsupervised distance-based model for weighted rank aggregation. The proposed method is based on the intuition that *the degree of expertise of a voter depends on the distance of her/his ranked list from the final aggregated list.* Hence, the voters whose submitted lists are close to the aggregated list should be treated as experts and be assigned higher weights. According to this rationale, the proposed method produces an aggregated list *that is proximal to the input lists of the expert voters and has larger distances from the non-expert ones.*

The model is based on an iterative procedure which modifies the weights of the voters, thus leading to a new improved aggregated list at the end of each iteration. The new distances of the individual lists from the new aggregated list are then recomputed and the weights of the voters are updated accordingly. This process is repeated until the weights of all voters converge to their final stable value.

The experimental evaluation was conducted by extending three state-of-the-art rank aggregation methods, namely, Borda Count, the Condorcet method and the Outranking approach of [9]. The extended methods were compared against their baseline non weighted versions by using datasets and relevance judgments from the Adhoc task of six TREC conferences. The results verified the usefulness of our proposal, since the method we introduce here led to an increase in performance which exceeded 10% in some cases.

REFERENCES

[1] Akritidis, L., Katsaros, D., and Bozanis, P. Effective Ranking Fusion Methods for Personalized Metasearch Engines. In *Proceedings of the 12th Panhellenic Conference on Informatics* (2008), pp. 39–43.

[2] Akritidis, L., Katsaros, D., and Bozanis, P. Effective Rank Aggregation for Metasearching. *Journal of Systems and Software 84*, 1 (2011), 130–143.

[3] Aslam, J. A., and Montague, M. Models for metasearch. In *Proceedings of the 24th ACM International Conference on Research and Development in Information Retrieval* (2001), pp. 276–284.

[4] Cohen, W. W., Schapire, R. E., and Singer, Y. Learning to order things. In *Advances in Neural Information Processing* (1998), pp. 451–457.

[5] de Borda, J. C. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences* (1781).

[6] De Condorcet, N. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, 1785.

[7] DeConde, R. P., Hawley, S., Falcon, S., Clegg, N., Knudsen, B., and Etzioni, R. Combining Results of Microarray Experiments: A Rank Aggregation Approach. *Statistical Applications in Genetics and Molecular Biology 5*, 1 (2006).

[8] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. Rank Aggregation Methods for the Web. In *Proceedings of the 10th International Conference on World Wide Web* (2001), pp. 613–622.

[9] Farah, M., and Vanderpooten, D. An Outranking Approach for Rank Aggregation in Information Retrieval. In *Proceedings of the 30th ACM Conference on Research and Development in Information Retrieval (SIGIR)* (2007), pp. 591–598.

[10] Kilgour, D. M. Approval Balloting for Multi-winner Elections. In *Handbook on approval voting*. Springer, 2010, pp. 105–124.

[11] Kolde, R., Laur, S., Adler, P., and Vilo, J. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics 28*, 4 (2012), 573–580.

[12] Meng, W., Yu, C., and Liu, K.-L. Building efficient and effective metasearch engines. *ACM Computing Surveys 34*, 1 (2002), 48–89.

[13] Pihur, V., Datta, S., and Datta, S. Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach. *Bioinformatics 23*, 13 (2007), 1607–1615.

[14] Rosti, A.-V., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R., and Dorr, B. Combining outputs from multiple machine translation systems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics* (2007), pp. 228–235.

[15] Sagae, K., and Lavie, A. Parser combination by reparsing. In *Proceedings of the Human Language Technology Conference of the NAACL* (2006), pp. 129–132.

[16] Young, H. P., and Levenglick, A. A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics 35*, 2 (1978), 285–300.