

Sequential Rank Aggregation Method

Yu-ling Lin
School of Management
University of Bath
Bath, UK
yll41@bath.ac.uk

Salem Chakhar
Portsmouth Business School
University of Portsmouth
Portsmouth, UK
salem.chakhar@port.ac.uk

Rui Yang
Portsmouth Business School
University of Portsmouth
Portsmouth, UK
rui.yang@port.ac.uk

Abstract—Rank aggregation is a fundamental technique of different application domains. In this paper, we propose a new rank aggregation method. This method models the rank aggregation problem as an assignment problem and solves it by integer programming, where the objective function is set to minimize the sum of the squared Euclidean Distance between each initial ranking and the aggregated ranking. To avoid the computational limitation in working with large datasets, a sequential aggregation approach has been adopted. This approach proceeds sequentially in several steps. In each step, only two rankings are aggregated. It thus reduces the computational limitation of the proposed method. An illustration of the proposed method using datasets of green car adoption in Taiwan is presented in this paper. The results show that the proposed method can solve the rank aggregation problem effectively and efficiently.

Index Terms—rank aggregation, ranking function, distance measure, optimization

I. INTRODUCTION

Rank aggregation seeks to meaningfully combine multiple individual rankings, and thus obtain a consensual and comprehensive new ranking. Rank aggregation is a fundamental technique, which has application in a variety of areas including meta-search, recommendation systems, database systems, information retrieval, collaborative filtering, bioinformatics, social choice and multi-criteria decision making. A series of rank aggregation methods have been proposed and applied in different domains (see, e.g. [1]–[12]). Some reviews on rank aggregation methods are available in [13]–[15].

The main issues encountered in solving rank aggregation problems are (i) the requirement of additional parameters (as observed in the probabilistic methods [12]), the problem of ties [16] and similar items [17], the diversity of preference representations ways [18] and the computational limitations in solving rank aggregation problems with large datasets [19].

In this paper, we address the above-cited issues by modeling rank aggregation problem as an assignment problem and solving it using 0-1 integer programming, where the objective function is set to minimize the sum of the squared Euclidean Distance between each initial rankings and the aggregated ranking. Since this formulation may lead to large mathematical programs that are difficult to be solved, a new sequential aggregation approach is then proposed in this paper.

The basic idea of this approach is to select a pair of rankings and combine them into a temporary ranking. The two initial rankings are then removed and replaced by the new ranking.

This process is repeated until a single final aggregated ranking is obtained. The method is illustrated using a dataset of green cars adoption in Taiwan. The results show that the proposed method can solve the rank aggregation problem effectively and efficiently.

The rest of the paper is organized as follows. Section II introduces the rank aggregation problem. Section III provides the mathematical formulation of this problem. Section IV details the resolution procedure. Section V illustrates the proposed method using real-world data. Section VI compares different aggregation strategies. Section VII evaluates the performances of the proposed method. Section VIII discusses some related work. Section IX concludes the paper.

II. PROBLEM DESCRIPTION

A. Ranking and Ranking Function

Let U be a set of n objects (or elements) Z_1, \dots, Z_n . An ordered list X with respect to U is an ordering (or ranking) of the elements of U . A ranking X is a list of the form $Z_{(1)}, \dots, Z_{(n)}$ with $Z_{(i)}$ is the element of order i , $Z_{(i)} \succeq X_{(j)}$ ($\forall i, j = 1, \dots, n$) and where ‘ \succeq ’ is an ordering relation on U .

Following [4] and for notational convenience, we define a ranking function r as $r : U \times U \rightarrow \{>, =, <\}$. The specification of the ranking function requires generally the definition of a scoring function s which is defined as a mapping $s : U \rightarrow \mathbb{R}$, assigning a numerical score $s(Z_i)$ to each object $Z_i \in U$.

We can then specify the ranking function as follows:

$$r(Z_i, Z_j) = \begin{cases} >, & s(Z_i) > s(Z_j), \\ =, & s(Z_i) = s(Z_j), \\ <, & s(Z_i) < s(Z_j). \end{cases} \quad (1)$$

Accordingly, three situations will hold for any pair of objects Z_i and Z_j in ranking X : (i) $Z_i \succ Z_j$, (ii) $Z_i \prec Z_j$ or (iii) $Z_i = Z_j$.

B. Rank Aggregation Problem

Let X^1, \dots, X^m be a collection of m rankings and let X_i^l be the rank of object $Z_i \in U$ ($i = 1, \dots, n$) in ranking X^l ($l = 1, \dots, m$). The objective of rank aggregation problem is to construct a new ranking Y by aggregating the initial rankings X^1, \dots, X^m . Rank aggregation problem can be

considered as a mapping from $X^1 \times \dots \times X^m$ into Y . The aggregated ranking should summarize as good as possible the initial rankings. One possible solution to this issue is to select an aggregated ranking Y with a minimal distance to the initial rankings. There are different ways to define this distance but in this paper the Squared Euclidean Distance will be used. The Squared Euclidean Distance is not a metric as Euclidean Distance (because it does not satisfy the triangle inequality) but it is frequently used in optimization problems in which only distances have to be compared.

The computation of the aggregated ranking is NP-hard [20], even when the number of initial rankings to be aggregated is only 4 [3]. Therefore, in this paper the computation of the ranking is modelled as an assignment problem and solved by integer programming.

C. Desirable Property of Aggregated Ranking

Since there may be several possible aggregated rankings, a desirable property of the aggregated ranking is defined as follows. Let Y_i ($i = 1, \dots, n$) be the aggregated rank of object $Z_i \in U$. Then, for each $Z_i \in U$, we define:

$$\underline{\mu}(Z_i) = \min_{l=1, \dots, m} X_i^l. \quad (2)$$

$$\bar{\mu}(Z_i) = \max_{l=1, \dots, m} X_i^l. \quad (3)$$

A desirable property of the aggregated ranking Y is that $Y_i \in [\underline{\mu}(Z_i), \bar{\mu}(Z_i)]$, for all $i = 1, \dots, n$. An object Z_i with an aggregated ranking $Y_i \notin [\underline{\mu}(Z_i), \bar{\mu}(Z_i)]$ will be called outlier in this paper.

D. Appropriateness of Rank Aggregation Method

A comprehensive and appropriate rank aggregation method should be able to address the issues mentioned in the introduction, i.e.: (i) requiring no or very few parameters; (ii) being enable to deal with ties and similar items; (iii) accepting different preference representations ways; and (iv) being enable to manage large datasets. Among them, the authors contend that the most important qualities of a comprehensive and appropriate rank aggregation method are to do with the computational time and the management of ties. In addition, the total number of outliers produced in the method should be minimized. Therefore, in this paper we argue that an appropriate rank aggregation method should be characterized by the following:

- reduced computation time for large datasets;
- minimal number of outliers; and
- ability to manage ties.

These qualities are ordered according to their criticality from the most to the least critical.

III. MATHEMATICAL FORMULATION

Let X^l ($l = 1, \dots, m$) be an initial ranking. The squared Euclidean Distance between X^l and Y is obtained by squaring the standard Euclidean distance as follows:

$$d^2(X^l, Y) = (X_1^l - Y_1)^2 + \dots + (X_n^l - Y_n)^2 \quad (4)$$

$$d^2(X^l, Y) = \sum_{i=1}^n (X_i^l - Y_i)^2 \quad (5)$$

The squared Euclidean Distance allows to place progressively greater weight on objects that are farther apart, which is a desired feature in our case.

The sum of squared Euclidean Distance between the aggregated ranking Y and the initial rankings X^1, \dots, X^m is then computed as follows:

$$d^2(X^1, Y) + \dots + d^2(X^m, Y) = \sum_{l=1}^m \sum_{i=1}^n (X_i^l - Y_i)^2 \quad (6)$$

The computation of the aggregated ranking Y is modelled as an assignment problem and solved by integer programming as follows. First, for each initial ranking X^l ($l = 1, \dots, m$), we construct a $n \times n$ distance matrix D^l . An element D_{ij}^l of D^l measures the the distance between the rank of object Z_i in initial ranking X^l and the aggregated rank j of object Z_i in the aggregated ranking Y . We denote also by C_{ij}^l ($i, j = 1, \dots, n$; $l = 1, \dots, m$) the dummy variables of the problem. Then, the objective function is defined as follows:

$$Z = \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n D_{ij}^l \cdot C_{ij}^l \quad (7)$$

The binary variables C_{ij}^l should be constrained as follows:

$$\sum_{i=1}^n C_{ij}^l = 1, j = 1 \dots, n; l = 1, \dots, m \quad (8)$$

$$\sum_{j=1}^n C_{ij}^l = 1, i = 1 \dots, n; l = 1, \dots, m \quad (9)$$

The first set of constraints ensure that each object is assigned a single rank while the second set of constraints ensure that each rank is assigned once to a single object. The last set of constraints specify that all variables should be binary.

The final 0-1 integer program that should be solved is then defined as follows:

(P) :

$$\begin{aligned}
\min \quad & Z = \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n D_{ij}^l \cdot C_{ij}^l \\
\text{s.t.} \quad & \sum_{i=1}^n C_{ij}^l = 1, j = 1 \dots, n; l = 1, \dots, m \\
& \sum_{j=1}^n C_{ij}^l = 1, i = 1 \dots, n; l = 1, \dots, m \\
& C_{ij}^l \in \{0, 1\}, \quad i, j = 1 \dots, n; l = 1, \dots, m
\end{aligned} \tag{10}$$

The final ranking is obtained based on the dummy variables C_{ij}^l ($i, j = 1 \dots, n$). The dummy variable $C_{ij}^l = 1$ means that object Z_j is ranked i in the aggregated ranking Y .

IV. RESOLUTION PROCEDURE

The difficulty of solving the 0-1 integer program (P) arises from the high number of variables that should be considered. For instance, for $m = 5$ initial rankings and $n = 10$ objects it involves the definition of 250 variables. The solution to this issue proposed in this paper consists in using a sequential aggregation approach which considers only two rankings at each step. This idea is illustrated graphically by Figure 1 for $m = 4$ initial rankings X^1, \dots, X^4 . The symbols T^1 and T^2 in this figure refer to intermediate rankings, ‘agg’ sets for the aggregation operation and Y is the final ranking.

Theoretically, there are $\prod_{\alpha=0}^{m-2} C_2^{m-\alpha}$ possible aggregation strategies. For instance, for three initial rankings X^1, X^2 and X^3 , there are three possible processing strategies as illustrated in Figure 2. The symbols T and Y in this figure refer to intermediate and final rankings, respectively and ‘agg’ sets for the aggregation operation.

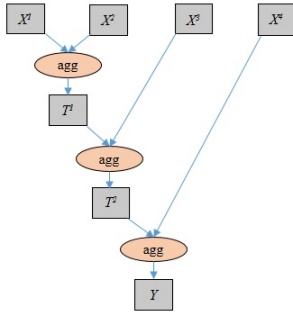


Fig. 1. Sequential Aggregation Approach of four Initial Rankings

In this paper, we advocate by aggregating at each step the two rankings X^i and X^j for which the absolute value of Kendall’s tau coefficient is the nearer to zero. The use of Kendall’s tau minimizes the number of outliers because, intuitively, the number of outliers increases with higher positive correlation. When two rankings are nearer to each other, the chance that the aggregated rank of an object Z_i falls outside the interval $[\mu(Z_i), \bar{\mu}(Z_i)]$ is higher. This is due to the optimization-based formulation of the rank aggregation

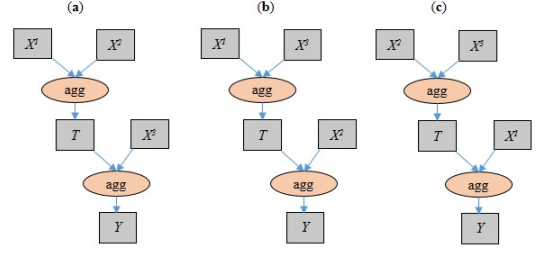


Fig. 2. Processing Strategies

problem. It is important to note that other non-parametric statistics such Spearman’s rho and Cohen’s kappa may be used for selecting the rankings to combine at each step. It is also important to mention that different numerical tests with different datasets show that the use of Kendall’s tau reduces the number of outliers. However, this should not be considered as a rule and more formal solutions to rankings selection should be investigated.

In light of this discussion, we designed the resolution procedure given in Algorithm 1. The description of this algorithm is straightforward. At this level we just mention that the aggregated ranking Y^{k+1} is obtained by solving mathematical programming relative to the aggregation of X^l and $X^{l'}$.

Algorithm 1: Aggregation

Input : $X = X^1, \dots, X^m$, // initial rankings.
Output: Y , // aggregated ranking.

- 1 $k \leftarrow m$;
- 2 **while** ($|X| \geq 2$) **do**
- 3 • Compute the Kendall’s tau coefficients between all the rankings in X ;
- 4 • Select two rankings X^l and $X^{l'}$ from X for which the absolute value of Kendall’s tau coefficient is minimal. If ties exist, arbitrary select one;
- 5 • Solve the mathematical programming relative to the aggregation of X^l and $X^{l'}$;
- 6 • Replace X^l and $X^{l'}$ in X with the aggregated ranking Y^{k+1} of X^l and $X^{l'}$;
- 7 • $k \leftarrow k + 1$;
- 8 **return** Y^{k-1} ;

V. ILLUSTRATION

A. Decision Problem

In the last two decades, Taiwanese’s government has been striving to address the environmental impact caused by ever-increasing car consumption and manufacture in Taiwan. Consequently, the Taiwanese government has rolled out a variety of green policies, such as tax reduction for green car consumers, research funding for green technology development by car manufacturers, etc. However, the existing green policies in Taiwan have not been entirely successful and the outcomes of those green policies are largely below the government’s expectations. This is mainly due to the fact that current policies were made solely based on the government’s green objectives at the time without taking sufficient consideration of car consumers’ and car manufacturer’s objectives. The aim of this illustrative example is to construct a new car consumption

model, which takes into account the government, consumers and manufacturers expectations jointly.

B. Dataset

The dataset used in this paper is given in Table I. It is an extract of the car dataset that covers the years from 2011 to 2015. This dataset was obtained from Taiwan’s Directorate General of Highways, and Ministry of Transportation and Communications. Table I contains three initial rankings of 27 green car models. The first ranking represents the Greenness Level of considered car models. It reflects the government’s objectives in addressing environmental issues. This ranking has been specified by the policy makers based on the fuel efficiency of cars and other technical parameters such as swept volume and car horsepower. The second ranking is defined based on Total Sales of considered car models. This ranking represents the consumer’s objectives in green car consumption. The last ranking is defined based on the Price of considered car models prices. This ranking represents the manufacturers’s objectives with respect to maximizing benefits.

TABLE I
INITIAL RANKINGS

Car Model	X^1 : Rank by Greenness Level	X^2 : Rank by Total Sales	X^3 : Rank by Price
1	7	24	19
2	17	10	3
3	20	9	17
4	23	4	14
5	27	25	18
6	16	8	2
7	9	13	25
8	18	3	13
9	12	15	26
10	4	18	5
11	3	26	27
12	22	5	6
13	10	14	8
14	24	7	12
15	2	11	7
16	15	16	10
17	21	2	15
18	11	22	24
19	5	17	11
20	14	19	4
21	6	27	21
22	25	23	9
23	8	21	23
24	1	1	1
25	13	12	22
26	19	20	20
27	26	6	16

C. Application and Results

The aggregation procedure has been applied to dataset in Table I. The first step of the aggregation procedure is to compute the Kendall’s tau coefficients between the rankings X^1 , X^2 and X^3 in Table I. The result is given in Table II. The second step of the algorithm is select two rankings for

which absolute value of Kendall’s tau coefficient is minimal. Based on Table II, the two rankings for which absolute value of Kendall’s tau coefficient is minimal are X^1 and X^3 .

TABLE II
KENDALL’S TAU BETWEEN INITIAL RANKINGS

Kendall’s tau	X^1	X^2	X^3
X^1	1	-0.23646	-0.025641
X^2		1	0.2991452
X^3			1

The third step is to solve the mathematical program corresponding to X^1 and X^3 . The program looks as follows:

$$\begin{aligned} \min \quad & Z = 360 C_{11}^1 + 260 C_{12}^1 + \dots + 113 C_{27,26}^2 + 122 C_{27,27}^2 \\ \text{s.t.} \quad & C_{1,1}^1 + C_{2,1}^1 + \dots + C_{26,1}^2 + C_{27,1}^2 = 1 \\ & C_{27,1}^1 + C_{27,2}^1 + \dots + C_{27,26}^2 + C_{27,27}^2 = 1 \end{aligned}$$

where all the dummy variables $C_{1,1}^1, C_{1,2}^1, \dots, C_{27,26}^2, C_{27,27}^2$ are defined as binary. This mathematical program has been solved using the GLPK solver [21]. The obtained dummy variables leads to the intermediate ranking $T^{1,3}$, which is given in the fourth column of Table III. In the second iteration of the aggregation procedure, the intermediate ranking $T^{1,3}$ is aggregated with initial ranking X^2 by solving the corresponding mathematical program. The obtained final ranking Y^{s_2} is given in the fifth column of Table III.

TABLE III
AGGREGATED RANKINGS

Car Model	Strategy 1		Strategy 2		Strategy 3	
	$T^{1,2}$	Y^{s_1}	$T^{1,3}$	Y^{s_2}	$T^{2,3}$	Y^{s_3}
1	18	19	10	19	23	14
2	14	8	8	6	4	6
3	16	17	22	16	14	22
4	11	11	23	13	7	15
5	27	26	27	27	22	27
6	9	4	6	3	2	4
7	6	15	16	15	19	12
8	3	5	15	5	5	8
9	13	21	24	22	21	20
10	4	3	3	8	13	3
11	15	23	13	23	27	13
12	12	9	12	4	3	9
13	8	7	5	7	10	5
14	19	14	21	14	9	21
15	2	2	2	2	8	2
16	20	13	9	11	15	16
17	7	10	20	10	6	11
18	24	27	18	24	25	23
19	5	6	4	9	16	7
20	23	12	7	12	12	10
21	22	24	11	21	26	19
22	26	18	17	25	17	26
23	17	22	14	20	24	18
24	1	1	1	1	1	1
25	10	16	19	17	18	17
26	25	25	25	26	20	25
27	21	20	26	18	11	24

For comparison purposes, we applied the aggregation procedure using three different aggregation strategies: (1) Strategy 1: aggregate X^1 and X^2 into intermediate ranking $T^{1,2}$ and then aggregate $T^{1,2}$ and X^3 to obtain the final ranking Y^{s_1} ; (2) Strategy 2: aggregate X^1 and X^3 into intermediate ranking $T^{1,3}$ and then aggregate $T^{1,3}$ and X^2 to obtain the final ranking Y^{s_2} ; (3) Strategy 3: aggregate X^2 and X^3 into intermediate ranking $T^{2,3}$ and then aggregate $T^{2,3}$ and X^1 to obtain the final ranking Y^{s_3} . The corresponding results are summarized in Table III.

VI. COMPARATIVE STUDY

We first compared the final rankings Y^{s_1} , Y^{s_2} and Y^{s_3} in Table III with respect to the number of outliers. The result of this comparison is given in Table IV. Table IV shows that, in this particular application, the selection of the two rankings, for which the absolute value of Kendall's tau coefficient is minimal in each step of the aggregation procedure, is the best in terms of the number of outliers.

TABLE IV
NUMBER OF OUTLIERS

Strategy	1	2	3
Number of outliers	4	3	5

We also used a series of non-parametric statistics to compare the initial and computed rankings. The statistics considered in this paper are: Kendall's tau, Spearman's rho, and the Unweighted and Weighted Cohen's kappa. The results of the statistical comparison are provided in Tables V and VI. Table V presents the agreement levels between initial and aggregated rankings. Based on this table, we can conclude that the aggregated rankings are nearer to initial ranking X^3 followed by initial ranking X^2 and finally initial ranking X^1 . This can be explained as follows. First, about 67% objects in X^3 are between the ranks in X^1 and X^2 . In addition, as shown in Table VI, the agreement level between X^2 and X^3 is higher than the agreement level between X^1 and X^3 and that between X^1 and X^2 . Table VI shows the agreement levels between aggregated rankings. Based on this table, we can conclude that the agreement levels between all the aggregated rankings are either high or very high, with a slightly higher agreement level between Y^{s_1} and Y^{s_2} . Table VI also shows that the agreement levels between Y^{s_1} and Y^{s_3} and between Y^{s_2} and Y^{s_3} are almost the same.

VII. PERFORMANCE ANALYSIS

A collection of randomly generated datasets have been used to evaluate the performance of the proposed rank aggregation method. The datasets vary with respect to the number of objects, the number of initial rankings and the presence/absence of ties. We first compared the evolution of the computing time with respect to the number of objects and the number of rankings with and without ties. The results are shown graphically in Figures 3 and 4, respectively. These graphs show that the computation time increases smoothly with

TABLE V
AGREEMENT LEVELS BETWEEN INITIAL AND AGGREGATED RANKINGS

Statistics	Initial Ranking	Aggregated		
		Y^{s_1}	Y^{s_2}	Y^{s_3}
Kendall's tau	X^1	0.1738	0.1453	0.4074
	X^2	0.4416	0.4929	0.2764
	X^3	0.5535	0.5897	0.3994
Spearman's rho	X^1	0.2295	0.2283	0.5574
	X^2	0.6422	0.6954	0.4005
	X^3	0.8016	0.7692	0.6227
Unweighted Cohen's kappa	X^1	0.0385	0.1154	0.0769
	X^2	0	0	0.0385
	X^3	0.0769	0.1154	0
Weighted Cohen's kappa	X^1	0.1346	0.1181	0.3489
	X^2	0.3901	0.4148	0.2170
	X^3	0.5467	0.5797	0.3901

TABLE VI
AGREEMENT LEVELS BETWEEN INITIAL RANKINGS

Statistics		X^1	X^2	X^3
Kendall's tau	X^1	1	-0.2365	-0.0256
	X^2		1	0.2991
	X^3			1
Spearman's rho	X^1	1	-0.2839	-0.0513
	X^2		1	0.4377
	X^3			1
Unweighted Cohen's kappa	X^1	1	0	0
	X^2		1	0.0385
	X^3			1
Weighted Cohen's kappa	X^1	1	-0.1374	-0.0797
	X^2		1	0.2170
	X^3			1

TABLE VII
AGREEMENT LEVELS BETWEEN AGGREGATED RANKINGS

Statistics		Y^{s_1}	Y^{s_2}	Y^{s_3}
Kendall's tau	Y^{s_1}	1	0.8348	0.7322
	Y^{s_2}		1	0.7265
	Y^{s_3}			1
Spearman's rho	Y^{s_1}	1	0.9536	0.8791
	Y^{s_2}		1	0.8864
	Y^{s_3}			1
Unweighted Cohen's kappa	Y^{s_1}	1	0.3462	0.1538
	Y^{s_2}		1	0.1923
	Y^{s_3}			1
Weighted Cohen's kappa	Y^{s_1}	1	0.8269	0.6951
	Y^{s_2}		1	0.6868
	Y^{s_3}			1

respect to the number of objects and the number of rankings. Furthermore, Figures 3 and 4 indicate that the presence of ties has no significant effect on the computation time.

We also compared the variation in the number of outliers with respect to the number of objects and the number of rankings with and without ties. The results are shown graphically in Figure 5 and Figure 6, respectively. It can be seen that the percentage of outliers decreases when the number of initial rankings increases. For a number of rankings between 2 to 5,

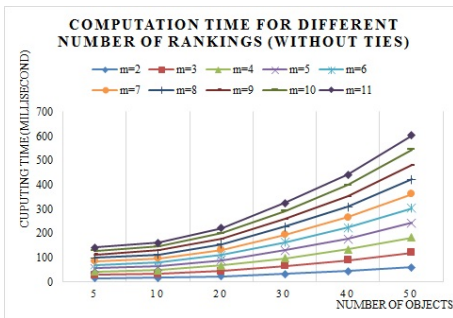


Fig. 3. CPU Time vs # of Objects and # of Rankings (Without Ties)

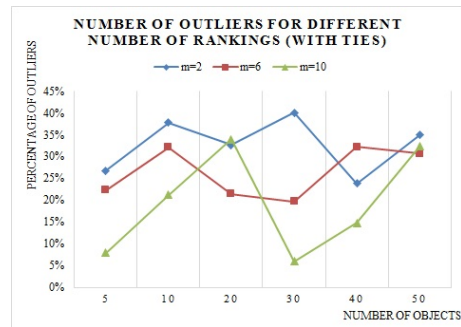


Fig. 6. Percentage of Outliers vs # of Objects and # of Rankings (With Ties)

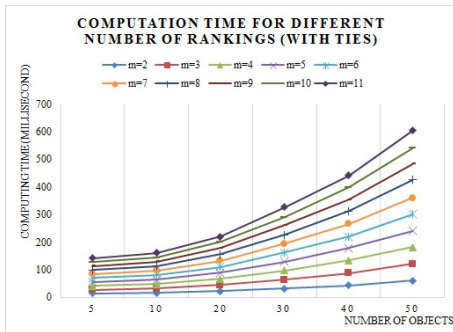


Fig. 4. Computation Time vs # of Objects and # of Rankings (With Ties)

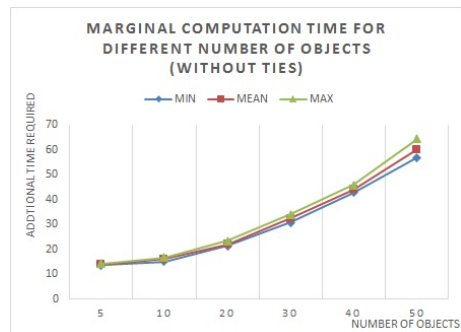


Fig. 7. Marginal Computation Time vs # of Objects (Without Ties)

there are a significantly higher percentage of outliers than the case of 6 to 11 rankings. However, comparing Figures 5 and 6, it appears that the percentage of outliers are higher in the presence of ties.

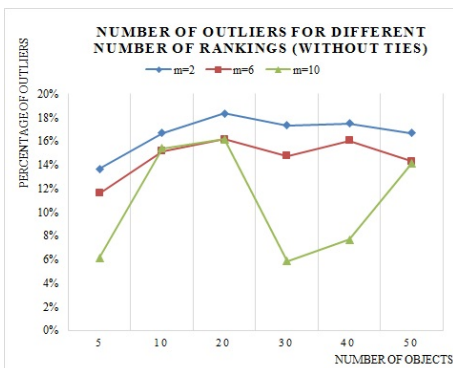


Fig. 5. Percentage of Outliers vs # of Objects and # of Rankings (Without Ties)

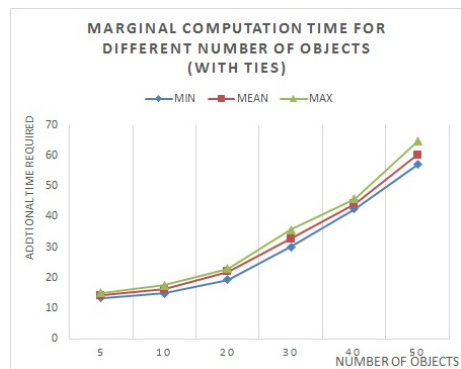


Fig. 8. Marginal Computation Time vs # of Objects (With Ties)

We also evaluated the variation in marginal computation time with respect to the number of objects. The marginal computation time is defined as the difference in computation time of solving rank aggregation problems with m and $m - 1$ initial rankings, respectively. Results are shown graphically in Figures 7 and 8, respectively. These figures indicate an almost linear increase in computing time.

VIII. RELATED WORK

The authors in [18] introduced a flexible probabilistic model to solve the rank aggregation problem. It allows the consideration of different preference forms over the considered objects. In [11], the authors proposed a robust rank aggregation model via matrix learning, which recovers a latent rank list from the possibly incomplete and noisy input rank lists. The authors in [6] presented a mathematical and algorithmic framework for learning to aggregate (partial) rankings without supervision. This permits to handle rank aggregation problems where domain knowledge are expensive to acquire.

The authors in [22] used listwise learning to rank with crowdsourcing labels obtained from multiple annotators. This

approach addresses the problem of unreliability in the existing concept of learning to rank, which assumes that each training instance is associated with a reliable label. This assumption is not necessarily true as it may be infeasible or remarkably expensive to obtain reliable labels for many learning to rank applications. The authors in [16] introduced a large scale study of algorithms for rank aggregation with ties and evaluated it with a very large panel of both real and carefully generated synthetic datasets.

The paper [12] addresses the limitations of Luce and Mallows rank aggregation models in terms of expressiveness or computational complexity. They proposed a new model, which was defined by a coset-permutation distance, and modeled the generation of a permutation as a stagewise process. This model has rich expressiveness and low complexity. They applied the proposed model to supervised rank aggregation, derived the learning and inference algorithms, and empirically studied their effectiveness and efficiency.

IX. CONCLUSION

Rank aggregation problem has received an increasing attention with new application domains characterized by large datasets, such as web search and browsing, information retrieval and multicriteria decision making. In this paper, we proposed a new rank aggregation method. This method relies on 0-1 integer programming formulation where the objective function is set to minimize the sum of the squared Euclidean Distance between each initial ranking and the aggregated ranking. To avoid computational limitations of using large datasets, a sequential aggregation approach has been adopted.

The method is illustrated using datasets of green car adoption in Taiwan. The results show that the proposed method can solve the rank aggregation problem effectively and efficiently. The results also show that the proposed processing strategy effectively reduces the number of outliers.

In this paper, the squared Euclidean Distance has been used to measure the similarity degree between the aggregated ranking and the initial rankings. The proposed sequential aggregation approach is generic and can also be used with other distance measures such as the Kendall's distance [23], Kendall's τ [24] or Spearman's footrule distance [25].

Several topics need to be considered in the future. First, other distance functions than the squared Euclidean Distance can be used in future research. Second, formal rules can be designed to identify the two rankings to combine at each step of the proposed method. Third, the performance and computational behavior of the proposed method can be evaluated with very large datasets. Fourth, the proposed method can be compared with other existing approaches (e.g. [20]) in terms of its effectiveness and efficiency.

REFERENCES

- [1] I. Caragiannis and E. Micha, "Learning a ground truth ranking using noisy approval votes," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 149–155.
- [2] S. Chatterjee, A. Mukhopadhyay, and M. Bhattacharyya, "A weighted rank aggregation approach towards crowd opinion analysis," *Knowledge-Based Systems*, vol. 149, pp. 47 – 60, 2018.
- [3] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proceedings of the 10th International Conference on World Wide Web*. New York, NY, USA: ACM, 2001, pp. 613–622.
- [4] P. Flach and E. Matsubara, "On classification, ranking, and probability estimation," in *Probabilistic, Logical and Relational Learning - A Further Synthesis*, ser. Dagstuhl Seminar Proceedings, L. de Raedt, T. Dietterich, L. Getoor, K. Kersting, and S. Muggleton, Eds., no. 07161. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.
- [5] A. Foroughi and M. Tamiz, "An effective total ranking model for a ranked voting system," *Omega*, vol. 33, no. 6, pp. 491 – 496, 2005.
- [6] A. Klementiev, D. Roth, and K. Small, "Unsupervised rank aggregation with distance-based models," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, New York, NY, USA, 2008, pp. 472–479.
- [7] F. Lotfi, M. Rostamy-Malkhalifeh, N. Aghayi, Z. Beigi, and K. Gholami, "An improved method for ranking alternatives in multiple criteria decision analysis," *Applied Mathematical Modelling*, vol. 37, no. 1, pp. 25 – 33, 2013.
- [8] B. Hofreiter and S. Marchand-Maillet, "Rank aggregation for QoS-aware web service selection and composition," in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, 2013, pp. 252–259.
- [9] X. Li, X. Wang, and G. Xiao, "A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications," *Briefings in Bioinformatics*, p. bbx101, 2017.
- [10] A. Mayyas, A. Qattawi, M. Omar, and D. Shan, "Design for sustainability in automotive industry: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 4, pp. 1845–1862, 16 2012.
- [11] Y. Pan, H. Lai, C. Liu, Y. Tang, and S. Yan, "Rank aggregation via low-rank and structured-sparse decomposition," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, ser. AAAI'13. AAAI Press, 2013, pp. 760–766.
- [12] T. Qin, X. Geng, and T.-Y. Liu, "A new probabilistic model for rank aggregation," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'10. USA: Curran Associates Inc., 2010, pp. 1948–1956.
- [13] E. Fields, G. E. Okudan, and O. Ashour, "Rank aggregation methods comparison: A case for triage prioritization," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1305–1311, 2013.
- [14] S. Lin, "Rank aggregation methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 5, pp. 555–570, 2010.
- [15] Y. Xiao, Y. Deng, J. Wu, H.-Z. Deng, and X. Lu, "Comparison of rank aggregation methods based on inherent ability," *Naval Research Logistics*, vol. 64, no. 7, pp. 556–565, 2017.
- [16] B. Brancotte, B. Yang, G. Blin, S. Cohen-Boulakia, A. Denise, and S. Hamel, "Rank aggregation with ties: Experiments and analysis," *Proc. VLDB Endow.*, vol. 8, no. 11, pp. 1202–1213, 2015.
- [17] D. Sculley, *Rank Aggregation for Similar Items*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007, pp. 587–592.
- [18] M. Volkovs and R. Zemel, "A flexible generative model for preference aggregation," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 479–488.
- [19] P. Yu and H. Xu, "Rank aggregation using latent-scale distance-based models," *Statistics and Computing*, 2018, <https://doi.org/10.1007/s11222-018-9811-9>.
- [20] J. Ding, D. Han, and Y. Yang, "Iterative ranking aggregation using quality improvement of subgroup ranking," *European Journal of Operational Research*, vol. 268, no. 2, pp. 596–612, 2018.
- [21] A. Makhorin, *GLPK (The GNU Linear Programming Kit)*, GNU Software Foundation, 2018, available at <http://www.gnu.org/software/glpk> (accessed: March 17, 2018).
- [22] O. Wu, Q. You, F. Xia, L. Ma, and W. Hu, "Listwise learning to rank from crowds," *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 1, pp. 4:1–4:39, 2016.
- [23] M. Kendall, *Rank correlation methods*. London: Griffin, 1948.
- [24] M. Kadziński and M. Michalski, "Scoring procedures for multiple criteria decision aiding with robust and stochastic ordinal regression," *Computers & Operations Research*, vol. 71, pp. 54–70, 2016.
- [25] P. Diaconis and R. Graham, "Spearman's footrule as a measure of disarray," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 2, pp. 262–268, 1977.