Unsupervised Ranking of Numerical Observations based on Magnetic Properties and Correlation Coefficient

Khalid A Alattas[⋆•], Aminul Islam[⋆], Ashok Kumar[⋆], and Magdy Bayoumi[⋆]

[⋆]School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70503, USA
[•]Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia
Email: {kalattas, aminul, ashok, mab0778}@louisiana.edu

Abstract

This paper addresses a novel unsupervised algorithm to rank numerical observations which is important in many applications in computer science, especially in information retrieval (IR). The proposed algorithm shows how correlation coefficients between attribute values and the concept of magnetic properties can be explored to rank multi-attribute numerical objects. One of the main reasons of using correlation coefficients between attribute values and the concept of magnetic properties is that they are easy to compute and interpret. Our proposed **U***nsupervised* **R***anking using* **M***agnetic properties and* **C***orrelation coefficient (URMC) algorithm can use some or all the numerical attributes of objects and can also handle objects with missing attribute values. The proposed algorithm overcomes a major limitation of the state-of-the-art technique while achieving excellent results.*

1. Introduction

With the rapid growth of the uses of information retrieval (IR) and social choice, ranking or categorization has become one of the key techniques for handling and organizing data. Ranking techniques are used to assign weights to the attributes of a specific dataset, to ultimately rank the objects in that dataset. This ranking helps any end user to make a decision on that dataset in a more efficient way. Ranking by hand is difficult, time-consuming, costly, and subjective, especially for a large dataset.

Ranking of multi-attribute objects are divided into two categories [1]. The first category comes with completely labeled training data and uses supervised ranking algorithms. The second category, unsupervised ranking algorithms, is more challenging because no ground truth data is available. For multi-attribute objects, majority of the datasets come with no ground truth dataset. This is because of the cost involved to create the ground truth dataset as well as the lack of any acceptable evaluation method.

Previous works on the unsupervised ranking of multi-attribute objects are primarily based on feature selection of attributes [2, 3]. Works based on feature selection of attributes use different techniques and rules to select the most important or relevant attributes for ranking. One of the main problems of feature selection on the current unsupervised ranking of multi-attribute objects is that each technique selects different attributes than others. However, removing some attributes from datasets seems more challenging and could have an effect on the result of ranking.

The traditional approaches of unsupervised ranking use complex rules to rank multi-attribute objects. Moreover, some of the techniques of unsupervised ranking cannot deal with missing value fields. For example, ranking principal curve (RPC) algorithm [1] requires full lists of attributes because it cannot deal with missing value attributes. As a matter of fact, most of the datasets are coming with missing values due to unavailable data or not enough information for the objects. From this perspective, URMC has the potential to rank multi-attribute objects using some (or all) of the attributes of a dataset. URMC also has the potential to deal with the problems of missing values in attributes by using correlation coefficient between attributes of a dataset. This is because correlation coefficient between attributes can be computed without much variation in the result even with some missing values in the attributes. A correlation coefficient (r) has been a fundamental and efficient tool for data analysis and information retrieval by finding the strength measures of a linear association between two attributes and ranges between -1 (perfect negative correlation) to +1 (perfect positive correlation) [4]. URMC uses Pearson's correlation coefficient (r) as this is the most common measure of correlation and is used when the value of variables are continuous.

In this paper, we propose a new algorithm that is inspired from magnetic properties. URMC algorithm cluster the attributes into two clusters (i.e., positive

URI: https://hdl.handle.net/10125/59553
ISBN: 978-0-9981331-2-6

and negative cluster) and place each attribute a weight by using Pearson (r) correlation. The idea of using magnetic properties is that if the correlation coefficient between two attributes is positive, it means that they attract each other to be in the same cluster, otherwise they repulse each other to be in different clusters. In later stage, attribute weights are used to compute the ranking of the objects.

Overall, we make the following contributions in this work.

1. We propose URMC algorithm for unsupervised ranking of multi-attribute objects. This algorithm uses magnetic properties and the correlation coefficient between each distinct pair of attributes to update the clusters and attribute weights of a dataset.

2. The proposed algorithm can deal with *all attributes*, so there is no need to select relevant attributes and remove the irrelevant ones. Actually, URMC algorithm assigns higher weight to relevant attributes and lower weight to irrelevant ones.

3. URMC algorithm can deal with *missing value* in the attributes.

The rest of this paper is organized as follows: the related works are discussed in Section 2. Our proposed unsupervised ranking algorithm (URMC) is described in Section 3. We use a walk-through example in Section 4. The experimental results on three datasets are described in Section 5. Direction for future research are briefly described in Section 6 and the paper is concluded.

2. Related Works

Web search data is a common example of both supervised and unsupervised rank aggregation. Rank aggregation is to combine ranking results of attributes from multiple ranking functions in order to produce a better attribute. Supervised rank aggregation only considers the linear model of base rankers for aggregation function [5]. Unsupervised ranking aggregation is widely used in the context of meta-search. It works by integrating the ranked list of documents returned by multiple search engine in response to a given query [6]. ULARA is a common example of the framework of an unsupervised algorithm for rank aggregation based on permutations [7–9]. The central idea of this method is that the large weights will be considered if the rank lists are closed to the average rank list, for each object. On the contrary, the smaller weights will be considered if the rank lists are quite different from the average rank list. NDCG [10] and MAP [11] are extensively used in web search indicators to evaluate the supervised ranking performance which comprises the label of target ranking. TREC and LETOR are paradigms of existing supervised ranking methods that focus on the search ranking symmetric with NDCG and MAP that are evaluated on two datasets of query searching result [12–14]. Furthermore, most existing unsupervised ranking aggregation methods focus on search ranking such as PageRank algorithm [15]. PageRank algorithm is the most famous unsupervised ranking which is used by Google Search to rank websites in the Google search engine outcome.

One problem with unsupervised ranking is how to provide a favorable ranking outcome since no ground truth label is available. For example, world universities, journals, sports, and countries datasets do not have target ranking available. This kind of ranking we can refer to as ranking of multi-attribute objects. Multi-Cluster Feature Selection (MCFS) and Multi-Cluster Feature Selection via Smooth Distributed Score (MCFS-SDS) are types of unsupervised ranking that use feature selection and work for clustering according to [16,17]. Various studies show which attributes (features) should be selected, and which should be removed to perform ranking. These attributes which should be selected have some impact in ranking. While the attributes which should be removed are irrelevant. The attribute with a high value is considered relevant to ranking. According to spectral feature selection [2] they describe for both supervised and unsupervised framework of spectral feature selection and show the potential of selected feature (attribute). The authors exploit the actual properties underlying the supervised and unsupervised feature selection based on spectral graph theory.

Two well-known state-of-the-art unsupervised ranking algorithms are two-phase attribute ordering for unsupervised ranking [3] and RPC [1]. Two-phase attribute ordering for unsupervised ranking [3] uses two phases. The first phase, Spearman Ranking Correlation Coefficients (SRCC), identifies irrelevant attributes that can adversely affect the ranking, and the second phase uses Extended Fourier Amplitude Sensitivity Test that presents the total effect for each attribute to ranking and then selects the attributes base on those phases. The idea for the first phase is to distinguish between attributes and identify the irrelevant attributes by using two rules: strict monotonicity and smoothness. All attributes selected are considered as monotonically related to ranking. SRCC distinguishes between attributes to recognize irrelevant attributes before ranking to avert irrelevant attributes. The second phase is carried out from reduced dataset to provide

a quantity of important measure for each attribute. These methods address the attribute selection for unsupervised ranking tasks. As we know, the ranking of a journal would be higher if it has a higher citation. It is not wise to remove it as not important or irrelevant attribute as mentioned in [3]. Ranking principal curve [1] proposed five meta-rules for unsupervised ranking which are scale and translation invariance, strict monotonicity, compatibility of linearity and nonlinearity, smoothness, and explicitness of parameter size. These five meta-rules are fundamental for RPC which is motivated by PageRank [15]. However, meta-rules are presented to evaluate the ranking models whether or not they are proper. RPC is a parametric design with a cubic Bézier curve of strict monotonicity. Bézier curve is a parametric curve frequently used in computer graphics that uses Bernstein polynomial as a basis to model a smooth curve and nonlinear regression. The five meta-rules are guidance for the ranking functions as constraints. RPC is visualized as graphical shapes. RPC requires a full list of attributes because it cannot deal with the missing value fields and cannot work with partial lists. For example, RPC on journals ranking removed some journals with missing data (i.e., 58 out of 451), as RPC cannot deal with missing data.

3. Unsupervised Ranking based on Magnetic Properties and Correlation Coefficient

In this section, we discuss how URMC works. URMC takes attributes of the dataset as input and returns weight for each of the attribute in the dataset as output. At the end of this section, we discuss how attributes' weights are used to rank the objects.

3.1. General Structure and Process

URMC clusters the attributes into similar groups and updates the weight of attributes that can be used to rank the objects. Figure 1 depicts the high-level workflow of our approach. URMC algorithm takes attributes of a dataset and assigns each attribute to a positive or negative cluster with weights. This is done by using the correlation coefficients between all possible pairs of attributes. Initially, all the attributes are set in positive cluster with weight 0. If the correlation coefficient between two attributes is negative, it means that they should be in different clusters. Otherwise, they should be in the same positive cluster. The algorithm is described next.

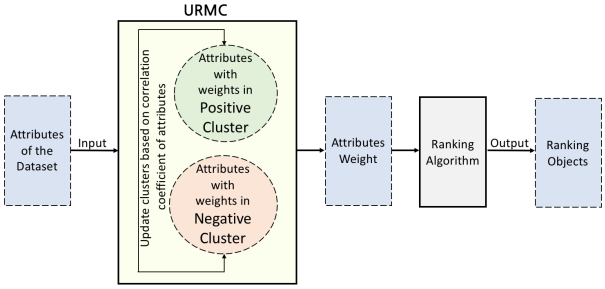

Figure 1. High-level depiction of the steps for unsupervised ranking of multi-attribute objects.

3.2. URMC Algorithm

Let X refer to a set of n objects, i.e., $X = (x_1, x_2, \ldots, x_i, \ldots, x_n)$ and each of these objects have m number of attributes. Thus, an object x_i can be represented as a set of these attribute values, i.e., $x_i = (a_{i1}, a_{i2}, \ldots, a_{ij}, \ldots, a_{im})$, where a_{ij} refers to the jth attribute value of object x_i. Again, let A_j refer to the set of jth attribute values of all the n objects, i.e., $A_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{nj})$.

The first step of ranking is to normalize the datasets. Normalizing is one of the fundamental requirements of a ranking algorithm and has been mentioned in the literature [18, 19]. In general, the range of numerical values in each attribute of a dataset widely varies. For example, in one of the evaluation dataset (i.e., the journal ranking dataset), the attribute 'Total Cites' ranges from 28851 to 105 and the attribute 'Impact Factor' ranges from 9.256 to 0.176. In our approach, we normalize an attribute value of an object into percentage using the following equation:

$$a_{ij} = \frac{a_{ij}}{\max A_j} \times 100 \quad (1)$$

where a_{ij} is the jth attribute value of object x_i, and $\max A_j$ is the largest value of attribute j.

Algorithm 1 shows the pseudocode of URMC algorithm which is based on magnetic properties and correlation coefficient using Pearson (r) correlation to cluster the attributes into two clusters (i.e., positive and negative cluster of attributes) and set each attribute a weight. If the correlation coefficient is positive between two attributes, it signifies that they attract each other to be in the same cluster, otherwise they repel to be in different clusters. A comprehensive overview of URMC algorithm is shown in Figure 2 which is divided into two parts: top part with positive correlation coefficient (i.e., $P(A_i, A_j) \geq 0$) and bottom part with negative correlation coefficient (i.e., $P(A_i, A_j) < 0$) between attributes, A_i and A_j.

Algorithm 1 : URMC Algorithm.

Input:

$A_1, A_2, \ldots, A_j, \ldots, A_m$ ▷ Set of attribute values, where $A_j = (a_{1j}, a_{2j}, \ldots, a_{ij}, \ldots, a_{nj})$

Output:

$W = (w_1, w_2, \ldots, w_j, \ldots, w_m)$ ▷ Set of attribute weights, where w_j is the weight of attribute i

Begin

1: $W \leftarrow 0$ ▷ initialize all the attributes' weight in W with 0
2: **for** $i = 1$ to m **do**
3: **for** $j = i + 1$ to m **do**
4: **if** $P(A_i, A_j) >= 0$ **then** ▷ $P(A_i, A_j)$ is the correlation coefficient between attribute A_i and attribute A_j
5: **if** $(w_i >= 0$ && $w_j >= 0)$ **then**
6: $w_i \leftarrow w_i + P(A_i, Aj)$
7: $w_j \leftarrow w_j + P(A_i, Aj)$
8: **else if** $(w_i < 0$ && $w_j < 0)$ **then**
9: $w_i \leftarrow w_i - P(A_i, Aj)$
10: $w_j \leftarrow w_j - P(A_i, Aj)$
11: **else if** $w_i >= 0$ && $w_j < 0$ **then**
12: $w_i \leftarrow w_i - P(A_i, A_j)$
13: $w_j \leftarrow w_j + P(A_i, A_j)$
14: **else**
15: $w_i \leftarrow w_i + P(A_i, A_j)$
16: $w_j \leftarrow w_j - P(A_i, A_j)$
17: **end if**
18: **else**
19: **if** $(w_i >= 0$ && $w_j >= 0)$ **then**
20: **if** $w_i < w_j$ **then**
21: $w_i \leftarrow w_i + P(A_i, A_j)$
22: $w_j \leftarrow w_j - P(A_i, A_j)$
23: **else**
24: $w_i \leftarrow w_i - P(A_i, A_j)$
25: $w_j \leftarrow w_j + P(A_i, A_j)$
26: **end if**
27: **else if** $w_i < 0$ && $w_j < 0$ **then**
28: **if** $w_i < w_j$ **then**
29: $w_i \leftarrow w_i + P(A_i, A_j)$
30: $w_j \leftarrow w_j - P(A_i, A_j)$
31: **else**
32: $w_i \leftarrow w_i - P(A_i, A_j)$
33: $w_j \leftarrow w_j + P(A_i, A_j)$
34: **end if**
35: **else if** $w_i >= 0$ && $w_j < 0$ **then**
36: $w_i \leftarrow w_i - P(A_i, A_j)$
37: $w_j \leftarrow w_j + P(A_i, A_j)$
38: **else**
39: $w_i \leftarrow w_i + P(A_i, A_j)$
40: $w_j \leftarrow w_j - P(A_i, A_j)$
41: **end if**
42: **end if**
43: **end for**
44: **end for**

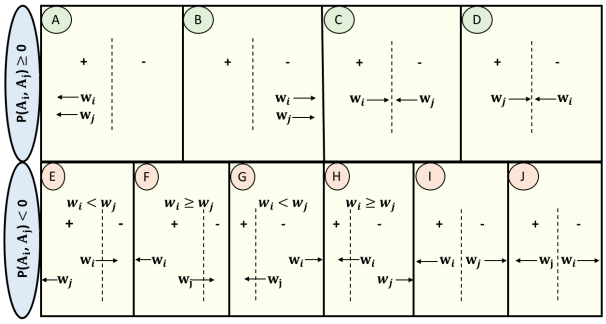

Figure 2. A comprehensive overview of URMC algorithm.

Cell A to D represent the top part with positive correlation coefficient between the attributes (Line 4-18, Algorithm 1). Initially, all the attributes are set in positive cluster with weight 0. When two attributes are in the same cluster (either positive or negative), positive correlation coefficient between the two attributes means that they attract each other to be in the same cluster with more weights. Now, if the correlation coefficient between two attributes is positive and they are in different clusters, it means that they attract each other to bring the other in its own cluster.

Cell A shows that if attributes w_i and w_j are in the positive cluster and their correlation coefficient is positive, then they should be in the positive cluster and their weight will be updated by adding the correlation coefficient to their previous weights. This represents the concept that both attributes attract each other to be more positive if they were in the positive cluster and their correlation coefficient is positive (Line 6-8, Algorithm 1). Cell B shows that if attributes w_i and w_j are in the negative cluster and their correlation coefficient is positive, then they should be in the negative cluster and their weight will be updated by subtracting the correlation coefficient from their previous weights. This shows that both attributes attract each other to be more negative if they were in the negative cluster and their correlation coefficient is positive (Line 9-11, Algorithm 1).

Cell C shows that if attribute w_i is in the positive cluster and attribute w_j is in the negative cluster and their correlation coefficient is positive, then w_i attracts w_j to be in the positive cluster and w_j attracts w_i to be in the negative cluster. Thus, the weight of w_i will be updated by subtracting the correlation coefficient from its previous weight. And the weight of w_j will be updated by adding the correlation coefficient to its previous weight (Line 12-14, Algorithm 1). Cell D shows that if attribute w_i is in the negative cluster and attribute w_j is in the positive cluster and their correlation coefficient is positive, then w_i attracts w_j to be in the

negative cluster and w_j attracts w_i to be in the positive cluster. Thus, the weight of w_i will be updated by adding the correlation coefficient to its previous weight. And the weight of w_j will be updated by subtracting the correlation coefficient from its previous weight (Line 15-17, Algorithm 1).

On the other hand, cell E to J represent the bottom part with negative correlation coefficient between the attributes (Line 20-41, Algorithm 1). In this part, since the correlation coefficient between two attributes is negative, it means that the two attributes repulse each other to be in different clusters.

Here, cell E shows that if attributes w_i and w_j are in the positive cluster and the weight of w_i is less than the weight of w_j (i.e., $w_i < w_j$) and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different clusters. Thus, the weight of w_i will be updated by adding the correlation coefficient to its previous weight. As the correlation coefficient is negative, adding it to the previous weight of w_i will shift w_i towards the negative cluster. And the weight of w_j will be updated by subtracting the correlation coefficient from its previous weight. Again, as the correlation coefficient is negative, subtracting it from the previous weight of w_j will move w_j towards more positive side (Line 21-23, Algorithm 1). Cell F shows that if attributes w_i and w_j are in the positive cluster and the weight of w_i is greater than or equal to the weight of w_j (i.e., $w_i \geq w_j$) and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different clusters. Thus, the weight of w_i will be updated by subtracting the correlation coefficient from its previous weight. And the weight of w_j will be updated by adding the correlation coefficient to its previous weight (Line 24-26, Algorithm 1).

Cell G shows that if attributes w_i and w_j are in the negative cluster and the weight of w_i is less than the weight of w_j (i.e., $w_i < w_j$) and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different clusters. Thus, the weight of w_i will be updated by adding the correlation coefficient to its previous weight. And the weight of w_j will be updated by subtracting the correlation coefficient from its previous weight (Line 28-31, Algorithm 1). Cell H shows that if attributes w_i and w_j are in the negative cluster and the weight of w_i is greater than or equal to the weight of w_j (i.e., $w_i \geq w_j$) and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different clusters. Thus, the weight of w_i will be updated by subtracting the correlation coefficient from its previous weight. And the weight of w_j will be updated by adding the correlation coefficient to its previous weight (Line 32-34, Algorithm 1).

Cell I shows that if attribute w_i is in the positive cluster and attribute w_j is in the negative cluster and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different cluster. Thus, the weight of w_i will be updated by subtracting the negative correlation coefficient from its previous weight. And the weight of w_j will be updated by adding the correlation coefficient to its previous weight (Line 36-38, Algorithm 1). It means that w_i and w_j will move towards more positive and more negative side of the cluster, respectively. Cell J shows that if attribute w_i is in the negative cluster and attribute w_j is in the positive cluster and their correlation coefficient is negative, then w_i and w_j repulse each other to be in different cluster. Thus, the weight of w_i will be updated by adding the negative correlation coefficient to its previous weight. The weight of w_j will be updated by subtracting the negative correlation coefficient from its previous weight (Line 39-41, Algorithm 1). It means that w_i and w_j will move towards more negative and more positive side of the cluster, respectively.

3.3. Ranking Algorithm

We mentioned in Section 3.2 that an object x_i can be represented as a set of attribute values, i.e., $x_i = (a_{i1}, a_{i2}, \ldots, a_{ij}, \ldots, a_{im})$, where a_{ij} refers to the jth attribute value of object x_i. Again the output of the URMC algorithm are the weights of each of the m attributes, i.e., $W = (w_1, w_2, \ldots, w_j, \ldots, w_m)$, where w_j is the weight of attribute j. Based on these notations, we compute the ranking score (we call it URMC score) of an object x_i using the following equation:

$$\begin{aligned}\text{URMC score of } x_i = w_1 \times a_{i1} + w_2 \times a_{i2} + \\ \ldots + w_j \times a_{ij} \ldots + w_m \times a_{im}\end{aligned} \quad (2)$$

Based on Equation 2, we compute the URMC scores for all the n objects and sort the objects by these scores in descending order to get the ranking order of the objects.

4. A Walk-Through Example

Suppose we have eight countries (i.e., objects) with four attributes which include gross domestic product (GDP), life expectancy at birth (LEB), infant mortality rate (IMR), and tuberculosis (Tub) as shown in Table 1[1].

The first step of ranking is to normalize the dataset as mentioned in Section 3.2 so that they are in the same quantity dimensions based on Equation 1. The results of the normalization are shown in Table 2.

[1]This is part of one of the evaluation datasets called Life Qualities of Countries (LQC) dataset.

Table 1. Life Quality of 8 Countries

Country	GDP	LEB	IMR	Tub
Finland	30469	79.09	3	3
France	29644	80.47	6	4
Germany	30496	79.48	3	4
Ireland	38058	79.4	6	4
Italy	27750	81.18	3	4
Spain	27270	80.28	13	4
UK	31580	79.3	6	5
USA	41674	77.93	2	7

Table 2. Percentage normalized

Country	GDP	LEB	IMR	Tub
Finland	73.11	97.43	23.08	42.86
France	71.13	99.13	46.15	57.14
Germany	73.18	97.91	23.08	57.14
Ireland	91.32	97.81	46.15	57.14
Italy	66.59	100	23.08	57.14
Spain	65.44	98.89	100	57.14
UK	75.78	97.68	46.15	71.43
USA	100	96	15.38	100

Table 3. Pearson's correlation coefficient between attributes

Attribute	GDP	LEB	IMR	Tub
GDP	1.00	-0.80	-0.39	0.70
LEB	-0.80	1.00	0.36	-0.60
IMR	-0.39	0.36	1.00	-0.23
Tub	0.70	-0.60	-0.23	1.00

Table 4. Weight of the attributes

Attribute	Weight of attributes
GDP	1.90
LEB	-1.76
IMR	-0.99
Tub	1.53

Table 5. Ranking result of the eight countries

Country	URMC Score	URMC Order
USA	158.97	1
Ireland	43.56	2
UK	36.13	3
Germany	31.52	4
Italy	15.33	5
Finland	10.16	6
France	2.87	7
Spain	-60.28	8

In URMC algorithm, we use Pearson's correlation coefficients (r) between attributes shown in Table 3. The next step of URMC algorithm is to compute the weight of each attribute (shown in Table 4). For example, to compute the weight of GDP, Algorithm 1 does the followings:

Initially, GDP is set in the positive cluster with weight 0. As cell F in Figure 2 shows that if attributes GDP and LEB are in the positive cluster and the weight of GDP is equal to that of LEB and their correlation coefficient is negative (i.e., -0.80), then GDP and LEB repulse each other to be in different clusters. Thus, the weight of GDP will be updated by subtracting the correlation coefficient of GDP with LEB from its previous weight (i.e., GDP = 0 - (-0.80) = 0.80).

Again, both GDP (with weight 0.80) and IMR (with initial weight 0) are in the positive cluster, the weight of GDP is greater than that of IMR and their correlation coefficient is negative (i.e., -0.39) means that Algorithm 1 will use the computation of cell F in Figure 2. Thus, the weight of GDP will be updated by subtracting the correlation coefficient of GDP with IMR from its previous weight (i.e., GDP = 0.80 - (-0.39) = 1.2).

Finally, as cell A in Figure 2 shows that if attributes GDP (with weight 1.2) and Tub (with initial weight 0) are in the positive cluster and their correlation coefficient is positive (i.e., 0.70), then GDP and Tub attract each other to be in the same cluster with more weight. Thus, the weight of GDP will be updated by adding the correlation coefficient of GDP with Tub to its previous weight (i.e., GDP = 1.2 + (0.70) = 1.9).

Next, we compute the ranking scores (i.e., URMC scores) for all the eight countries using Equation 2. For example, the ranking score of *Finland* using Equation 2, Table 2 and 4 can be computed as follows:

Ranking score of *Finland* $=$ *(weight of GDP* $\times$ *Percentage of GDP for Finland)* $+$ *(weight of LEB* $\times$ *percentage of LEB for Finland)* $+$ *(weight of IMR* $\times$ *percentage of IMR for Finland)* $+$ *(weight of Tub* $\times$ *percentage of Tub for Finland)*
$= (1.90 \times 73.11) + (-1.76 \times 97.43) + (-0.99 \times 23.08) + (1.53 \times 42.86) = 10.16$

Sorting the eight countries by these ranking scores in descending order will be the ranking order of the eight countries as shown in Table 5.

5. Experimental Result and Discussion

To evaluate and compare our algorithm to the RPC algorithm [1], one of the state-of-the-art algorithms on the task, we used the following three datasets: Journals, Webometrics, and Life Qualities of Countries (LQC).

RPC algorithm [1] also used these datasets to evaluate their algorithm.

5.1. URMC and RPC on Journals Ranking (JR) Dataset

This dataset presents data about academic journals in the sciences and social sciences and is available from the Web of Knowledge [2] which is associated with Thomson Reuters. RPC algorithm [1] used JCR2012 version of this dataset. Though this dataset has eight attributes, authors of the RPC algorithm select only five out of the eight attributes to rank the journals.

We compare URMC with RPC in two different settings. First, we compare URMC with RPC using only the five attributes selected by RPC. Second, we use all the eight attributes provided in the main dataset to see how URMC does without selecting attributes compared to RPC with selected attributes.

5.1.1. Experiment on Five Attributes In this experimental setting, we use the same five attributes (shown in Table 6) that RPC used.

The Pearson correlation coefficients between URMC's and RPC's ranking orders and scores are 0.9987 and 0.9829, respectively. As there is no ground truth for this dataset, these very strong correlation coefficients show that URMC is very comparable with RPC. In Table 6, we show the top and the bottom five journals out of the 393 journals with five attributes ranked by URMC and their corresponding ranking by RPC on this dataset.

Figure 3 shows how attribute weights significance change when attributes are compared with each other using Algorithm 1. Initially, weights of all the attributes are set to 0. Numbers on the x-axis represent the attribute that is compared with the rest of the attributes. For example, 1 on the x-axis shows the weights of the attributes after comparing attribute one with the rest four attributes. Similarly, 2 on the x-axis shows the weights of the attributes after comparing attribute two with the rest three attributes, and so on. All these procedures are significant because they show the distinctiveness between attributes and how the weights get more spread or separated by each step. The Pearson correlation coefficient represents either the strength or weakness of the relationship between two attributes. URMC significantly (T-test, the p-value is < 0.00001) outperforms RPC on JR dataset with five attributes.

[2] http://wokinfo.com/

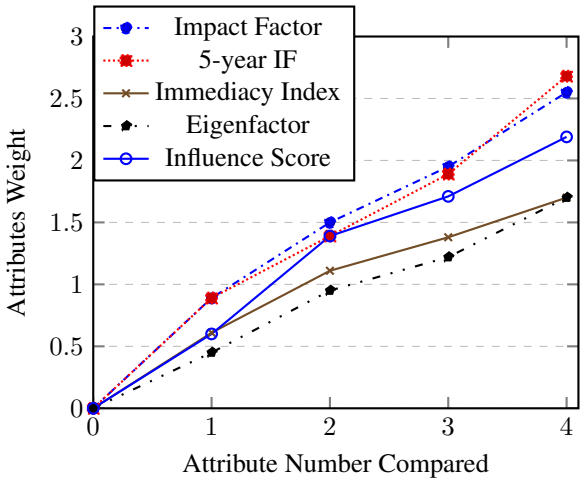

Figure 3. How attribute weights change using Algorithm 1 on the JR dataset with five attributes.

5.1.2. Experiment on Eight Attributes In this experimental setting, we use all the eight attributes (shown in Table 7) present in the main dataset.

The Pearson correlation coefficients between URMC's and RPC's ranking orders and scores are 0.9805 and 0.9776, respectively. Here, RPC uses only the five selected attributes. These very strong correlations indicate that URMC's ranking, without selecting any attribute, is comparable to that of RPC which uses only the selected attributes. In Table 7, we show the top and the bottom five journals out of the 393 journals ranked by URMC (with all the eight attributes) and their corresponding ranking by RPC (with five selected attributes) on this dataset.

Figure 4 shows how attribute weights change when attributes are compared with each other using Algorithm 1. The figure also shows that as the number of attributes compared with increases, from one (1) to seven (7), the weight of each attribute gets more distinctive.

Furthermore, 'Cited Half-life', one of the eight attributes of the main dataset used by URMC algorithm, has 16 missing values. The very strong correlation coefficients between URMC's and RPC's ranking orders and scores suggest that URMC algorithm is effective even with missing value attributes. URMC significantly (T-test, the p-value is < 0.00001) outperforms RPC on JR dataset with eight attributes.

5.2. URMC and RPC on Webometrics Dataset

This dataset presents data about the top 500 world universities and is available from the Webometrics Ranking of World Universities [3] which is associated

[3] http://webometrics.info/

Table 6. Results showing the top and bottom five journals on the JR dataset with five attributes.

Journal Title	Attributes					RPC		URMC	
	Impact Factor	5 -year Impact Factor	Immediacy Index	Eigenfactor Score	Influence Score	Score	Order	Score	Order
IEEE T PATTERN ANAL	4.795	6.144	0.625	0.05237	3.235	1	**1**	705.6233	**1**
ENTERP INF SYST UK	9.256	4.771	2.682	0.00173	0.907	0.95051	**2**	638.1533	**2**
MIS QUART	4.659	7.474	0.705	0.01036	3.077	0.91046	**4**	631.9720	**3**
J STAT SOFTW	4.91	5.907	0.753	0.01744	3.314	0.91622	**3**	623.7083	**4**
ACM COMPUT SURV	3.543	7.854	0.421	0.0064	4.097	0.90923	**5**	612.8080	**5**
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
NEURAL NETW WORLD	0.362	0.381	0.029	0.00033	0.082	0.00685	**389**	30.2121	**389**
J INF SCI ENG	0.299	0.326	0.03	0.00095	0.088	0.00625	**390**	28.7437	**390**
INT J SOFTW ENG KNOW	0.295	0.336	0.03	0.00044	0.107	0.00550	**391**	28.5385	**391**
J COMPUT SYS SC INT	0.249	0.242	0.078	0.00066	0.08	0.00104	**392**	26.1747	**392**
COMPUT INFORM	0.254	0.305	0.06	0.00031	0.065	0.00000	**393**	25.5589	**393**

Table 7. Results showing the top and bottom five journals on the JR dataset with eight attributes. Here, RPC uses only the five underlined attributes.

Journal Title	Attributes								RPC with 5 Attributes		URMC with 8 Attributes	
	Total Cites	Impact Factor	5-Year Impact Factor	Immediacy Index	Articles	Cited Half-life	Eigenfactor Score	Influence Score	Score	Order	Score	Order
IEEE T PATTERN ANAL	24947	4.795	6.144	0.625	192	10	0.00054	3.235	1	**1**	786.8565	**1**
MIS QUART	7277	4.659	7.474	0.705	61	4.5	0.00324	3.077	0.91046	**4**	697.4675	**2**
ENTERP INF SYST UK	579	9.256	4.771	2.682	22	4.5	0.00459	0.907	0.95051	**2**	693.6994	**3**
ACM COMPUT SURV	2907	3.543	7.854	0.421	38	9.6	0.0064	4.097	0.90923	**5**	652.7896	**4**
J STAT SOFTW	2629	4.91	5.907	0.753	77	5	0.00005	3.314	0.91622	**3**	646.3808	**5**
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
ADV COMPUT	152	0.389	0.452	0.043	23	9.6	0.00029	0.195	0.02148	**275**	1.9415	**389**
PROBL INFORM TRANSM	445	0.298	0.387	0.062	32	10	0.04144	0.264	0.02497	**371**	1.7215	**390**
INT J COMPUT GEOM AP	215	0.176	0.253	0	22	7.4	0.00427	0.272	0.01233	**386**	-0.3940	**391**
J EXP THEOR ARTIF IN	182	0.317	0.57	0	29	10	0.00201	0.186	0.02159	**374**	-0.4197	**392**
INT J ARTIF INTELL T	263	0.25	0.453	0.054	56	10	0.00062	0.174	0.01809	**380**	-0.6728	**393**

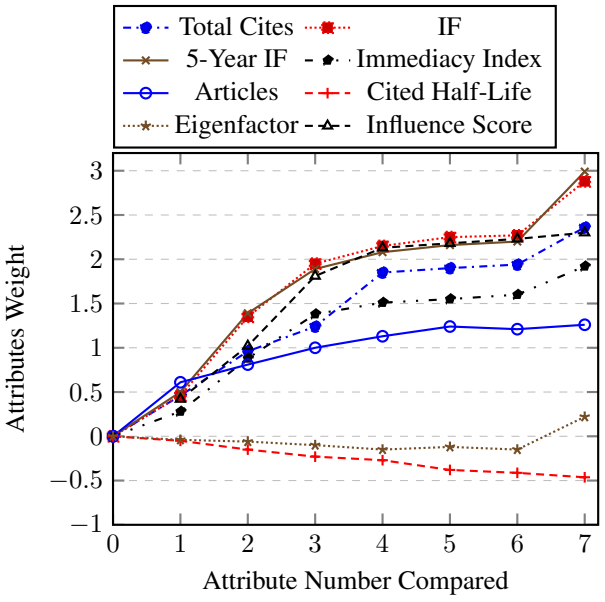

Figure 4. How attribute weights change using Algorithm 1 on the JR dataset with eight attributes.

with Cybermetrics Lab, a research group belonging to the Consejo Superior de Investigaciones Científicas (CSIC), the largest public research body in Spain. As this dataset provides a ranking order, we compare URMC with this ranking order as well as with RPC.

The Pearson correlation coefficients between URMC's and RPC's ranking orders and scores are 0.9704 and 0.9768, respectively. Again, these very strong correlation coefficients show that URMC is very comparable with RPC.

In Table 8, the dataset shows the top and the bottom five universities out of 500 world universities ranked by URMC and their corresponding ranking by RPC and Webometrics. Figure 5 shows how attribute weights change when attributes are compared with each other using Algorithm 1. The Pearson correlation coefficient between URMC's and Webometrics' ranking orders is 0.87. Again, the Pearson correlation coefficient between RPC's and Webometrics' ranking orders is 0.89, which shows that URMC is comparable to RPC based on Webometrics' ranking orders.

5.3. URMC and RPC on Life Qualities of Countries (LQC) Dataset

This dataset presents data about life qualities of countries and is available from GAPMINDER [4]. RPC

[4]http://www.gapminder.org/

Table 8. Results showing the top and bottom five universities on the Webometrics dataset.

University Name	Attributes				RPC		Webometrics	URMC	
	Presence	Visibility	Openness	Excellence	Score	Order	Order	Score	Order
Massachusetts Institute of Technology	1559	1466	678	28	0.98357	**2**	**3**	457.2426	**1**
University of Illinois Urbana Champaign	1587	1060	85	369	0.88908	**7**	**20**	371.6282	**2**
Harvard University	1573	734	1074	138	0.91362	**4**	**1**	349.9087	**3**
University of British Columbia	1091	1132	159	377	0.82897	**25**	**22**	349.4531	**4**
Stanford University	1559	124	1667	774	1.00	**1**	**2**	337.9593	**5**
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Nankai University	2	11	122	7	0.01270	**498**	**433**	10.3521	**496**
University Politechnica of Bucharest	25	8	91	5	0.01573	**496**	**490**	9.6224	**497**
Cardiff University	51	1	1	1	0.01286	**497**	**484**	4.5788	**498**
Universitss Paris	5	3	30	10	0.00254	**499**	**500**	3.6348	**499**
Wright State University	1	1	25	3	0.00	**500**	**460**	2.0516	**500**

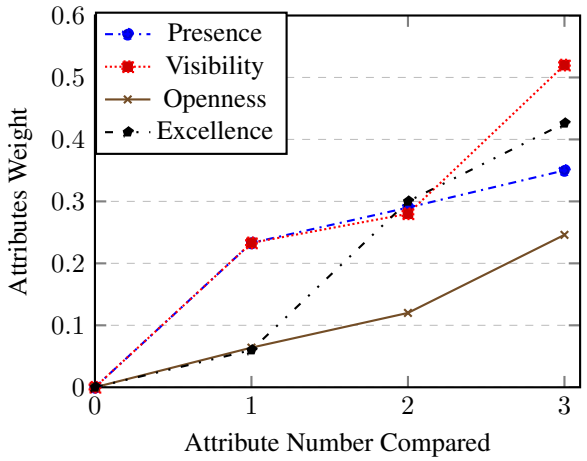

Figure 5. How attribute weights change using Algorithm 1 on the Webometrics dataset.

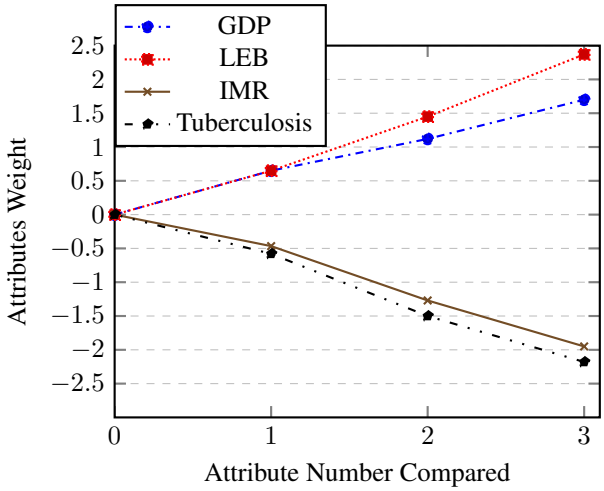

Figure 6. How attribute weights change using Algorithm 1 on the LQC dataset.

used a fraction of this dataset to rank 171 countries based on four attributes which include life expectancy at gross domestic product (GDP), life expectancy at birth (LEB), infant mortality rate (IMR), and tuberculosis (Tub). To fairly compare with RPC, we also use the same fraction of the dataset and attributes mentioned in [20].

The Pearson correlation coefficients between URMC's and RPC's ranking orders and scores are 0.9976 and 0.9897, respectively. These very strong correlation coefficients indicate that URMC's ranking is strongly comparable to that of RPC.

In Table 9, we show the top and the bottom five countries out of the 171 countries ranked by URMC and their corresponding ranking by RPC on this dataset. Figure 6 shows how attribute weights change when attributes are compared with each other using Algorithm 1. The figure also shows that as the number of attributes compared with increases, from one (1) to three (3), the weight of each attribute gets more distinctive. URMC significantly (T-test, the p-value is < 0.00001) outperforms RPC on LQS dataset.

6. Conclusion

In this paper, we proposed an unsupervised ranking algorithm for multi-attribute numerical objects by incorporating correlation coefficients between attribute values using the concept of magnetic properties. We showed how the proposed algorithm computed more distinctive weights for attributes to rank the objects. Unlike other algorithms, our proposed URMC algorithm can deal with objects' missing attribute values. We showed that URMC, which does not select attributes, is comparable to the algorithm that selects some important attributes to rank multi-attribute numerical objects. Experimental results on three different datasets confirmed that URMC is strongly comparable to state-of-the-art unsupervised ranking algorithms that cannot deal with attributes with missing values and needs to select attributes before ranking.

One of the important future works on this task could be to rank numerical and nonnumerical multi-attribute

Table 9. Results showing the top and bottom five countries on the LQC dataset.

Country	Attributes				RPC		URMC	
	GDP	LEB	IMR	Tuberculosis	Score	Order	Score	Order
Luxembourg	70014	79.56	6	4	1	**1**	391.2335	**1**
Norway	47551	80.29	3	3	0.89098	**2**	341.5337	**2**
Singapore	41479	79.627	12	2	0.85184	**4**	322.0661	**3**
Iceland	35630	81.43	2	2	0.81824	**7**	317.6761	**4**
United States of American	41674	77.93	2	7	0.84922	**5**	315.6420	**5**
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
Congo, Dem. Rep.	330	47.629	183	129	0.17951	**163**	-116.6396	**167**
Angola	3533	45.523	119	154	0.19208	**162**	-118.5150	**168**
Afghanistan	874	42.88	76	165	0.19725	**161**	-127.3279	**169**
Sierra Leone	790	46.365	219	160	0.12698	**168**	-176.7179	**170**
Swaziland	4384	44.99	422	110	0.00000	**171**	-199.3435	**171**

objects (i.e., text). The challenge with ranking these multi-attribute objects would be to convert texts into their representative numerical values. Another future work could be to find intra-attribute weight by analyzing the attribute values without comparing the attribute with other attributes.

References

[1] C. G. Li, X. Mei and B. G. Hu, "Unsupervised ranking of multi-attribute objects based on principal curves," 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, 2016, pp. 1526-1527.

[2] Zhao, Z. and Liu, H., 2007, June. Spectral feature selection for supervised and unsupervised learning. In Proceedings of the 24th international conference on Machine learning (pp. 1151-1157). ACM.

[3] Li, C.G., Mei, X. and Hu, B.G., 2014, December. Two-Phase Attribute Ordering for Unsupervised Ranking of Multi-Attribute Objects. In Data Mining Workshop (ICDMW), 2014 IEEE International Conference on (pp. 175-182). IEEE.

[4] Ellen Marshall. Scatterplots and correlation in SPSS. Available at: https://www.sheffield.ac.uk/polopoly_fs/1.531428!/file/MASHScatterplot_correlation_SPSS.pdf, (downloaded 10/24/2017)

[5] Liu, Y.T., Liu, T.Y., Qin, T., Ma, Z.M. and Li, H., 2007, May. Supervised rank aggregation. In Proceedings of the 16th international conference on World Wide Web (pp. 481-490). ACM.

[6] Wei, F., Li, W. and Liu, S., 2010. iRANK: A rank-learn-combine framework for unsupervised ensemble ranking. Journal of the Association for Information Science and Technology, 61(6), pp.1232-1243.

[7] Klementiev, A., Roth, D. and Small, K., 2007, September. An unsupervised learning algorithm for rank aggregation. In European Conference on Machine Learning (pp. 616-623). Springer, Berlin, Heidelberg.

[8] Klementiev, A., Roth, D. and Small, K., 2008. A framework for unsupervised rank aggregation. Urbana, 51, p.61801.

[9] Klementiev, A., Roth, D. and Small, K., 2008, July. Unsupervised rank aggregation with distance-based models. In Proceedings of the 25th international conference on Machine learning (pp. 472-479). ACM.

[10] Järvelin, K. and Kekäläinen, J., 2000, July. IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 41-48). ACM.

[11] Baeza-Yates, R. and Ribeiro-Neto, B., 1999. Modern information retrieval (Vol. 463). New York: ACM press.

[12] Lin, S., 2010. Rank aggregation methods. Wiley Interdisciplinary Reviews: Computational Statistics, 2(5), pp.555-570.

[13] Volkovs, M.N. and Zemel, R.S., 2014. New learning methods for supervised and unsupervised preference aggregation. The Journal of Machine Learning Research, 15(1), pp.1135-1176.

[14] Klementiev, A., Roth, D., Small, K. and Titov, I., 2009. Unsupervised rank aggregation with domain-specific expertise. Urbana, 51, p.61801.

[15] Brin, S. and Page, L., 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18), pp.3825-3833.

[16] Cai, D., Zhang, C. and He, X., 2010, July. Unsupervised feature selection for multi-cluster data. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 333-342). ACM.

[17] Liu, F. and Liu, X., 2012, July. Unsupervised feature selection for multi-cluster data via smooth distributed score. In International Conference on Intelligent Computing (pp. 74-79). Springer, Berlin, Heidelberg.

[18] Ginevičius, R., 2008. Normalization of quantities of various dimensions. Journal of business economics and management, 9(1), pp.79-86.

[19] Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. and Li, H., 2007, June. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning (pp. 129-136). ACM.

[20] Zinovyev, A.N.D.R.E.I. and Gorban, A.N., 2010. Nonlinear quality of life index. arXiv preprint arXiv:1008.4063.