Communication Papers of the Federated Conference on Computer Science and Information Systems pp. 13–19
DOI: 10.15439/2019F95
ISSN 2300-5963 ACSIS, Vol. 20

Selecting representatives

László Aszalós
Faculty of Informatics at University of Debrecen,
26 Kassai út H4028 Debrecen , Hungary
Email: aszalos.laszlo@inf.unideb.hu

Dávid Nagy
Faculty of Informatics at University of Debrecen,
26 Kassai út H4028 Debrecen , Hungary
Email: nagy.david@inf.unideb.hu

***Abstract*—We use representatives to reduce complexity in many areas of life. Clusters are often replaced with their centre, and then these representatives are used to classify new objects. If the objects are described as a vector of real numbers, then the centre can be easily calculated. However, this method is unusable if only a similarity relation is given instead of coordinates of the object or the distances between the objects. Google can filter and rank relevant pages for a particular question; and here we follow a similar approach. The difference is that we have an undirected graph while the PageRank algorithm uses a directed one. In this article we show what conditions we set for our own ranking system. Following the description of the details of this method we demonstrate that it satisfies our criteria and how it selects the (mathematically proven) most typical elements of each cluster. Finally, we apply this method on several partitions of the natural numbers and on non-transitive tolerance relations to present the representatives of the numbers.**

I. Introduction

INSTEAD of examining the entire population, polls usually only survey a small sample. This can be done because the results obtained are very close to what we would get by examining the entire population. However, the sample should be selected carefully. Almost everyone knows the concept of the *representative sample*, but only few know exactly what it means. Many people think that the larger the sample, the better; which is not true. The sample is representative in some respects, i.e., the specific properties are as similar in the sample as in the entire population. The sample can be representative in one aspect, while not representative in another. There are various standard methods for determining the sample.

If the population is significantly inhomogeneous, i.e. it has high variability according to the survey, then the stratified (random) sampling is used. In this case, the population is divided into several sub-populations (strata), where these sub-populations are homogeneous according to the examined criteria. From a homogeneous strata, we can randomly select the individuals to be sampled (i.e., the representative of the group), typically in proportion to the size of the group.

If we can represent an object with a vector of numbers, we can consider the difference of vectors belonging to each object, where this difference/distance usually meets the requirements of metrics. Using this distance function, many clustering methods have been developed over the last sixty years. The most well-known *k-means method* replaces a cluster with its centre (one representative). The *k-medoids algorithm* is a version of this k-means method, and it replaces the cluster with the sample element closest to the cluster centre. The *CURE method* (clustering using representatives) goes one step further, replacing non-ellipsoid clusters with maximum c sample elements. The most common use of k-means (or its enhancements) is the *k-nearest neighbours* (k-NN) classification algorithm, where newly added objects must be categorized into an existing cluster/class. Since comparing the new elements with all stored elements in a large database is a time costly task, by replacing the elements of the clusters with some of their representatives we can significantly reduce the complexity of the classification of new elements.

Polls can not ask too many questions from a person because their patience is finite. However, there are cases where we leave behind a lot of information, think for example our medical cards, our data stored at different kinds of service providers, or our digital footprint on the social network. In these cases, it is not worth transforming this information into a unified form in order to be able to define the differences between the data of objects. It is much easier to directly decide for two given objects whether they are similar or not.

In this article, we present a mathematical method which—having an existing partition and similarity relation–determines which is the most typical object in a given cluster, i.e. which one can be considered *representative*. We assign a real number, a *rank* to each of these objects, and the highest-ranking object in each cluster becomes the representative of the cluster.

In the next section, we present the requirements we expect from the rank of the objects. In the Section III we present the power method and how it can be used for our purposes. Next we demonstrate the results of our method through two special relations, and how matches our expectations. Finally we conclude our results.

II. The properties of ranking

In the following, we identify how to describe relations using signed graphs. Each element in a relation will be a vertex of the graph and two vertices are connected with an edge if and only if their two corresponding elements are in relation.

In case of graphs, we can speak of the distance between two vertices (as the shortest path between the two vertices), but it carries much less information than the difference of two large vectors. Therefore, the similarity information should

The publication is supported by the EFOP-3.6.1-16-2016-00022 and EFOP-3.6.3-VEKOP-16-2017-00002 projects. These projects are co-financed by the European Union and the European Social Fund.

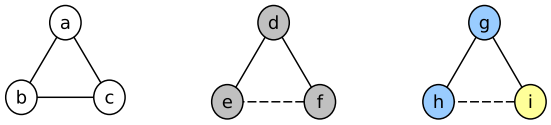

Fig. 1. Simple ranking problems

already be included in the graph, so the graph will correspond to a similarity relation. As we usually have partial similarity relations in practice, we will have edges in the graph that denote the similarity and we will have edges that denote the dissimilarity. The partiality is represented by missing edges.

For example, links between individual websites or citations between scientific articles define a directed graph, i.e. a partial similarity relation, but there is no representation of *dislike* i.e. dissimilarity.

Google's PageRank algorithm [2] is a great example of a ranking system on directed graphs. Considering the web pages (vertices) and the links between them (edges) as a directed graph, the boundary distribution of the random walk on the graph gives the rank of each page. For example, if the web page p is more likely to be accessed than the web page q, then the rank of p will be higher than that of q and will therefore be ranked higher in the hit list. Here, if a page with a low rank refers to a page with a high rank, or a novice author refers to a well-known author in his article, it raises the rank of the page/author to a higher rank, but—through a non-symmetrical relation—this reference has no effect on the rank of the page/author with the lower rank.

However, if the graph is not directed, the edge between the two vertices will affect the rank of both vertices. Since the similarity relation is a tolerance relation (reflexive, symmetric, but not necessarily transitive), the associated graph is not directed.

Let's see our (naive) expectations of a ranking method. In Fig. 1 on the left, we have three vertices (a, b and c) in a common cluster. In this figure, a cluster is represented by vertices of the same colour, while the similarity of vertices is denoted by a solid, and the difference by a dashed line. Because in this sub-graph each element is similar to each other, we expect the same rankings due to symmetry. In the middle graph of Fig. 1—where d, e and f are in a common cluster—a difference appears. This graph is called a *minimal frustrated graph*, because there is no such partition of vertices where similar elements are common, and different elements are clustered separately. In this graph, vertex d has only similar vertices, while vertices e and f both have similar and also different vertices. The fact that an object differs from an object in its own cluster reduces the rank of the object/vertex and thus the chance of being a representative of the cluster. Conversely, if an object similar to an object in its own cluster, then this increases its rank. Based on these, this cluster will be represented by vertex d because it will have the highest rank. Moreover—according to the symmetry—the rank of vertices e and f should be the same.

Finally, take the graph on the right side of the Fig. 1. Here, the vertices were divided into two clusters: $\{g,\ h\}$ and $\{i\}$. The fact that the vertices of g and i are similar, but are found in different clusters also reduces the rank of both vertices, because similarity to vertices in others clusters means *deviation* from the idealized characteristics of the group. The vertex h is similar to vertex g which belongs to h's cluster, and h is different from the vertex i belongs to another cluster. This latter also raises the rank of the vertex h and hence h becomes the representative of its own cluster. In the other cluster, the only vertex will be the representative.

Based on the examples above, the similar objects of the same cluster and dissimilar objects of other clusters can be called the *fosterer* of the object, while the similar objects of different clusters, and dissimilar objects of the same cluster can be called the *adversary* of the object. The fosterer objects help an object to become a representative, while the adversary objects prevent it this from happening.

Let's summarise what we would expect from the rankings.

- Be symmetric, that is, if two vertices have the same number of vertices of the same rank in the same type relation (fosterer or adversary), then their rank is the same.
- The rank of a particular item is immediately raised if:
  - one of its fosterer object increases in rank, or
  - one of its adversary object falls in rank, or
  - a new fosterer object appears.
- The rank of a particular item is immediately reduced if:
  - one of its adversary object increases in rank, or
  - one of its fosterer object falls in rank, or
  - a new adversary object appears.
- It does not directly change the rank of a particular item if:
  - another object that is not compared to it or is incomparable appears in any cluster, or
  - the rank of such an object changes.

We were mindful of our usage of the word *immediately*. If a new object that is dissimilar to the current object, but similar to another object of the cluster, is added to this cluster, then it raises the rank of objects that are similar to it. This will have a ripple effect on objects that are similar to objects are similar to the new object, and so on. Therefore, if we think of representing this with an algorithm, we need an iterative method that will escalate these effects step by step. On the other hand, since almost every objects are related to every objects, we need to treat the rank of all objects altogether.

III. Our ranking method

Let V denote the set of objects/vertices, and for simplicity, denote the objects with numbers: $V = \{1, 2, \ldots, n\}$. The set of the clusters means a partition. This partition is interpreted as a function—denoted by p—that assigns a number to each object, so $p : V \to \mathbb{N}$. The objects i and j are in the same cluster, if $p(i) = p(j)$; and they are in different clusters, if

$p(i) \neq p(j)$. Our similarity relation is a (possibly partial) tolerance relation T—that is reflexive, symmetric, but not necessarily transitive. In Section IV we present the numerical results of our ranking method through two partial tolerance relations.

A. Social ranking

We use similar approach as PageRank or various evaluation sites (accommodations, restaurants, marketplaces), where the rankings of individual websites, hotels, restaurants are summed up by aggregating individual ratings.

We could take the rank of each object as the difference between the number of fosterer and adversary objects, but we want to introduce a more sophisticated method. We think that there is a significant difference between the the cases where the adversary object is the representative of the other cluster, or it just a marginal object there. In the former case it decreases the rank of the current object to a greater extent. We find that a value proportional to the rank of the adversary or fosterer objects defines a a good amount by which we can decrease or increase the rank of the current object.

The rank of object i is determined by its relation to all objects. Let a_{ij} indicate the relation (fosterer or adversary) between objects i and j. We want to consider each object with its rank as weights, and so we get the following relation: $r_i = \sum_j a_{ij} \cdot r_j$ for all $i \in V$, which we want to solve for values r_i. These equations combine to $R = AR$ in matrix notation.

So that we do not have to use different values a_{ij} from task to task—depending on its size—let's introduce a constant c to normalise the values a_{ij}. This changes the previous relation as follows $r_i = \sum_j (c \cdot a_{ij}) \cdot r_j$, but it can be transformed to $r_i = c \cdot \sum_j a_{ij} \cdot r_j$, which gives us $R = cAR$. If we take the reciprocal of c to be denoted by λ, we get a known relation: $AR = \lambda R$, i.e. R is an eigenvector of the matrix A.

Taking into account the ideas from the previous chapter, the normalized value a_{ij} should be 1 for fosterer objects, -1 for adversary objects, and 0 for all other cases (where the tolerance relation is partial).

According to the much cited example, a bald man does not resemble a hairy man, although an uprooted hairline does not change a person. As a person can have up to two million hairlines, the linear model of this example is reduced to only four states. Here 1 (bald) and 4 (hairy) correspond the two end states, while 2 and 3 are two intermediate states. 1 and 2 are in a common cluster, and so are 3 and 4. Fig. 2 shows two graphs demonstrating this problem, where we denote clusters by assigning different colours to the vertices. The similarities (denoted by solid lines) are the same in both cases, but the dissimilarities (denoted by dashed lines) have changed: dissimilarity appears between the second neighbours in the latter case. Hence the first graph/relation is partial, and the second is total.

Two matrices based on the relations and partitions for each graph are:

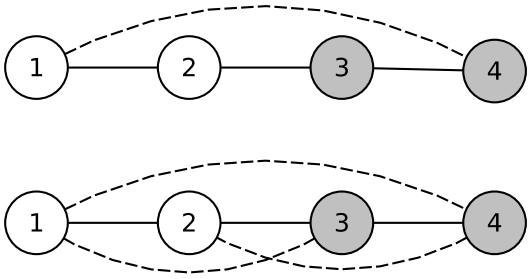

Fig. 2. Two simple symmetric linear model

$$A_1 = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

The only adversary relation is between 2 and 3, because they are similar, but are in different clusters. In the first case 1-3 and 2-4 are not comparable, so the corresponding values of the matrix are 0. In the second case these pairs of objects are dissimilar, but they are in different clusters, so these are fosterer relations.

Calculating the eigenvalues and eigenvectors for the first graph, we get:

r_1	r_2	r_3	r_4	λ
0.7071	0.7071	0.3344	0.1368	2.4142
0.5000	-0.5000	0.6770	-0.5873	-0.4142
0.0000	0.0000	-0.6230	-0.6938	2.4132
0.5000	-0.5000	-0.2041	0.3939	-0.4142

Our graph is symmetric, so we would expect, that $r_1 = r_4$ and $r_2 = r_3$. Unfortunately, none of the eigenvectors satisfy this. Therefore, we need to look for another method!

B. Power-method

The algorithm of von Mises [5] for a diagonal matrix A results in the biggest eigenvalue (with the highest absolute value), and the corresponding eigenvector. The method starts with an arbitrary vector R_0 that in our case should be $\mathbf{1} = (1, \ldots, 1)^T$. Then R_{k+1} is determined as follows: the rank vector R_k is multiplied by the matrix A and normalized as shown by (1).

$$R_{k+1} \leftarrow \frac{AR_k}{||AR_k||} \tag{1}$$

Unfortunately, this iteration is converges slowly, but it is easy to use even for large sparse matrices; which is why it is used in PageRank implementation. If $R_i \approx R_{i+1}$, the method is stopped and the values in the vector R_i are considered the rank of the objects. If matrix A has an eigenvalue that is strictly greater in magnitude than its other eigenvalues, then R_i converges.

Algorithm 1 Python implementation of our ranking method

```
def power_method(A, eps = 1e-9):
    B = A
    R = np.ones((len(A),))
    B2 = B@B
    B2 /= m = np.max(B2@R)
    while np.linalg.norm(B@R - B2@R) > eps:
        B, B2 = B2, B2@B2
        B2 /= np.max(B2@R)
    return B2
```

If this method is applied to the matrices shown in Fig. 2, then we get $R = (1, 0.414, 0.414, 1)^T$ and $R = (1, 0.618, 0.618, 1)^T$, where $r_1 = r_4$ and $r_2 = r_3$ as we expected. These values also fit to our naive ideas: for the first graph, for object 1 both objects 2 and 4 are fosterers, while for object 2 object 1 is a fosterer and object 2 is an adversary. So the expected relation $r_1 > r_2$ is fulfilled. In the second case, when there are more fosterer relations, the rank gained is also higher for objects 2 and 3.

Summarising the gives:

$$R_{k+1} \leftarrow \frac{AR_k}{||AR_k||} \approx \frac{A^{k+1}\mathbf{1}}{||A^{k+1}\mathbf{1}||}$$

If k is a power of 2, then we can then calculate the same values by repeated squaring using the following recurrence relation:

$$B_1 \leftarrow A \text{ and } B_{i+1} \leftarrow \frac{B_i B_i}{||B_i B_i||} \tag{2}$$

If $B_i\mathbf{1} \approx B_{i+1}\mathbf{1}$, then let $R \leftarrow B_i\mathbf{1}$. Not surprisingly, the two calculations give the same result.

Based on these, it is not difficult to write the ranking program in Python using the services of the Numpy package (Algorithm 1). Remark, that for Numpy, the operator `@` is the matrix multiplication operation.

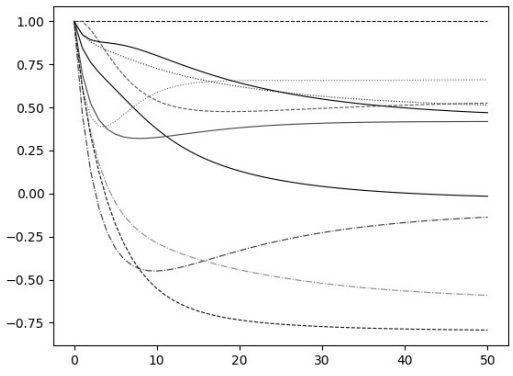

Fig. 3. Changes in the rank of the objects in a random tolerance relation

Fig. 3 shows how each value r_i changes for a random matrix A. Here, the axis x represents the number of applications of (2), while the axis y represents the current values of the ranks. Due to normalisation, the highest rank is always 1, but as we can see from the chart, the rank of the objects changes from time to time. The algorithm is terminated when the ranks cease changing.

IV. Ranks of numbers

As there is no standard similarity relations for larger databases, the various clustering/classification methods are usually tested on random graphs [6]. On the other hand, we spent a long time looking for a tolerance relation, that is **not** an equivalence relation, freely scalable, easy to understand for the reader and enables reproducible computations. Finally, we found the *common divisor* as the basis of similarity: let two numbers be similar if they have a non-trivial common divisor, i.e. $\gcd(a, b) > 1$, where $a,\ b \in \mathbb{N}^+$. Then $4T6$ and $6T9$ are fulfilled (the common divisors are 2 and 3), but $4T9$ is not, so this relation is not transitive. If we are interested in the correlation clustering of numbers $1, \ldots, n$, it can be easily formulated if $n < 111\,546\,435$, otherwise the situation become complicated [3].

In the following, the rank of each element is determined by using the optimal clustering of the set of numbers, except in the first case, where we place the numbers $1, \ldots, 12$ in a common cluster.

As we have a single cluster in the equation (2), we can replace the matrix A in the calculation with the matrix of the relation T in Fig. 4. In this matrix T, the relation between the numbers i and j is given by the j^{th} number from the i^{th} row: similar numbers are denoted by 1, dissimilar numbers by -1. 12 and 6 are similar to the multiples of 2 and 3, these are eight numbers including themselves, and different from four of them (1, 5, 7, and 11). The powers of 2 are similar to every even number (six numbers) and different from all odd number (six numbers). The powers of 3 (itself and 9) are similar to four numbers and different from eight numbers. 5 is similar to its duplicate, but there is no such number for 7 and 11 and for 1. Because in each case the numbers mentioned together are similar to the same numbers, so—according to the symmetry—their rank is the same (Table I/A). In this table the ranks are rounded to the closest hundreds, to fit on the page.

If we do the same calculation for numbers $1, \ldots, 100$ (Table I/B) then the numbers will typically increase. There are many more even numbers, which will increase the rank of even numbers, but they have the opposite effect on the rank of powers of 3. In this set, there are numbers similar to 7 or 11, so their rank increases; and there are more such numbers for 7, so its rank grows more.

A. Using optimal partition

Consider the partition of natural numbers obtained by correlation clustering [3]. The largest such cluster is the set of even numbers. This is followed by a set of odd numbers divisible by 3; next the set of numbers which are divisible

TABLE I
RANKING NUMBERS...

A) $1, \ldots, 12$ in a common cluster.

1	2	3	4	5	6	7	8	9	10	11	12
-0.62	0.97	0.06	0.97	-0.32	1.00	-0.62	0.97	0.06	0.89	-0.62	1.00

B) $1, \ldots, 100$ in a common cluster.

1	2	3	4	5	6	7	8	9	10	11	12	...
-0.55	0.99	-0.03	0.99	-0.28	1.00	-0.37	0.99	-0.03	0.95	-0.44	1.00	...

C) $1, \ldots, 12$ using optimal partition.

C_1	C_2						C_3		C_5	C_7	C_{11}
1	2	4	6	8	10	12	3	9	5	7	11
1.00	1.00	1.00	0.73	1.00	0.84	0.73	0.73	0.73	0.84	1.00	1.00

D) $1, \ldots, 15$ using optimal partition.

C_1	C_2							C_3			C_5	C_7	C_{11}	C_{13}
1	2	4	6	8	10	12	14	3	9	15	5	7	11	13
1.00	1.00	1.00	0.67	1	0.79	0.67	0.86	0.79	0.79	0.54	0.79	0.86	1.00	1.00

E) $1, \ldots, 12$ using common cluster and the weakened relation.

1	2	3	4	5	6	7	8	9	10	11	12
-0.53	1.00	0.06	0.83	-0.34	0.68	-0.53	0.70	-0.17	0.42	-0.53	0.81

F) $1, \ldots, 12$ using optimal partition and the weakened relation.

C_1	C_2						C_3		C_5	C_7	C_{11}
1	2	4	6	8	10	12	3	9	5	7	11
1.00	1.00	0.89	0.53	0.83	0.53	0.62	0.77	0.88	0.89	1.00	1.00

$$T = \begin{pmatrix}
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1
\end{pmatrix},$$

$$A = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1
\end{pmatrix}$$

Fig. 4. Similarity relation on numbers $1, \ldots, 12$, and the suitable matrix based on the optimal partition.

by 5, but not by 3 or 2, and so on. This can be formulated as $C_2 = \{x \in U : 2|x\}$, $C_3 = \{x \in U : 3|x\} \backslash C_2$, $C_5 = \{x \in U : 5|x\} \backslash C_2 \backslash C_3$, ... Of course, each prime number has one (possibly empty) cluster.

If we consider the fosterer and adversary objects based on the the optimal partition and the similarity relation T in Fig. 4, then we get the matrix A presented in Fig. 4. The partition mentioned above is optimal because it minimises the number of negative numbers in the matrix A. This, of course, also has an impact on rankings. While keeping the numbers in a common cluster, multiple ranks were negative due to dissimilarities, by using the optimal partition each cluster as a cluster by applying similarity (Table I/C). For singletons (containing big primes and 1) the rank 1.0 is natural, as there are no similar numbers. In the set of even numbers, the rank of the powers of two will also be 1.0, as they are similar to all even numbers, and the C_2 cluster has only even numbers, and all even numbers are here. Numbers with other divisors will have similar numbers in other clusters. The more prime divisors a number has, the more clusters contain similar numbers, so its rank will be reduced. In the cluster C_3 the rank of the powers of three will be the highest, but it will not reach level 1.0, because there are nearly as many numbers that are dividable by 3—that is, similar—in the cluster C_2, as in the cluster C_3.

If we apply our method on numbers $1, \ldots, 15$ instead of numbers $1, \ldots, 12$, then the ranks change (Table I/D). The rank of 7 fell, as its adversary number (14) appeared. As number 15 is adversary for both 5 and 10, it reduces the rank of both of them. The reader may be wondering how these ranks look when we have more numbers, e.g. $1 \ldots, 1000$ (Table II, column r). Perhaps it is clear from the above that in case of an optimal clustering, the representatives of the individual clusters come from the powers of primes. Each power of a prime is given the same rank because it is completely symmetrical in terms of similarity.

B. Weakening the tolerance relation

Let's see what happens if we expect one of the numbers to be the *divisor* of the other number instead of the existence of a real common divisor. (Remember, the relation must be symmetrical.) The partial relation here is made from a complete relation. We suggest to the reader to compare the first matrices in Fig. 4 and 5. If two numbers were dissimilar at the original tolerance relation, they will be dissimilar at the

TABLE II
RANG OF NUMBERS $1, \ldots, 1000$ USING THE OPTIMAL PARTITION

x	$r(x)$	$r'(x)$
1	1.0	1.0
2	1.0	1.0
3	0.71587172	0.79465596
5	0.78447065	0.81182099
7	0.82770503	0.84160798
11	0.88038405	0.88841938
13	0.89607481	0.90279018
17	0.91498232	0.91946078
4	1.0	0.8102978
6	0.75571551	0.55292589
8	1.0	0.71548803
10	0.85121315	0.5761841

$$T' = \begin{pmatrix}
1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & -1 & 0 & -1 & 1 & -1 & 0 & -1 & 1 \\
-1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & -1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 0 & -1 & 1 & -1 & 0 & -1 & 0 \\
-1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & -1 & 0 \\
-1 & 1 & -1 & 0 & 1 & 0 & -1 & 0 & -1 & 1 & -1 & 0 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & -1 & 1
\end{pmatrix}$$

$$A = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}$$

Fig. 5. Weakened similarity relation on numbers $1, \ldots, 12$, and the suitable matrix based on the optimal partition.

weakened relation, too. Moreover for example, numbers 4 and 6 were similar before, but not anymore; so the relation holds less often.

At first glance in Table I/E the ranks of 2, 4, 8 are different, as well as ranks of 3 and 9. Once we have only one cluster, it is sufficient to count how many positive and negative values are in each row of the first matrix in Fig. 5. The number 2 is similar to every even number, that is, to every object in its own cluster C_2, and there is no other similar numbers anywhere else. The number 4 is similar to half of the numbers of its own cluster, but not dissimilar to any numbers in this cluster. The number 8 is similar to quarter of the numbers of its own cluster, etc. The fact that we replaced some ones with zeros, the symmetry disappeares. We remark that the numbers 1, 7 and 11 are dissimilar from all the other numbers in this case, so now they have the same (negative) rank.

Let's see what happens, if we take our previous optimal partition (Table I/F). The previous asymmetry remains. The primes have the highest rank, and the powers of primes have lower ranks. To see this tendency, let's see the outcome of ranking $1, \ldots, 1000$ (Table II, column of r').

If we apply the optimal partition for tolerance relation T to the weakened tolerance relation T' (Table I/F), the number of negative numbers in matrix A will also be significantly reduced. However, because of the change in the relation, the symmetry within the clusters is severely damaged, which also affects the rankings. Therefore, when some ranks weaken in the C_2 cluster, this may affect elements in other clusters. If there exists fosterer elements in other clusters, these will in turn increase in rank.

Which partition could be optimal for this weakened relation? Let us take the number of similar and dissimilar numbers in the clusters according to some given number. If the difference of these numbers not in their own cluster is maximal, by moving the actual number to the maximal cluster, we get a better partition. So now we wish to examine, whether we improve the optimal partition. Let $V = \{1, \ldots, 1000\}$, and take the number 75 as an example, which is $3 \cdot 5^2$. With the original relation T, C_2 contains 267 numbers are similar to it (which are divisible by 3 or 5), and 233 which are dissimilar. C_3 contains 167 similar numbers (all of them divisible by 3) and C_5 contains 67 similar numbers. At a weakened relation T', the previously dissimilar numbers remain dissimilar, and in C_2 there are no divisors of 75, just its even multiples. These by definition are similar to it, so C_2 contains 6 members similar to 75. In C_3 we have three divisors of 75 (including itself), as well as 6 of its odd multiples, so C_3 contains 9 similar numbers. In C_5 we have two divisors of 75 and it does not contain any of its multiples (because any multiple needs to have a prime divisor 3, so it would either be in C_3, or in C_2 if the number is even). If we take the difference of similar and dissimilar numbers for each clusters, this will be maximal in the number's own cluster, as here for C_3. This means that this kind of partition is stable for T', and with a very high probability it is the optimal partition, but this needs to be proved.

V. CONCLUSION AND FURTHER WORK

In this article, we presented a method that assigns a rank to each object by using a similarity relation and a partition of objects. In a special case, the method can be used for a single cluster, as shown by two examples. The similarity relation does not have to be complete, the method also works for partial relations. Based on our tests, using successive squaring to apply our method to a thousand objects, it is enough to use around dozens of squaring matrices. Since the tools used (matrix multiplication, norm calculation) are available in standard mathematical packages in almost all programming languages, this method is widely applicable.

Our future plans include a method to test real-life data, for example, in case of wines described by their chemical characteristics [1], we want to search for a typical wine—i.e. representative—of some wine region. Similarly we are planning to apply this method to medical data, where we will look at suggesting treatments based on the patients' medical history.

In real life it is very rare to find perfect similarity, thus we need to introduce a system of different levels of similarity. So then a statement such as *Bob is more like Charlie than Alice*, can be translated into partial similarity by saying that the similarity between Bob and Charlie is 0.8, whilst the

similarity of Alice and Bob is only 0.2. Therefore we can think of partial similarity as an extension of the similarity relations. In our case, we only used three values in the matrix of the similarity relation T: 1, -1 and 0. Then, for partial similarities we can extend this set of values to the interval $[-1, 1]$, therefore covering a range of different possibilities. The algorithm of von Mises can still be used in this case.

Previously, we have developed a method for describing the similarity of objects given by incomplete information [4], which can be traced back to partial similarity.

REFERENCES

[1] S Aeberhard, D Coomans, and O De Vel. Comparison of classifiers in high dimensional settings. *Dept. Math. Statist., James Cook Univ., North Queensland, Australia, Tech. Rep*, 92(02), 1992.

[2] Alon Altman and Moshe Tennenholtz. Ranking systems: the pagerank axioms. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 1–8. ACM, 2005.

[3] László Aszalós, Lajos Hajdu, and Attila Pethő. On a correlational clustering of integers. *Indagationes Mathematicae*, 27(1):173–191, 2016.

[4] László Aszalós and Tamás Mihálydeák. Rough classification in incomplete databases by correlation clustering. In *2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*. Atlantis Press, 2015.

[5] R. V. Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung . *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.

[6] Z. Néda, R. Sumi, M. Ercsey-Ravasz, M. Varga, B. Molnár, and Gy. Cseh. Correlation clustering on networks. *Journal of Physics A: Mathematical and Theoretical*, 42(34):345003, 2009.