A Rank-Based Approach of Cosine Similarity with Applications in Automatic Classification

Conference Paper · September 2012

DOI: 10.1109/SYNASC.2012.24

2 authors:

Liviu P Dinu
University of Bucharest
100 PUBLICATIONS **510** CITATIONS
SEE PROFILE

Radu Tudor Ionescu
University of Bucharest
83 PUBLICATIONS **630** CITATIONS
SEE PROFILE

A Rank-based Approach of Cosine Similarity with Applications in Automatic Classification

Liviu P. Dinu and Radu-Tudor Ionescu
Department of Computer Science
University of Bucharest
14 Academiei, Bucharest, Romania
E-mails: ldinu@fmi.unibuc.ro, raducu.ionescu@gmail.com

***Abstract*—This paper introduces a new rank-based approach of the cosine similarity that can be applied in the competitive area of automatic classification. We describe the method and then we present some of its mathematical and computational properties.**

Tests are performed on a dataset consisting of handwritten digits extracted from a collection of Dutch utility maps. The experimental results are compared with other reported results based on different combining methods which used the same dataset. The obtained results show that the rank-based approach of the cosine similarity can successfully be used for automatic classification or similar tasks.

***Keywords*-rank distance; cosine similarity; median string; classification; digits classification; handwritten digits classification; voting; aggregation scheme; combination scheme; combination rule.**

I. Introduction

Decision taking processes are a common and frequent task for most of us in our daily life. Many of the decision taking problems can be solved with the help of computers. A very common task of such kind is object classification. Usually, the solutions for object classification or similar decision taking problems are approximate and researchers continuously investigate new methods that improve the accuracy and performance. Most of the proposed solutions are based on statistics or machine learning. For many state-of-art machine learning techniques (such as SVM [1], kernel methods [2]) the decision is taken based on some kind of similarity or dissimilarity between objects. Currently, a very high number of distance measures exist in literature and there are efforts to group them in specific areas of interest [3].

Despite of the great number of distances in literature, new distances are periodically explored. Mostly, these tools are based on some quantitative properties or features of implied objects. The last decade reveals in some practical cases the advantage of ordinal distances in similarity and classification methods [4, 5].

In some cases, regarding features as ordinal variables performs better than regarding them as frequency or other measures. Looking at function features as ordinal variables means that in the calculation of a distance/similarity function, the ranks of features (made up according to their quantitative properties) will be used rather than the actual values of these quantitative properties. Usage of the ranking in the computation of the distance/similarity measure instead of the actual values of the frequencies may seem as a loss of information. From another point of view, the process of ranking makes the distance/similarity measure more robust acting as a filter and eliminating the noise contained in the values of the quantitative properties. For example, the fact that a particular feature has the rank 2 (is the second most frequent feature) in one object and the rank 4 (is the fourth most important feature) in another object can be more relevant than the fact that the same feature appears 34% times in the first object and only 29% times in the second one.

Researchers periodically study and develop new methods for automatic object classification. There are many algorithms that are able to solve classification tasks with great accuracy. Starting with a similarity measure based on rankings, the idea of combining state-of-art classifiers using the similarity measure comes naturally. The problem of combining classifiers has been intensively studied in the last period [6] and various classifier schemes [7, 8, 9] have been devised and experimented in different domains: document classification, document image analysis, biometric recognition (personal identification based on various physical attributes such as iris, face, fingerprint) or speech recognition are few of them. A typical combination schema consists of a set of individual classifiers and a combiner which aggregates the results of the individual classifiers to make the final decision. In many situations, the results of individual classifiers are *rankings* (an ordered list of objects). Each ranking can be considered as being produced by applying an ordering criterion to a given set of objects.

The concept of ordering several objects, and consequently obtaining a ranking is encountered in many situations: an electoral process (where the ordering criterion between the participants is given by the number of votes they gained); the results of a football tournament (where the criterion is the number of points obtained by each team at the end of the tournament), etc. However, it is not the general case to have very simple methods to decide which is the ordering

criterion, and, as a consequence, it becomes difficult to define and build the ranking. To support this statement, we mention situations like selecting documents based on multiple criteria, building search engines for the WEB [10] or finding the author of a given text. Examples of multi-criteria selection arise when trying to select a product from a database of products, such as travel plans or restaurants (users might rank restaurants based on several different criteria like cuisine, driving distance, ambiance, star-rating, etc). Other situations when we combine rankings are those when we take decisions based on subjective or sensorial criteria (for example, perceptions). Especially when working with perceptions, but not only, we face the situation of operating with rankings of objects where the essential information is not given by the numerical value of some parameter of each object. Instead, this information is given by the position occupied by the object in the ranking, like movies or music tops (according to a natural hierarchical order, in which on the first place we find the most important element, on the second place the next one and on the last position the least important element). In order to make a decision in all these situations, we have to combine two or more rankings which have been ordered by using different criteria. We deal with the so-called *rank aggregation problem*.

Let us describe the paper organization. Section II gives a brief discussion about rankings and associated metrics. A rank based approach of the well known cosine similarity is proposed in section III. This section also introduces a combining ranking schema based on the *cosine rank similarity*. The performance of our method on the dataset consisting of handwritten digits is tested in section IV. Here we compare our experimental results with other reported results on the same dataset that were obtained with different combining methods. Finally, in section V we draw our conclusions and give some ad-hoc extensions of the proposed similarity.

II. Ordinal Measures

A. Rankings

A ranking is an ordered list of objects. Every ranking can be considered as being produced by applying an ordering criterion to a given set of objects.

Let U be a finite set of objects, called the universe of objects. We assume, without loss of generality, that $U = \{1, 2, \ldots, |U|\}$ (where by $|U|$ we denote the cardinality of U). A ranking over U is an ordered list: $\tau = (x_1 > x_2 > \ldots > x_d)$, where $\{x_1, \ldots, x_d\} \subseteq U$, and $>$ is a strict ordering relation on $\{x_1, \ldots, x_d\}$, what we have called in section I *an ordering criterion*. It is important to point the fact that $x_i \neq x_j$ if $i \neq j$. For a given object $i \in U$ present in τ, $\tau(i)$ represents the position (or rank) of i in τ.

A ranking defines a partial function on $\mathcal{U}$ where for each object $i \in \mathcal{U}$, $\tau(i)$ represents the position of the object i in the ranking τ. Observe that the objects with high rank in τ have the lowest positions.

If the ranking τ contains all the elements of U, then it is called a *full list (ranking)*. It is obvious that all full lists represent all total orderings of U (the same as the permutations of U). However, there are situations (see [10] for example) when some objects cannot be ranked by a given criterion: the list τ contains only a subset of elements from the universe U. Then, τ is called *partial list (ranking)*. We denote the set of elements in the list τ with the same symbol as the list.

B. Metrics on Rankings

Usually, the distance measures between rankings are defined for the case of full lists. Some of the most used measures are (see [11]): the *Spearman footrule distance* and the *Kendall tau distance*.

A problem arises when one tries to apply the distances above to partial lists: in the most cases the newly defined functions do not preserve the property of being a metric function (as it is shown in [10]). In [12] a distance is introduced which preserves this property, namely the *rank distance*. It is the distance that we will use in developing our method. In the following, we shortly present the rank distance.

A few preliminary notations are explained first. Let $\sigma = (x_1 > x_2 > \ldots > x_n)$ be a partial ranking over U; we say that n is the length of σ. For an element $x \in U \cap \sigma$, one defines the order of the object x in the ranking σ: $ord(\sigma, x) = |n + 1 - \sigma(x)|$. In other words one assigns different weights to each element of the ranking in decreasing order from top to bottom. More precisely, one attributes the highest rank n to the first element of the ranking, then the rank $n - 1$ to the second element, and so on until the final element is assigned with the lowest rank 1. If $x \in U \setminus \sigma$, we have $ord(\sigma, x) = 0$.

Definition 1: Given two partial rankings σ and τ over the same universe $\mathcal{U}$, we define the rank distance between them as:

$$\Delta(\sigma, \tau) = \sum_{x \in \sigma \cup \tau} |ord(\sigma, x) - ord(\tau, x)|.$$

Example 1: Let $\sigma = (1 > 2 > 3 > 4)$ and $\tau = (5 > 1 > 2)$ be rankings over the universe $U = \{1, 2, 3, 4, 5\}$. According to Definition 1, we have:

$$\Delta(\sigma, \tau) = \sum_{x \in \{1,\ldots,5\}} |ord(\sigma, x) - ord(\tau, x)|$$

$$= |4 - 2| + |3 - 1| + |2 - 0| + |1 - 0| + |0 - 3| = 10$$

In [12] Dinu proves that Δ is a distance function. The rank distance is an extension of the Spearman footrule distance [13].

Note that Rank Distance can be extended to compute the distance of one ranking to a multiset of rankings in the intuitive way by computing the distance between the given

ranking and each of the ranking from the multiset and then by adding these values:

Definition 2: Let $T = L_1, L_2, \ldots, L_n$ be a multiset of n rankings and let L be a ranking. Then, the rank distance from L to T is defined by:

$$\Delta(L, T) = \sum_{x=\{1,\ldots,n\}} \Delta(L, L_i).$$

The motivation for the usage of objects' order in a ranking, instead of the rank itself, comes from at least two directions. First, one considers that the distance between two rankings should be greater if they are more different at top (on the high ranked objects), since in many applications the low ranked objects are neglected. Consequently, the objects with high ranks should have a greater weight. Second, the length of the rankings is also important: if a ranking is longer, we consider that the criterion that produced it performed a more profound analysis of the objects, hence, it is more reliable than another criterion that produced a shorter ranking. Although, for example, two rankings of different length may have the same object on the first position, this object has different orders (in the sense of the upper definition) in the two rankings, and, this difference should be reflected in the total distance.

From the point of view of the time complexity needed to compute the rank distance between two rankings, the time needed to compute the distance between two partial rankings over an universe U is linear in the cardinality of U. Thus, rank distance is easy to implement and has an extremely good computational behavior. Another advantage of rank distance is that it imposes minimal hardware demands: it runs in optimal conditions on modest computers, reducing the costs and increasing the number of possible users. For example, the time needed to compare a DNA string of $45,000$ nucleotides length with other 150 DNA strings (with similar length), by using an laptop with 224 MB RAM and 1.4 GHz processor is no more than six seconds. Two algorithms that compute rank distance in linear time are given in [14].

III. Cosine Rank Similarity

A. Cosine Similarity

Cosine similarity is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them. Given two vectors of attributes, A and B, the cosine similarity, θ is represented using a cross product as:

$$similarity = cos(\theta) = \frac{A \times B}{\parallel A \parallel \parallel B \parallel}.$$

For text matching, the attribute vectors A and B are usually the term frequency vectors of two text documents. The cosine similarity can be seen as a method of normalizing document length during comparison.

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same. Obviously, 0 indicates independence, and in-between values indicate intermediate similarity or dissimilarity between vectors A and B.

In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (TF-IDF weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90^0. When the angle is 90^0, it means that the two term frequency vectors are completely opposite.

B. Rank-based Cosine Similarity

Definition 3: Given two full ranking $f = (f_1, f_2, \ldots, f_n)$ and $g = (g_1, g_2, \ldots g_n)$ over the same universe U of cardinality $|U| = n$, we define $CosRank(f, g)$ similarity as follow:

$$CosRank(f, g) = \frac{< f, g >}{\parallel f \parallel \parallel g \parallel} = \frac{< f, g >}{1^2 + 2^2 + \ldots + n^2}$$

$$= \frac{\sum_{x \in U} ord(x \mid f) \times ord(x \mid g)}{1^2 + 2^2 + \ldots + n^2}$$

In other words, we are interested by the positions of x in f and g, respectively, and we use a cosine-like measure. Note that $CosRank(f, g) = 1$ if and only if f and g are identical rankings. As f and g are more different, $CosRank(f, g)$ tends to get closer to 0.

In what follows, we will use the $1-CosRank$ measure. It is not hard to show that $1-CosRank$ is a distance function. Let's consider the following standard definition of a distance function

Definition 4: A metric on a set X is a function (called the distance function or simply distance) $d : X \times X \to \mathbb{R}$. For all $f, g, h \in X$, this function is required to satisfy the following conditions:

1. $d(f, g) \geqslant 0$ (non-negativity, or separation axiom);
2. $d(f, g) = 0$ if and only if $f = g$ (coincidence axiom);
3. $d(f, g) = d(g, f)$ (symmetry);
4. $d(f, g) \leqslant d(f, h) + d(h, g)$ (triangle inequality).

Note that conditions 1 and 2 produce positive definiteness. Also, observe that the first condition is implied by the others. Since $CosRank(f, g) = 1$ if and only if f and g are identical rankings, it is obvious that $1-CosRank(f, g) = 0$. Thus, we have the coincidence axiom. From Definition 3 observe that $CosRank$ is symmetric (i.e., $CosRank(f, g) = CosRank(g, f)$) since:

$$\sum_{x \in U} ord(x \mid f) \times ord(x \mid g) = \sum_{x \in U} ord(x \mid g) \times ord(x \mid f)$$

The triangle inequality is ensured by the fact that f, g, h are all full rankings.

C. CosRank Combining Scheme

In [15] we introduce a rank based combining scheme starting from the aggregation problem. We apply with good results this scheme in digit recognition [15] and in text categorization [4]. Starting from ideas developed in [15], we introduce a similar combining scheme based on a aggregation problem, only that here we use the CosRank similarity instead of rank distance.

In a selection process, rankings are issued for a common decision problem, therefore a ranking that "combines" all the original (base) rankings is required. One commonsense solution is finding a ranking that is as close as possible to all the particular rankings. Apart from many paradoxes of different aggregation methods, this problem is NP-hard for most non-trivial distances [16]. On the other hand, a solution for this problem can be found in polynomial time for rank distance [17].

Formally, given a multiset $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_k\}$, we aggregate the rankings by using $1 - CosRank$ similarity. This means that we are looking for those rankings σ that have a minimal $1 - CosRank$ distance to all the rankings in the multiset. In other words, we have to minimize the sum:

$$1 - CosRank(\sigma, \mathcal{T}) = 1 - \sum_{\tau \in \mathcal{T}} CosRank(\sigma, \tau).$$

The CosRank classification scheme has two important steps. The first one is to obtain all the rankings which minimize the upper equation in a similar fashion to the solution of rank aggregation [17]. The second step is to apply voting on all the obtained rankings.

IV. Application

In this section we make a comparative study regarding the behavior of six combining schema on the same input data set. The input dataset consists of handwritten digits extracted from a collection of Dutch utility maps. We compare the results of the CosRank combining scheme with the *rank distance combining* (RDC) scheme from [15].

In [8] an experiment regarding the error rate (in percentage) of different classifiers and classifier combination schemes on a digit classification problem is reported. We use the same data to test the performance of our combining scheme.

A brief description of the experiment is available in [15]. For a more comprehensive description see [8] and [18]. The experiments are done on a data set which consists of six different feature sets for the same set of objects. The six feature sets are:

- Fourier: 76 Fourier coefficients of the character shapes.
- Profiles: 216 profile correlations.
- KL-coef: 64 Karhunen-Love coefficients.
- Pixel: 240 pixel averages in 2×3 windows.
- Zernike: 47 Zernike moments.
- Morph: 6 morphological features.

The 12 classifiers $(c_1, \ldots, c_{12})$ used in the experiment and the 6 combining rules are listed in [15]. The results of the CosRank combining scheme on the dataset consisting of handwritten digits are summarized bellow.

In Table I we present results for 12 individual classifiers $(c1, c2, ..., c12)$. The 12 classifiers used in the experiment are listed in [15]. Note that the combining rules are applied on the six feature sets for a single classification rule. The results obtained with CosRank are better than those obtained with the RDC scheme for the top classifiers. This is a very good result if we consider that RDC scheme was the one of the best schemes tested in [15].

Table I
CosRank vs. RDC: Results for 12 individual classifiers. The success rate of each classifier (listed in lines) is given for each combining scheme (listed on columns).

Classifier	CosRank	RDC
c1	97.3%	96.5%
c2	97.9%	97.1%
c3	94.8%	93.9%
c4	97.8%	97.3%
c5	97.8%	97.2%
c6	98.4%	97.5%
c7	96.6%	97.2%
c8	65.8%	85.0%
c9	51.8%	74.5%
c10	49.0%	81.9%
c11	77.1%	95.0%
c12	94.3%	95.8%

Table II shows the results of the CosRank and RDC combining rules on each of the six feature sets. It seems that CosRank has a slightly lower accuracy than RDC, but it is still above most of the combining rules tested in [15].

Table II
CosRank vs. RDC: Results for six feature sets. The success rate for each feature set (listed in lines) is given for each combining scheme (listed on columns).

Feature set	CosRank	RDC
f1	82.6%	83.6%
f2	95.5%	96.6%
f3	96.0%	96.7%
f4	94.5%	96.6%
f5	80.9%	83.4%
f6	68.0%	70.7%

Finally, we applied all 12 classifiers to all six features and obtained for each document a multiset of 72 rankings. In [15] these rankings are combined by using RDC scheme. The reported success rate of RDC is 98.2%. Using CosRank combining scheme instead of RDC we obtain a succes rate of 97.9%. However, if we aggregate the best classifiers with feature sets $f1, f2, ..., f6$, the success rate of CosRank classification scheme is 98.7%. This is the best success rate over all methods.

V. Conclusions

In this paper we introduced CosRank similarity, a rank based approach of cosine similarity. It is a ordinal measure that can be computed in linear time. We tested the similarity on a dataset consisting of handwritten digits extracted from a collection of Dutch utility maps. Our results are compared with other reported results based on a different combining method, namely RDC [15]. We conclude that the rank-based approach of the cosine similarity is the best over all methods that report results on the same dataset. In future work we intend to propose a weighted variant of the CosRank combining method.

Acknowledgment

The contribution of the authors to this paper is equal. Radu Ionescu thanks his Ph.D. supervisor Denis Enachescu from the University of Bucharest. The research of Liviu P. Dinu was supported by the CNCS-PCE Idei grant 311/2011. The authors also thank to the anonymous reviewers for helpful comments.

References

[1] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[3] E. Deza and M.-M. Deza, *Dictionary of Distances*. The Netherlands: Elsevier, 1998.

[4] L. P. Dinu and A. Rusu, "Rank distance aggregation as a fixed classifier combining rule for text categorization," *In Proceedings of CICLing 2010*, pp. 638–647, 2010.

[5] L. P. Dinu and M. Popescu, "Ordinal measures in authorship identification," *In Stein, Stamatos, Koppel, Agire (eds.) PAN '09*, pp. 62–66, 2009.

[6] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[7] T. K. Ho, J. Hull, and S. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[8] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[9] C.-L. Liu, "Classifier combination basedon confidence transformation," *Pattern Recognition*, vol. 38, pp. 11–28, 2005.

[10] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," *In Proceedings of the 10th International Conference on WWW*, pp. 613–622, 2001. [Online]. Available: http://doi.acm.org/10.1145/371920.372165

[11] P. Diaconis, "Group representation in probability and statistics," *IMS Lecture Series 11*, 1988.

[12] L. P. Dinu, "On the classification and aggregation of hierarchies with different constitutive elements," *Fundamenta Informaticae*, vol. 55, no. 1, pp. 39–50, 2003.

[13] P. Diaconis and R. L. Graham, "Spearman footrule as a measure of disarray," *Journal of Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 2, pp. 262–268, 1977.

[14] L. P. Dinu and A. Sgarro, "A Low-complexity Distance for DNA Strings," *Fundamenta Informaticae*, vol. 73, no. 3, pp. 361–372, 2006.

[15] L. P. Dinu and M. Popescu, "A multi-criteria decision method based on rank distance," *Fundamenta Informaticae*, vol. 86, no. 1–2.

[16] C. de la Higuera and F. Casacuberta, "Topology of strings: Median string is np-complete," *Theoretical Computer Science*, vol. 230, pp. 39–48, 2000.

[17] L. P. Dinu and F. Manea, "An efficient approach for the rank aggregation problem," *Theoretical Computer Science*, vol. 359, no. 1-3, pp. 455–461, 2006.

[18] R. P. W. Duin and D. M. J. Tax, "Experiments with classifier combining rules," *In Proceedings of MCS '00*, pp. 16–29, 2000.